

CS485/685 Computer Vision
Spring 2012 – Dr. George Bebis
Programming Assignment 3
Due Date: 4/17/2012

In this assignment, you will implement the Harris-Laplace interest point detector to extract scale invariant regions.

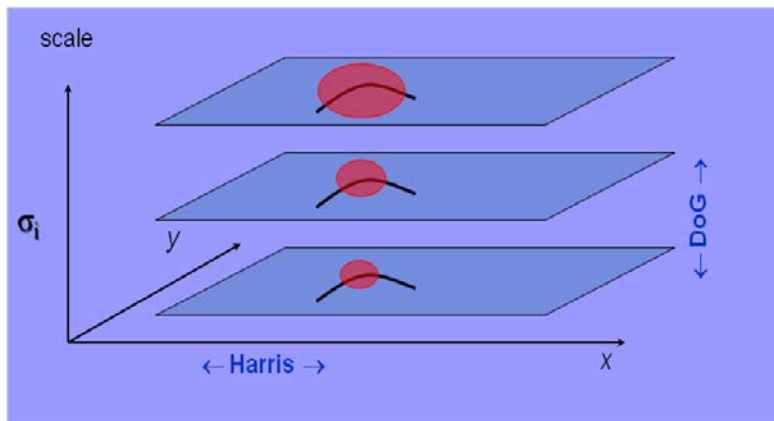
1. Harris Corner Detector

Implement the Harris corner detector by following the steps discussed in class. Using the test images provided on the course's webpage (i.e., Test1, Test2, and Test3), show the following results: (i) corner response $R(A_w)$ (i.e., using $\alpha=0.06$, derivative of Gaussian function, $\sigma_D=\gamma\sigma_1$, $\sigma_1=1.5$ and $\gamma=0.7$), (ii) points with large corner response (i.e., $R(A_w)>t$, where $t = 1\%$ of the maximum observed $R(A_w)$ value), (iii) local maxima of $R(A_w)$ from previous step (i.e., using a 3×3 neighborhood). To better visualize the corners detected, overlay them with the original image.

2. Harris-Laplace Interest Point Detector

Implement the Harris-Laplace interest point detector by following the steps discussed in class. Here is an outline of the main steps; for more details, look at the lecture slides and [1]. You will be using a DoG Scale Space to approximate the Laplacian of Gaussian (LoG) Scale Space.

1. Generate a Gaussian Scale Space (G_{SS})
2. Generate a Difference-of-Gaussian Scale Space (DoG_{SS})
3. At each level of the G_{SS} , extract a set of interest points using the Harris corner detector.
4. Perform non-maxima suppression at each scale to find the strongest corners.
5. Perform non-maxima suppression along scales to find most stable corners and extract their characteristic scale.



Steps 1 and 2

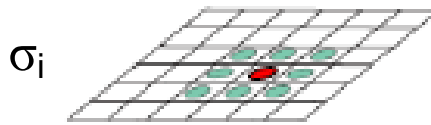
To create the G_{SS} , start from some initial scale σ_0 and iterate n times by increasing the scale by a factor k each time (i.e., $\sigma_0, k\sigma_0, k^2\sigma_0, \dots, k^n\sigma_0$). Use $\sigma_0=1.5$, $k=1.2$, and $n=17$; for simplicity, do not perform sub-sampling. Once the G_{SS} has been built, build the DoG_{SS} by subtracting adjacent levels in G_{SS} as you did in assignment 1.

Step 3

Using the Harris corner detector, compute a set of interest points (i.e., corners) at each level of the G_{SS} . Note that the Harris detector applies smoothing prior to computing the derivatives. Therefore, you should not apply the Harris detector on the G_{SS} images because you will be smoothing twice. Instead, to compute the Harris corners at level m , apply the Harris corner detector on the original image using $\sigma_D=\gamma\sigma_1$, $\sigma_1=k^m\sigma_0$, $\sigma_0=1.5$, and $\gamma=0.7$

