

Image Processing: HW 2

Scale-space Blob Detection

Saleem Ahmed

50247637

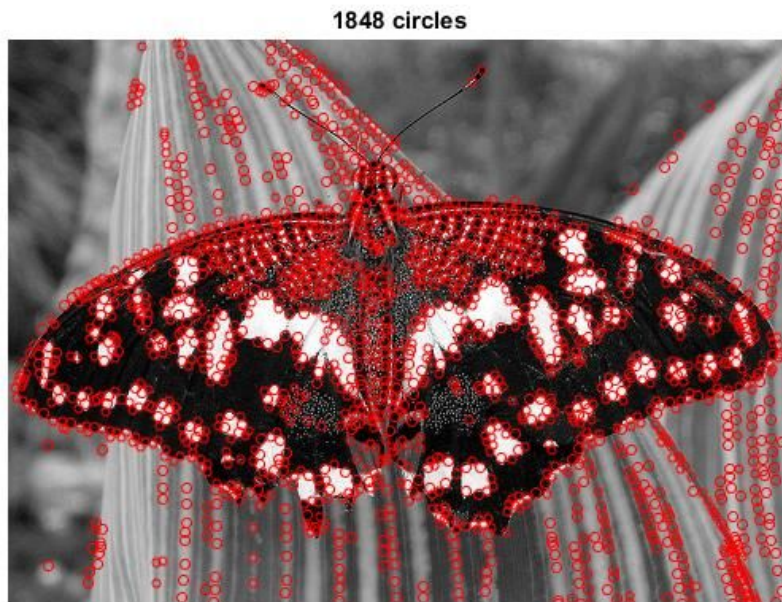
10/15/2017

Values of Constants

Number of Levels in Scale Space

- `numScales = 15;`

This value basically is the number of iterations through which we run our scale space algorithm. That means, the number of times the LoG filter is applied on the image. In each iteration, either the image is downsampled to $1/k$ th of the one before or the filter is increased by a factor k

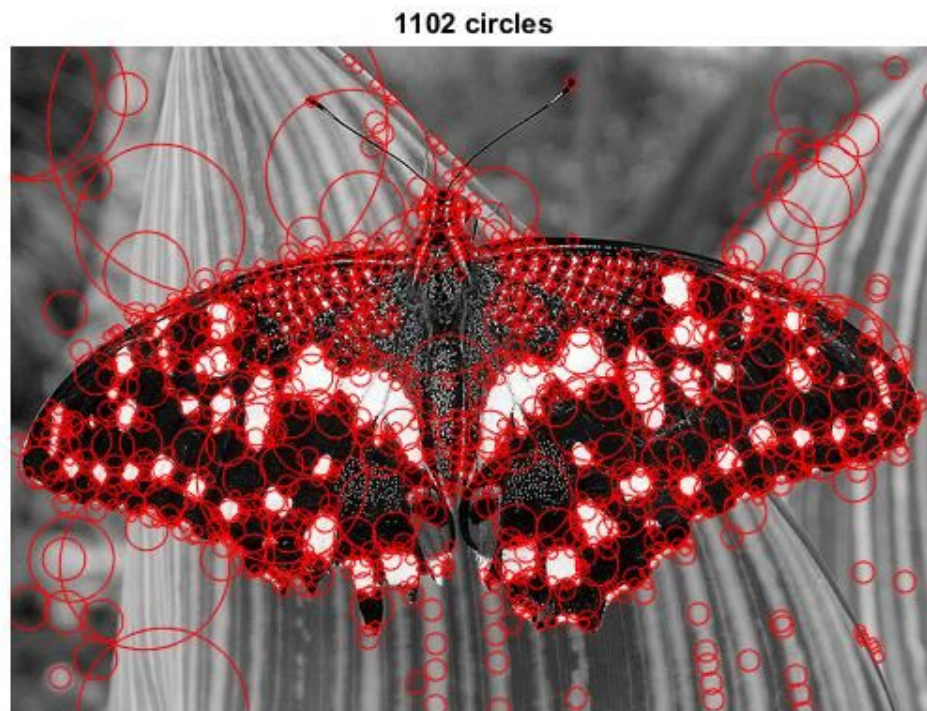


The above shows an output for taking `numScales = 2`. That is blobs are detected at only two levels of the pyramid, as one can see it is not very efficient and actually ambiguous, as all the blobs detected are relatively of the same size.

