Laboratory 9 “Computational Vision”

Laboratory 9: Retrieval of images based on textures

In this practice we will see how we can use the textures for image retrieval. The main issues are:

1) Textures – described by Derivatives of Gaussian (DoG) filters.
2) Descriptors based on image texture and colour. To complete the laboratory it is necessary to know and apply the basic concepts of textures. For this, refer to the material of the theoretical slides.

Texture descriptors extraction

The textures that we will use in these exercises will be the Gaussian filters (look at class transparencies). These can be downloaded from http://www.robots.ox.ac.uk/~vgg/research/texclass/code/makeLMfilters.m. An explanation can be found at:

http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html.

9.1 Download the code to generate a filter bank of the Derivatives of the Gaussian, in other words: the Leung-Malik (LM) filter bank and display it in a figure:

```
function [  ] = testFiltros()

      F=makeLMfilters(); % generate the filters
      visualizeFilters(F);
end

function [ ] = visualizeFilters(F)

      figure, % visualize the  filters
      for k=1:size(F,3);
          subplot(8,6,k);
          imagesc(F(:,:,k)); colorbar;
      end

end
```

Note: Look at the *imagesc* and *colorbar* commands. What do they do? What are the different filters? What values do they have?

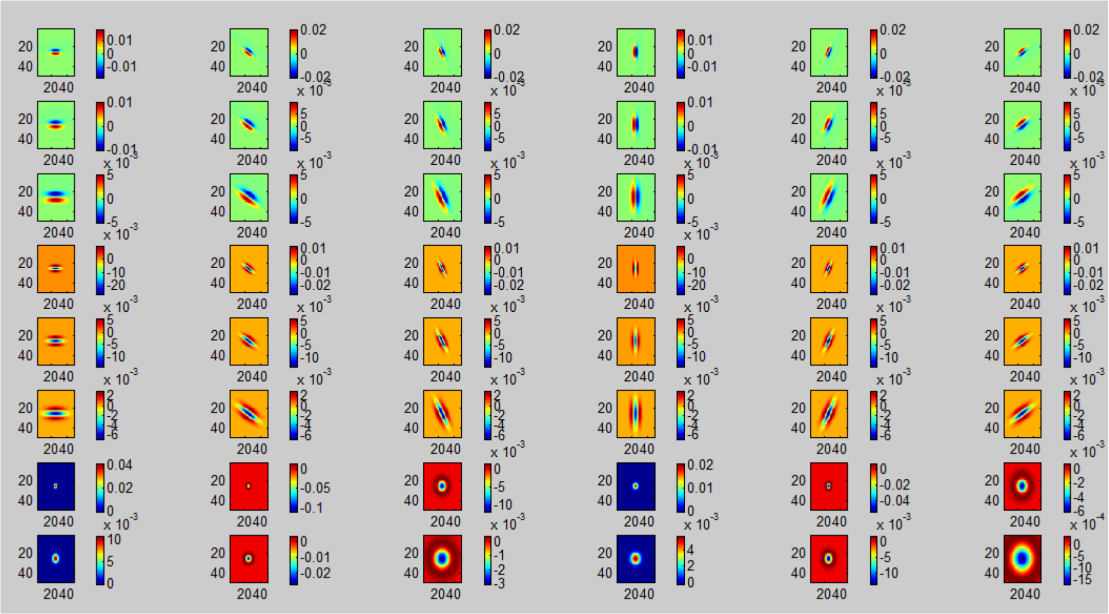

Figure 1. Bank of Gaussian filters

9.2 Implement a function *getFeatures* that for an image builds its texture descriptors, where each element of the descriptor is the average result of the convolution of the image with the filters. How big is the descriptor?

Note 1: When looking at the texture of the image, we will not consider the colour at this stage. Therefore, you must convert the colour image to grayscale.

Note 2: Use *conv2*() instead of *imfilter*() in Matlab to get positive and negative responses. Display the result of the convolution with the filters. Observe which filters have a better response on the image you have chosen and discuss why.