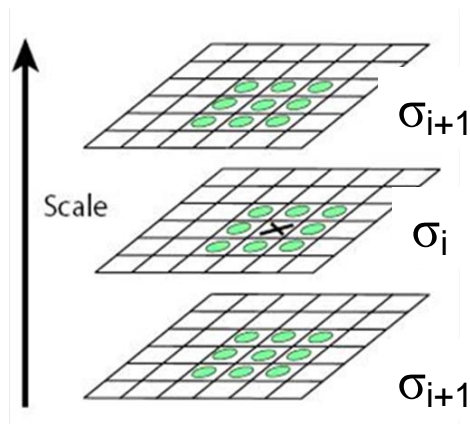
Step 4

For each interest point detected in step 3, determine whether it corresponds to a local extremum (i.e., local max or min) by comparing its $R(A_W)$ value to the $R(A_W)$ values of its eight neighbors (e.g., using a 3×3 window) as shown in the figure below. Reject corners which do not correspond to a local extremum or their $R(A_W)$ value is less than a threshold t_1 (in the literature, $t_1=1500$ has been used but you might need to experiment more with this threshold).



Step 5

For each interest point detected in step 4, determine whether it corresponds to a local extremum by comparing its DoG value to the DoG values of its eight neighbors at the same scale as well as the nine neighbors at the scale above and the nine neighbors at the scale below (i.e., 26 neighbors) as shown in the figure below. Reject corners which do not correspond to local maxima or their DoG value is less than a threshold t_2 (in the literature, $t_2=10$ has been used but you might need to experiment more with this threshold).



Visualizing interest points: you need to show each interest point along with its spatial extent (i.e., characteristic scale) using a circle of radius $r = \sqrt{2}\sigma$ where σ corresponds to the scale (level) from which the interest point was detected. For clarity, show the location of each interest point using a distinctive symbol (e.g., cross). The interest points should be shown on the original image. Here is some code for circle drawing:

<http://2k8618.blogspot.com/2011/11/midpoint-circle-algorithm-c-program.html>

Experiments

Use the images provided on the course's webpage. For each image, show the interest points detected after (i) step 3, (ii) step 4, and (iii) step 5. In each case, report the number of interest points found. To find which corners match between images, use the transformation matrix provided (i.e., homography) for different pairs of images to map the corners from one image onto the other. Two points \mathbf{x}_a and \mathbf{x}_b correspond if the error in relative point location is less than 1.5 pixel: $\|\mathbf{x}_a - H \mathbf{x}_b\| < 1.5$, where H is the homography between the images. A homography is a 3 by 3 matrix H which relates the pixel co-ordinates in the two images: $\mathbf{x}_a = H \mathbf{x}_b$. When applied to every pixel the new image is a warped version of the original image. Two images are related by a homography H and only if both images are viewing the same plane from a different angle (your assignment).

Graduate Students Only

Implement the affine adaptation step to turn circular blobs into ellipses as shown in the lecture (just one iteration is sufficient). Note that each region should be adapted separately. The selection of the correct window function is essential here. You should use a Gaussian window that is a factor of 1.5 or 2 larger than the characteristic scale of the blob. Note that the lecture slides show how to find the relative shape of the second moment ellipse, but not the absolute scale (i.e., the axis lengths are defined up to some arbitrary constant multiplier). A good choice for the absolute scale is to set the sum of the major and minor axis half-lengths to the diameter of the corresponding DoG circle. Show your results before and after the affine iterations. You need to show each interest point along with its characteristic scale and shape using an ellipse. Here is a link to some code for ellipse drawing:

<http://2k8618.blogspot.com/2011/11/draw-ellipse-midpoint-ellipse-algorithm.html>

What to turn in

You are to turn in a report including a print-out of your source code. Your report should include the following: a description of the experiments, results (i.e., include graphic output of your results), discussion of results and comparisons, and a brief summary of what you have learned.

The report is very important in determining your grade for the programming assignment.

Be well organized, type your reports, and include figure captions with a brief description for all the figures included in your report. Motivation and initiative are greatly encouraged and will earn extra points.

References

[1] K. Mikolajczyk and C. Schmid, "Scale and Affine Invariant Interest Point Detectors", *International Journal of Computer Vision*, 60(1), pp. 63-86, 2004.