Computational Vision Laboratory

Laboratory 3: Retrieval of images based on texture

In this work, we will see how we can use the textures for image retrieval. The main topics that we need are:

(1) textures - Gaussian filters.
(2) descriptors of images based on texture.

To complete the work, it is necessary to understand and apply the basic concepts of textures. For this, refer to the material of the theory.

Extracting texture descriptors

The texture descriptors that we will use in the first paragraph will be the Gaussian filters. These can be downloaded from http://www.robots.ox.ac.uk/~vgg/research/texclass/code/makeLMfilters.m. A brief explanation can be found in:
http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html.

3.1 Download the code to generate a Bank of filters of the Gaussian (in other words: the Bank of filters of Leung-Malik (LM) and display them in a figure):

```
function [  ] = testFilters()
% Illustrating Gaussian filter bank

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

function [ ] = visualizeFilters(F)
% Visualizing the filters by pseudocolors

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

end
```

Note: What are the commands imagesc() and colorbar?

What are the different filters? What values do?

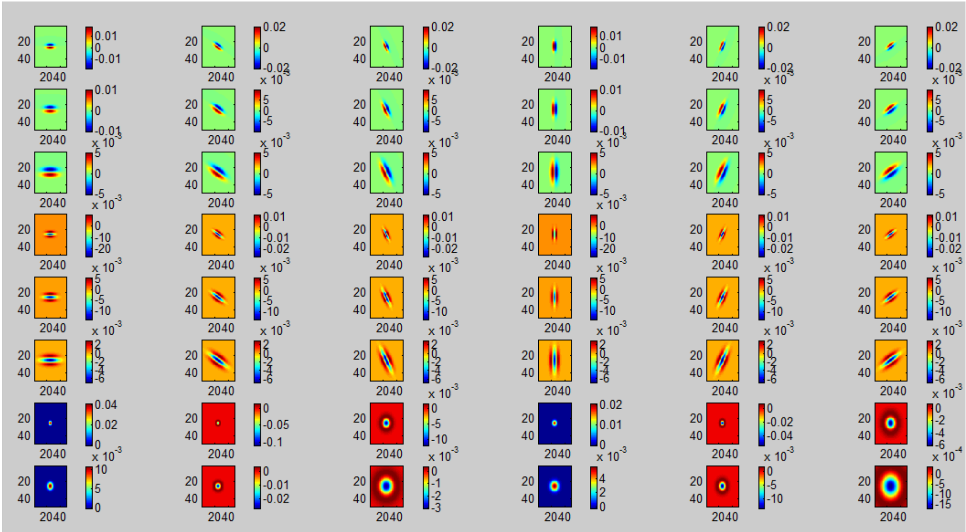

Fig.1. Gaussian filters bank

3.2. Implement a function getFeatures() in Matlab that given an image, builds a descriptor of textures where each element of the descriptor is the average of the result of the convolution of the image with any of the filters. Display the result of the convolution with some filters. See which filters have a better response on the image you have chosen and comment why. What size does the texture descriptor have?

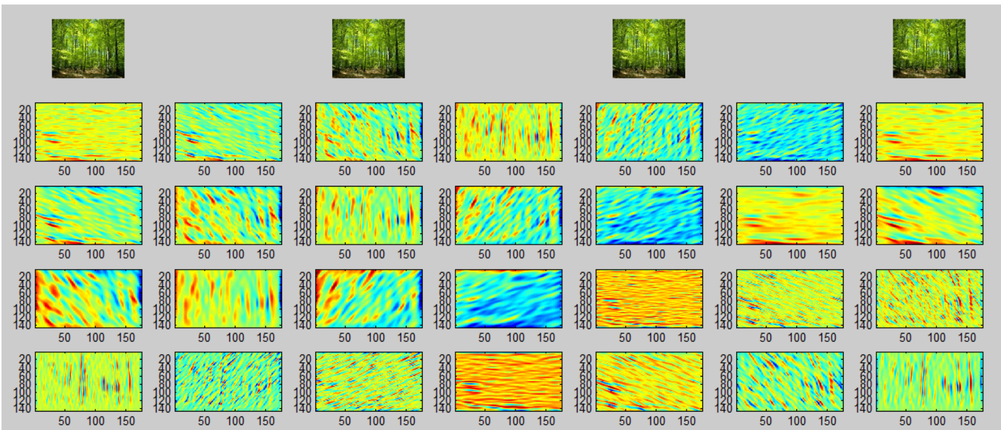

Fig.2 Result of the convolution of the image above (the image is repeated) with different filters

3.3 Write a function getClassFeatures() in Matlab that given a directory and an extension, reads all images in the directory with the specific extension, calculates their texture descriptors (using 3.1) and stores them in an array, where each row corresponds to an image and each column to a feature. Apply this function to build three arrays of descriptors of texture of the 3 kinds of images in the directory textureimages downloaded from the Virtual Campus. Help: given a directory in the variable 'directory' (e.g. ' images/sunset /'), the following command reads the files that exist in this directory:

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

Note: The access to the i-th file will be through files(i).name (check what the command fullfile does).