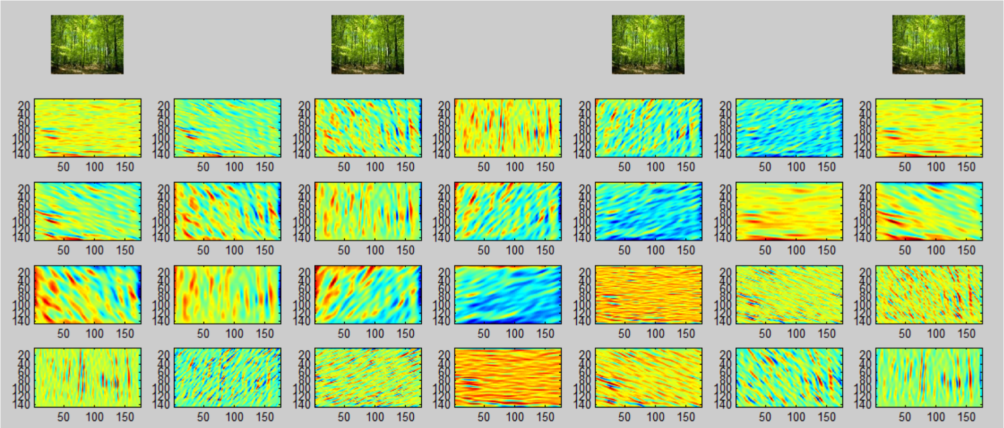

Figure 2. Results of the convolution on the image with different DoG filters.

9.3 Write a *getClassFeatures* function that given a directory and an extension, read all images in the directory with the extension given, calculates their texture descriptors (using *getFeatures* of 9.1.2) and stores them in a matrix where each row corresponds to an image and each column to a feature. Apply this to build three matrices of texture descriptors corresponding to the 3 image directories.

Note: Given a directory in the variable directory (for example: 'textureimages/sunset/'), the following command reads the files that are in that directory:

```
	files=dir(fullfile(directory,'*.jpg'));
	im=imread(fullfile(directory, files(i).name));
```

Note: The way to access the i-th file in the directory will be through files(i).name. Check what makes *fullfile*.

9.4 Write a *visualizeFeatures* function to display in a plot a list of one or several features (eg [25, 41]) for images of the three directories using a different color for each directory (help *plot*). Define another function *visualizeFeatures3D* to display as 3D plots the three groups of images (corresponding to the 3 directories) using three specific features (eg. [25, 38, 41]. What does each axis correspond to in the 2D and 3D visualizations?

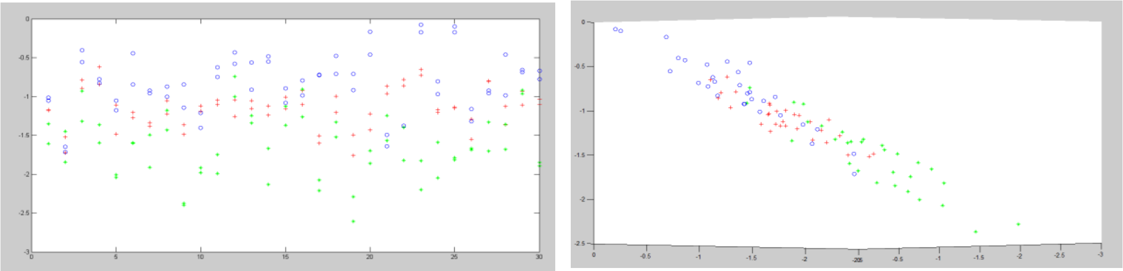

Figure 3. Separation of the 3 kinds of images by texture: features 41 and 25 in the 2D case (left) and features 41, 25 and 38 in 3D (right) case.

9.5 For each image of the three classes, write the function *retrieveKImages* to retrieve and display the k (e.g. k = 9) more similar images as texture descriptors (see Fig. 4).
Note: Use the command *knnsearch*.

Observe the result when the colour (r, g, b) is added to the texture features as additional features per image. What dimension will the feature space have if we add the colour? Observe what are the most discriminative features for each set of images first without using colour and after adding colour features. Discuss your observations on the performance of the algorithm and possible improvements.

