## **Redundancy Reduction by Retinal Ganglion Cells (simulated with Difference of Gaussians)**

Retinal ganglion cells reduce redundancy. To simulate a retinal ganglion cell we use a Difference of Gaussians (DoG) mask. Produce an on-centre, off-surround DoG mask by subtracting a Gaussian mask with standard deviation 3.5 from a Gaussian mask with standard deviation 2.0. Use the MATLAB command fspecial to create the Gaussian masks that are subtracted from each other.  
  
To simulate the response of retinal ganglion cells to all different parts of an image we can convolve the DoG mask with the image. Use the DoG mask, defined as described in the previous paragraph, to simulate the responses of on-centre, off-surround retinal ganglion cells to all different parts of the woods image. Before performing the convolution convert the woods image from uint8 format to double format. Ensure that the output of the convolution is the same size as the input image. Show the result.  
  
Use the correlation coefficient, which can be calculated for two equally-sized images (or image patches) using the MATLAB command corr2, to measure the similarity between the output of the simulated retinal ganglion cells at neighbouring locations. Ensure that the two image parts being compared are as large as possible by using the method described in week02's MATLAB exercise. Plot a graph of the correlation coefficients calculated between the DoG convolved image and the same image shifted by 0 to 30 pixels vertically.

### **EXERCISE:**

For the output produced by convolving the image with the DoG mask, report the value at the following location (correct to at least 2 decimal places):  
row=134, column=78 Answer  
  
Report the correlation coefficient (correct to 2 at least decimal places) for the following shifts:  
5 pixel shift: Answer  
24 pixel shift: Answer

## **Colour Opponent Cells**

For this exercise use the rooster image converted from uint8 to double format. Simulate the response of blue-off, yellow-on colour-opponent cells to all different parts of the rooster image. The Gaussian used to simulate both the centre and surround should have a standard deviation of 2.25. Ensure that the simulated response is the same size as the input image. Show the result.

### **EXERCISE:**

For the simulated colour-opponent cells report the output at the following locations in the image (correct to at least 2 decimal places):  
row=302, column=5 Answer  
row=171, column=256 Answer

## **Laplacian of Gaussian (LoG) Masks**

A method of combining a derivative mask with a smoothing mask to perform edge detection is provided by the Laplacian of Gaussian (LoG) mask. This is obtained by combining the omni-directional second-derivative mask (the Laplacian) with a Gaussian.  
  
Create two Laplacian of Gaussian masks by convolving a 2-D Gaussian (one with a standard deviation 1.5 and another with a standard deviation of 5) with the following Laplacian mask:

-1/8 -1/8 -1/8  
-1/8 1 -1/8   
-1/8 -1/8 -1/8

Ensure that the resulting masks are sufficiently large to accurately represent the LoG functions.  
  
Generate mesh plots of the two Laplacian of Gaussian masks you have created, put these as subplot(2,2,1) and subplot(2,2,2) in a figure.  
Use the woods image converted from uint8 to double format. Convolve the woods image with each of the Laplacian of Gaussian masks you have created (using "same" as the shape parameter for the conv2 function). Display images showing the output of these two convolutions as subplot(2,2,3) and subplot(2,2,4) in the same figure window.

### **EXERCISE:**

For the output produced by convolving the woods image with the LoG (produced using the Gaussian with the smaller standard deviation), report the values at the following locations (correct to at least 2 decimal places):  
row=76, column=22 Answer  
row=160, column=24 Answer  
  
For the output produced by convolving the woods image with the LoG (produced using the Gaussian with the larger standard deviation), report the values at the following locations (correct to at least 2 decimal places):  
row=76, column=22 Answer  
row=160, column=24 Answer

## **Equivalent LoG and DoG Masks**

The result of convolving an image with a Difference of Gaussians (DoG) mask is similar to the result of convolving an image with a LoG mask: both masks perform edge detection. It is possible to approximate a LoG mask using a DoG, and your task is to determine the standard deviations of the Gaussians used to create a DoG mask that will accurately approximate a LoG mask.  
  
Create a Difference of Gaussians mask by subtracting one Gaussian (with standard deviation 6) from another Gaussian (with standard deviation 3).  
  
Create a Laplacian of Gaussian mask by convolving a 2-D Gaussian with a standard deviation of 5.0 with the following Laplacian mask:

-1/8 -1/8 -1/8  
-1/8 1 -1/8   
-1/8 -1/8 -1/8

Ensure that the resulting LoG mask is sufficiently large to accurately represent the LoG function.  
  
Calculate a numerical measure of the similarity between the DoG and LoG masks using the following commands:  
log=log./max(max(log));  
dog=dog./max(max(dog));  
sqrt(sum(sum((dog-log).^2)))  
  
The above assumes that you have given your LoG mask the variable name "log", that you have given your DoG mask the variable name "dog", and that you have created two masks of the same size. It finds the Euclidean distance between the DoG and LoG masks after they have both been scaled to have the same amplitude.

### **EXERCISE:**

Write a programme that will systematically vary the standard deviations (in steps of 0.10) for the two Gaussians used to generate the DoG mask and use this code to search for the values of these standard deviations which will generate a DoG mask that is as similar as possible (as measured using the above code) to the LoG mask created using a Gaussian with standard deviation 5.0.  
  
Report the two standard deviations you have found:  
smaller standard deviation= Answer  
larger standard deviation= Answer

## **Laplacian Image Pyramid**

A Laplacian image pyramid is a multi-scale representation of a single image that highlights intensity discontinuities at multiple scales. It is obtained by convolving an image with a Gaussian mask and subtracting the smoothed image from the original one. The next level of the pyramid is obtain by repeating this process on the smoothed image after it is down-sampled.  
  
Create a 3 level Laplacian image pyramid of the woods image (after conversion from uint8 to double format), using a Gaussian with a standard deviation of 1.5, using "same" as the shape parameter for the conv2 function, and resizing by a scale factor of 0.5 using nearest-neighbour interpolation. Display the images in the pyramid as subplots in the same window.

### **EXERCISE:**

For the last level of the pyramid report the values at the following locations (correct to at least 2 decimal places):  
row=3, column=6 Answer  
row=1, column=3 Answer

## **Complex Cells (at multiple orientations)**

Combining the outputs of complex cells selective to different orientations can be used to detect edges regardless of phase and orientation.  
  
In this question you should work with the elephant image after you convert it from uint8 format to double format.  
Using the method described, and parameters used, in the week 4 MATLAB exercise on Complex Cells, simulate the output of complex cells at the following orientations [0,15,30,45,60,75,90,105,120,135,150,165] for the elephant image. Combine the outputs of these complex cells by taking the maximum response of all complex cells at each pixel location. Show an image of the result.  
  
You will need the m-file "gabor2.m" to create Gabor filters. This file is available from the module's KEATS webpage. Copy this m-file into the directory where you are executing MATLAB.

### **EXERCISE:**

For the image you have created report the values at the following locations (correct to at least 2 decimal places):  
row=406, column=218 Answer  
row=314, column=142 Answer