Computer Vision

Exercises of Lab 6

Exercise 6.1: Canny edge detection

The goal of this exercise is to understand the Canny edge detector and implement non-maximum suppression.

Run Exercise6.2.m, read the code and inspect Figure 1 carefully.

We are going to implement non-maximum suppression, constructing thin edge lines in the output image. To do this, construct two gradient vectors (opposite directions) with length 1 for the point (i,j). The x- and y-values should be stored in the vectors, u and v, respectively. Look at slide 33 in Lecture5.1-2_EdgesLines.pdf.

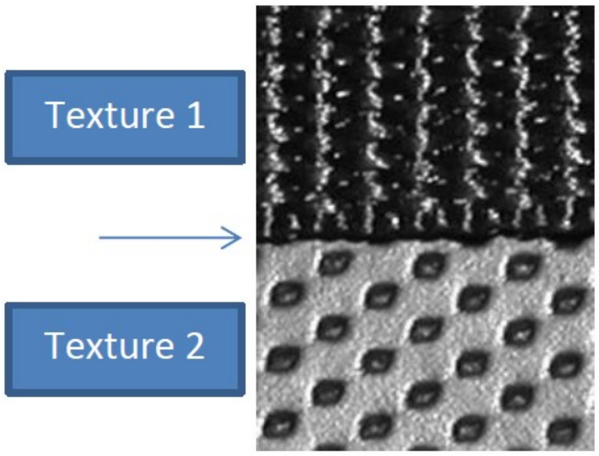
Hint: use the gradient maps, Ix and Iy, and normalize with Inorm.

Then, plot the gradient vectors on top of the gradient map using the quiver function.

Finally, implement the actual edge thinning by assigning EdgeMap(i,j)=1 if the current gradient at (i,j) is larger than or equal to its two interpolated neighbors.

Exercise 6.2: Texture gradients

The goal of this exercise is to detect the edge between two textured areas.



Make sure that you have VLFeat. Open Exercise6.3.m and change the variable VLFEATROOT to the unpacked paths at your hard drive.

Run the Matlab script and study Figure 1 (the program pauses four times – hit any key to proceed). Notice how the differently oriented filters provide significantly different outputs.

Cluster the pixels in the image using the outputs of the 4 different filters. You should construct a variable, datapoints, with 4-dimensional feature vectors and feed them to the clustering algorithm kmeans. You can use VLFeat's implementation vl_kmeans(datapoints,numClust) that takes datapoints and clusters them into numClust clusters.

Following the clustering procedure, construct a texton map, textonMap. Each pixel should be assigned a cluster number as its intensity. Use show_norm_image(textonMap) to visualize it.

Run the remainder of the code and study it carefully. Make sure that you understand exactly what happens in Figure 3 and 4.

Try changing the different parameters of the algorithm and look at their influence. Change the number of clusters, numClust. Change the number and orientations of the directional filters.