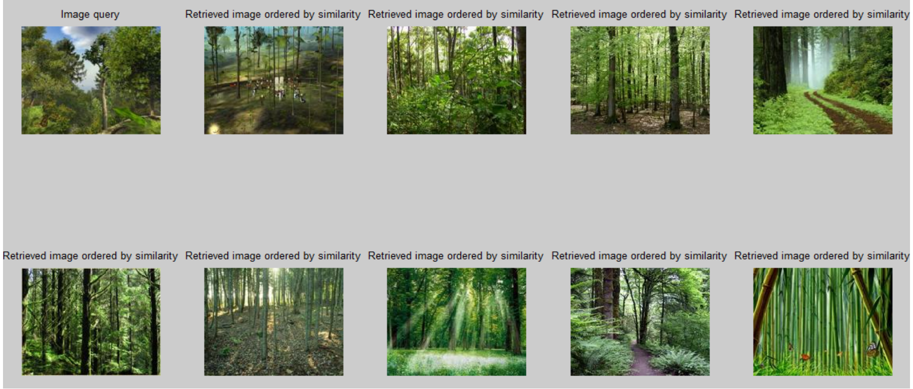
Image query
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity

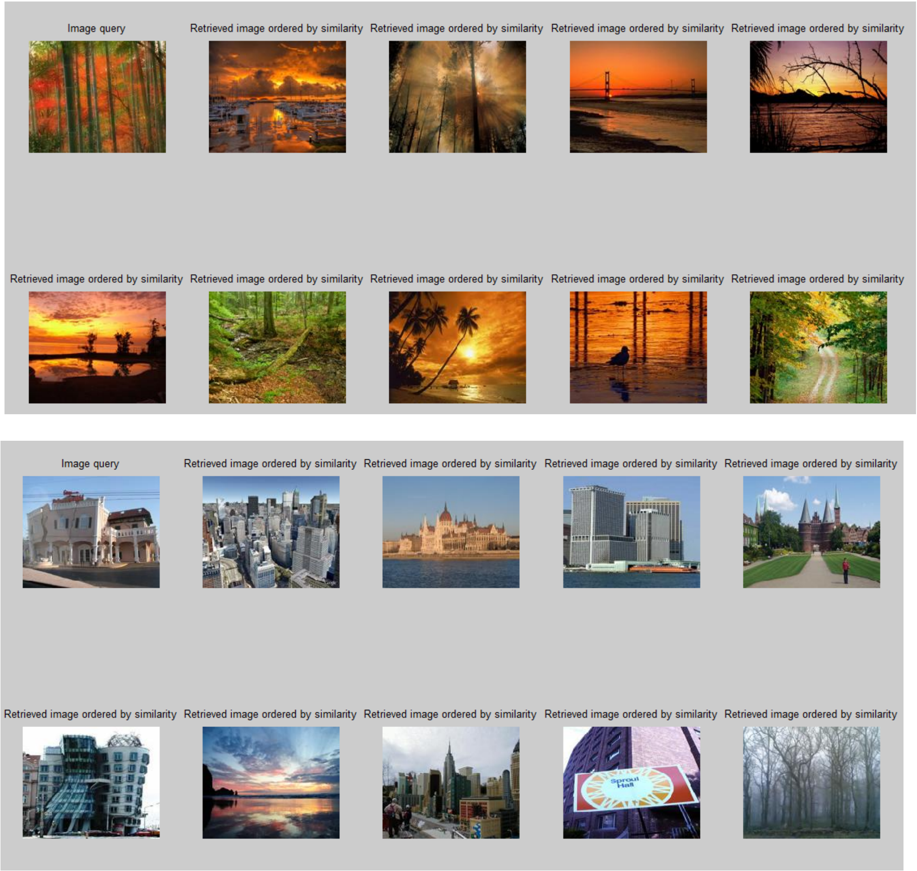
Image query
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Image query
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity
Retrieved image ordered by similarity

Figure 4. Search results by similarity using descriptors based on texture and colour.

Delivery of the laboratory

Deadline: 30 December at 23: 00h.

This part is to be delivered together with the next laboratory. It should be stored in a compressed file named "names_surnames_P9-10.zip" containing: - .m files with the created functions and a pdf file explaining the implementation and the performance of the algorithms and the answers raised in the exercises.