Value of **k** scale multiplier constant in above pyramid :

- `scaleMultiplier = sqrt(sqrt(2));`

This is taken as a small positive integer. This constant plays an interesting role in determining overlapping blobs at the same scale, for example for a value of `scaleMultiplier = 2` we get the below output :



As one can see we get many circles of the same radius crammed together at the same spots, while many other features are missed.

Value of Spread of the Laplacian of the Gaussian Filter

- `sigma = 2;`

This is the parameter taken by the pre-defined function in matlab, that is nothing but the standard deviation of the curve - that is the distance between the max point and the central tendency. In simple terms higher the value of sigma more is the (blur) spread of the filter. The default value taken by matlab is 0.5. If we take a value less than 1(no spread) , we get no blobs, as the filter created has no effect on the image when convolved.

Value of Threshold for the Laplacian

- `threshold = 0.0095;`

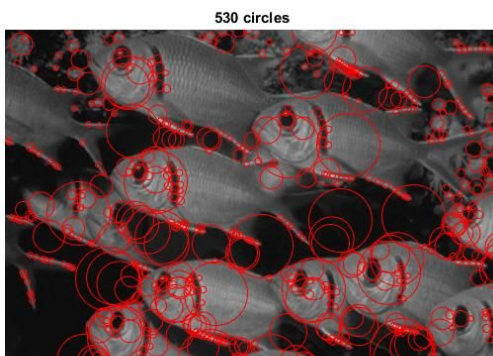
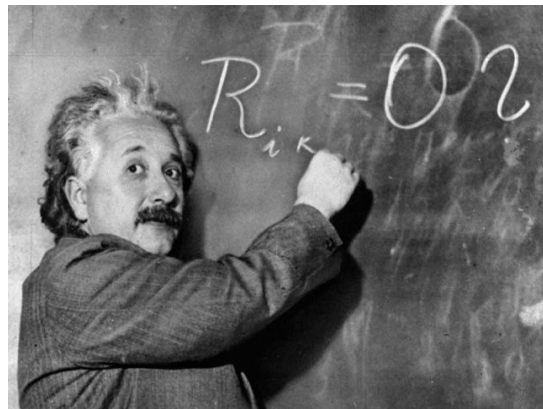
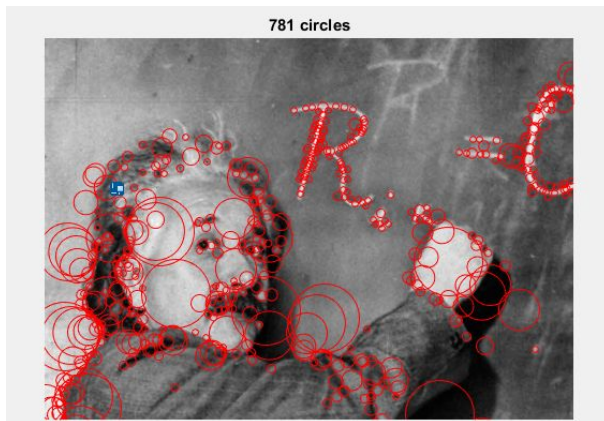
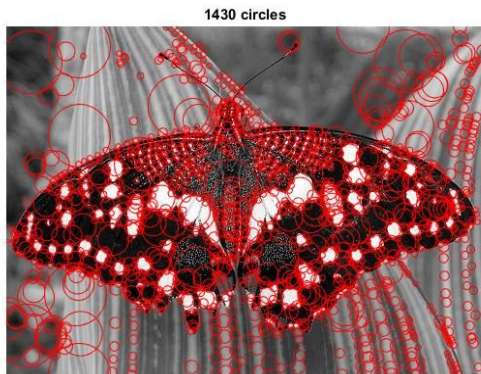
For the double image which is all 0->1, this value is applied on the squared laplacian to basically act as

the cutoff on which the blob detected region is reported. A very low value of this constant leads to very high number of detections (faulty!). Also a very large value of this constant leads to poor detection, ie, very less blobs. I realized a value ~ 0.001 works best for given other parameters. Following is the output for a threshold value = 0 :