3.4. For each image in the three classes, write a function retrieveKImages() in Matlab that retrieves and displays the k (e.g. k = 9) more similar images according to their texture descriptors (see Fig. 3). You can use the knnsearch() command in Matlab. Comment the performance of the retrieval by knn and the used descriptors.

(**Optional**): observe how the outcome improves if apart from the texture filters you add color features per each image. What will be the features space dimension if we add color features too? See what are the most discriminative characteristics for each set of images first without using color and then using color. Discuss your observations on the performance of the algorithm and possible improvements.

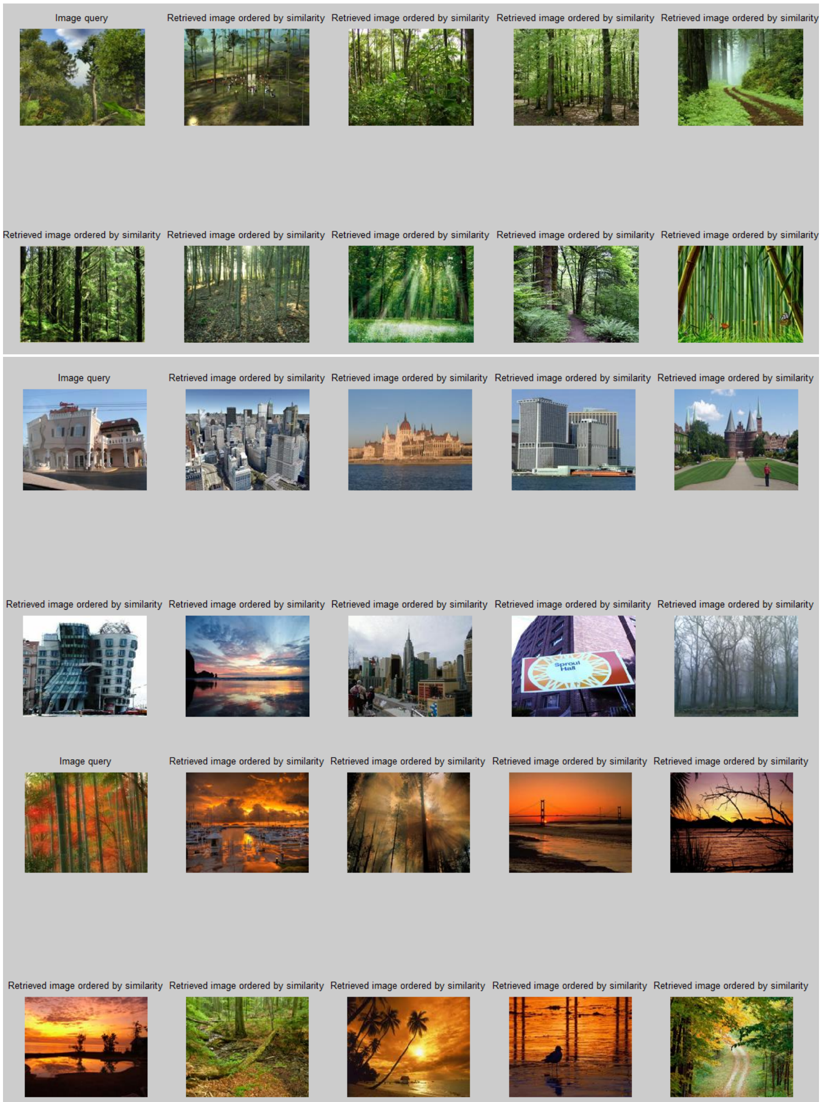

Fig.3 *Search results for similarity using descriptors based on texture and color*

Deadline: November, 20, 2015. This part will be delivered as a zipped file with separate functions corresponding to each exercise (`testFilters()`, `getFeatures()`, `getClassFeatures()`, `retrieveKImages()`) together with part I (6.10.2015) and part II (13.10.2015).