Modelling Low-Level Biological Vision with MATLAB

1. Introduction

This laboratory session provides the opportunity to experiment with some simple models of low-level biological image processing. The hours timetabled for this laboratory are insufficient to complete it. You should therefore work on this in your own time, and use the timetabled hours to get help and clarification.

This laboratory is assessed. A type-written report needs to be submitted online through KEATS by the deadline specified on the module's KEATS webpage.

> Boxes like this one indicate instructions that you need to carry out, or commands you need to execute, in MATLAB.
>
> 1.0.1: Boxes like this one describe what is required for your lab. report. Note that it is not necessary to write a formal report (one with an abstract, introduction, methods, results and discussion), you simply need to provide the information requested in boxes like this one. You also do not need to provide your code, except where you are specifically asked to do so.

I suggest you write a MATLAB script file, containing all your work, so that you can easily save your work and return to it at a later time.

2. Redundancy in Natural Images

Natural images exhibit a high degree of redundancy, i.e. the similarity between the intensity of a pixel at location (x, y) and that at a neighboring location (x + o, y) is inversly proportional to the distance (o) between these points. Matlab provides a command `corr2(A,B)` that can be used to calculate the correlation coefficient between the elements of two equally sized matrices (image patches).

> Copy the following two image files to the directory in which you are running MATLAB: rooster.jpg woods.png
>
> Load these images into MATLAB.
>
> Convert these images into greyscale images with the double data type.
>
> Execute the command:
> `corr2(Ia,Ia)`
> Where Ia is a matrix representing an image.
>
> Now perform the same command with two parts of the same image that are shifted one pixel with respect to each other.
>
> Write a simple MATLAB function or script that will automate the process of calculating the correlation coefficient between the same image shifted by different values.
>
> Using this program, calculate (for both images) the correlation coefficient for values for shifts between 0 and 30 pixels
>
> 2.0.1: Plot a graph of shift vs correlation coefficient for both the rooster and the woods image. Include in your report a print out of this figure and a print out of your code. Briefly describe and explain the results you have obtained. [6 marks]

3. Retinal Ganglion Cells Modelled Using DoG Masks

A simple model of the responses of retinal ganglion cells is provided by convolving an image with a radially symmetric Difference of Gaussians (DoG) mask. Off-centre, on surround RF types can be simulated, in addition to on-centre, off-surround types, by using the additive inverse of the DoG mask.

3.1 Redundancy Reduction

> Create a Difference of Gaussians mask by subtracting one Gaussian (with standard deviation 6) from another Gaussian (with standard deviation 2).
>
> Convolve each image with this mask.

Now repeat the process (described in section 2) of calculating the correlation coefficients for an image shifted by different amounts using the convolved images.

Repeat the above with a Difference of Gaussians mask generated by subtracting a Gaussian (with standard deviation 4) from another Gaussian (with standard deviation 0.5).

3.1.1: Plot graphs of shift vs correlation coefficient for both the rooster and the woods image after convolution with both the DoG masks. Include in your report a print out of this figure. Briefly explain why the results for the two DoG masks differ, also explain why the current results differ from those in the previous section. [4 marks]

3.2 Colour Oppoent Cells

Rather than convolving an image with a DoG mask, we can produce an identical result by convolving the image twice with two different Gaussian masks, and taking the difference between these two outputs.

Using this insight, show how colour opponent cells would respond to the rooster image.

Generate an image with 4 subplots, with the subplots showing the response of the following centre-surround colour opponent cell combinations:
1. red-on, green-off
2. green-on, red-off
3. blue-on, yellow-off
4. yellow-on, blue-off

Use a Gaussian with standard deviation 2 for the centre, and a Gaussian with standard deviation 3 for the surround.

You will need to convolve the separate colour channels of the rooster image with Gaussians. Use a command like this:
```
Rg1=conv2(Ia(:,:,1),g1,'same');
```
Assuming Ia is an RGB image (stored using the double precision data type), this will generate a new image that is the convolution of the red channel (channel 1) with the mask g1.

You will also need to find the mean of two coulour channels. To do this use the following command:
```
mean(Ia(:,:,1:2),3)
```
Assuming Ia is an RGB image (stored using the double precision data type), this will generate a new image that is the mean of the red and green channels (i.e a yellow channel).

3.2.1: In your report include a print out of the figure with four subplots that you have just created. Briefly describe and explain the results you have obtained. [4 marks]

Repeat the above using the same standard-deviation for both Gaussians, i.e. use a Gaussian with standard deviation 2 for the centre, and a Gaussian with standard deviation 2 for the surround.

3.2.2: In your report include a print out of the figure with four subplots that you have just created. Briefly describe and explain the results you have obtained. [4 marks]

4. V1 Orientation Selective Cells Modelled Using Gabor Masks

A simple model of response of all V1 simple cells with the same orientation preference is provided by convolving an image with a Gabor mask with the appropriate orientation.

The m-file gabor.m providess a function that will generate a 2D gabor mask. Copy this m-file into the directory where you are executing MATLAB.

Use the commands:
```
help gabor
imagesc(gabor(10,15,90,1,0))
```
(and variations on the numerical values in the latter command) to explore the effects of the different parameter values.

4.1 Simple Cells (at one orientation)

Copy the elephant.png image file to the directory in which you are running MATLAB.

Load this image into MATLAB, and convert to a greyscale image with the double data type. Convolve the elephant image with a Gabor mask generated with the following parameters:

`gabor(4,8,90,0.5,0);`
Use 'valid' as the shape parameter in `conv2`).

Plot an image of the output as `subplot(1,3,1)`

4.2 Complex Cells (at one orientation)

Convolve the elephant image with a second Gabor mask which has a phase of 90, but is otherwise identical to the previous mask.

These two masks form a quadrature pair. If we combine the output from these two masks by taking the L2-norm (i.e. by calculating the square root of the sum of the squared responses of the odd and even symmetry masks at each image location) then the combined output will be invariant to the phase of the input.

Convolve (using 'valid' as the shape parameter) the elephant image with both these masks and combine the two outputs using the L2-norm.

Plot an image of the result as `subplot(1,3,2)`

4.3 Complex Cells (at multiple orientations)

Model the output of complex cells at the following orientations
[0,15,30,45,60,75,90,105,120,135,150,165].

Combine the outputs of these complex cells by taking the maximum response at each pixel location.
This can be done using the following command:

```
max(Ic,[],3)
```

Assuming Ic is a three-dimensional matrix, the third dimension of which represents different orientations.

Plot the maximum response across orientations as `subplot(1,3,3)`

4.3.1: In your report include a print out of the figure with three subplots that you have just created. Include colorbars for each subplot. [6 marks]