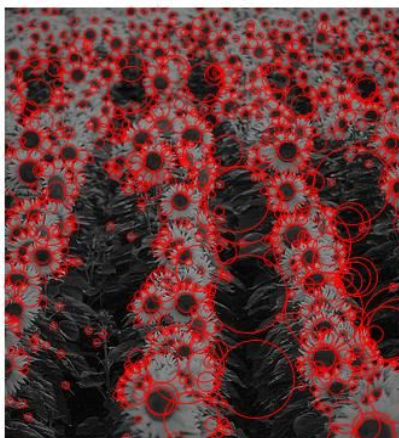
4251 circles



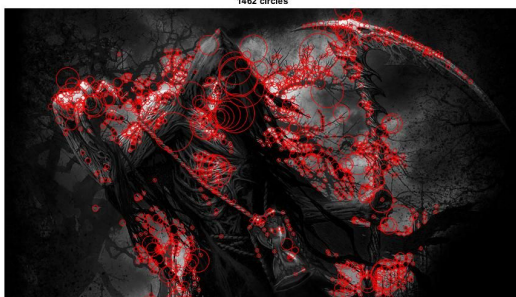
Output For Blob Detection



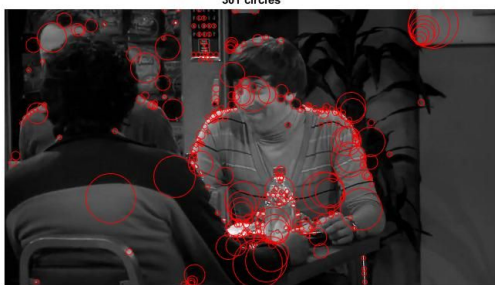
967 circles



1462 circles



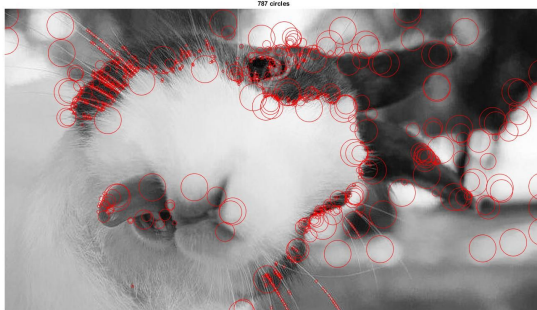
301 circles



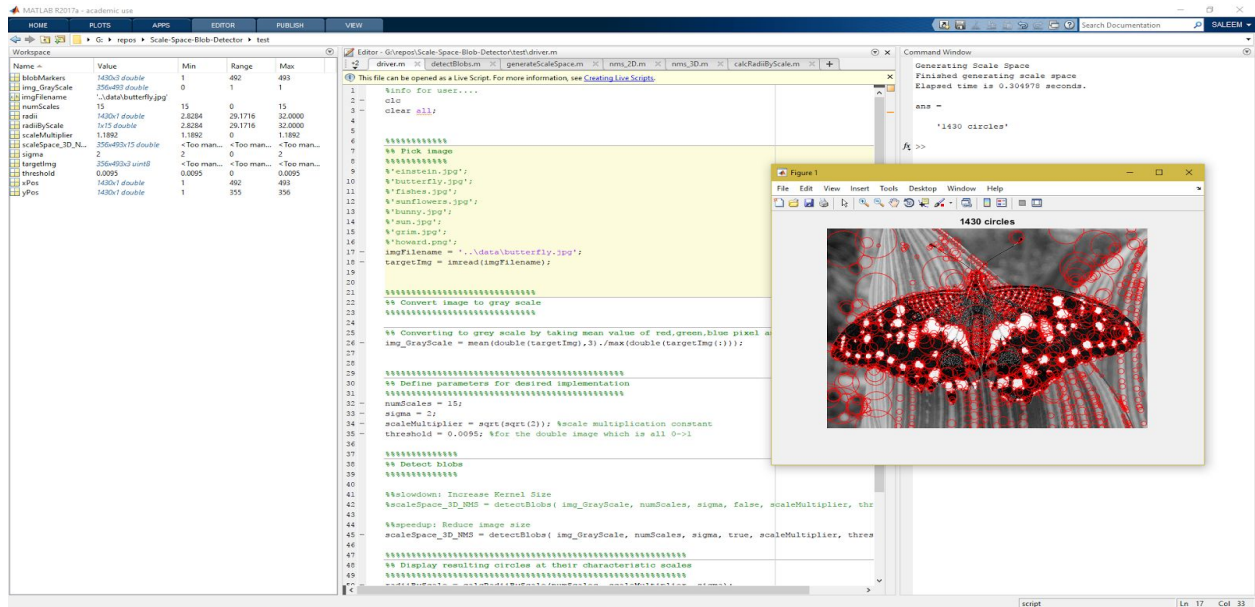
1197 circles



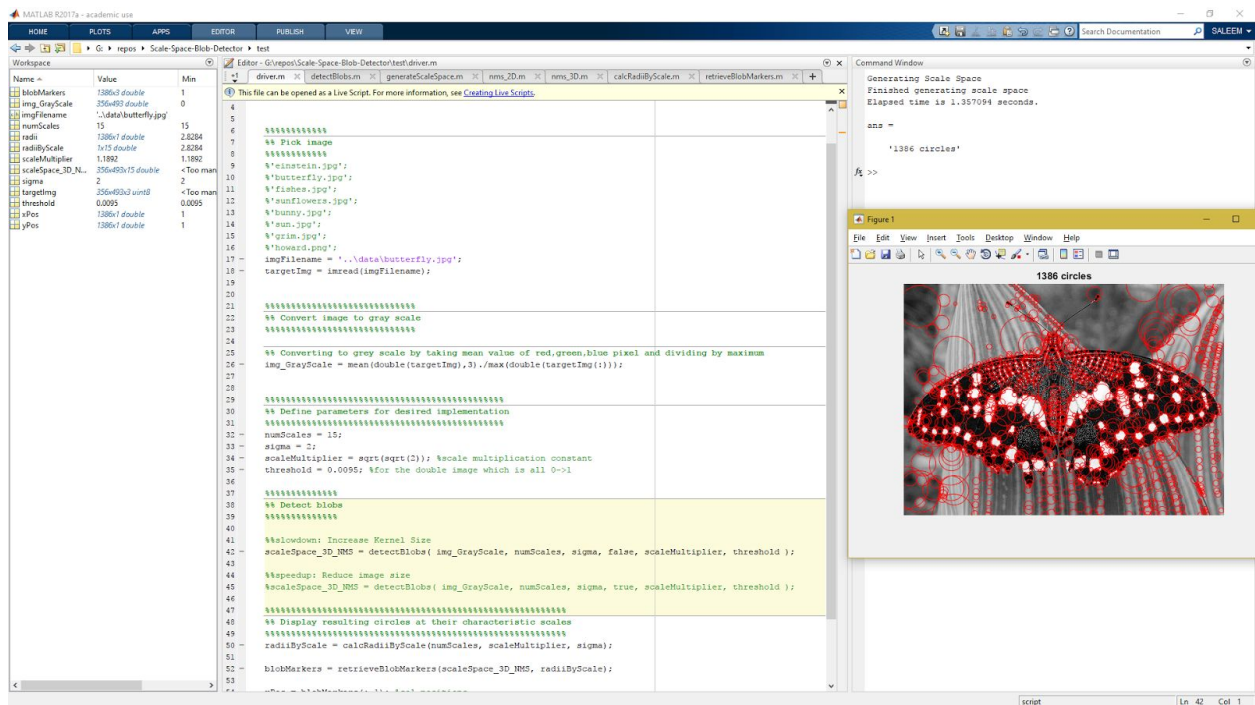
787 circles



Time Comparison for Efficient and Inefficient Method



Time for efficient Method (sample down images) on butterfly ~ 0.3.04978 seconds



Time for In-efficient Method (increase size of filter) on butterfly ~ 1.357094 seconds

Also it is interesting to note that the efficient method not only gives us a faster output but also detects more blobs (1430 nos) as compared to the inefficient method (1386nos)

References :

<http://www.cs.utah.edu/~manasi/coursework/cs7960/p1/project1.html>

<http://www.cs.utah.edu/~jfishbau/advimproc/project1/>

<http://slazebni.cs.illinois.edu/spring16/harris.m>