Exercise Sheet 02

(Texture-based Image Retrieval)

Please note: The exercises will be neither collected, nor corrected, or graded.

Exercise 1

Calculate the granularity (F_{crs} in the following algorithm description) and provide it together with the histogram of the best sizes (Sbest) for each of the following images, according to the Tamura texture measure:

1. http://www.uvmapper.com/help/checker_large.gif
2. http://oddstuffmagazine.com/wp-content/uploads/2013/12/satellite10.jpg
3. http://earthobservatory.nasa.gov/IOTD/view.php?id=40997

Necessary steps for calculating the granularity ("IEEE Transaction on Systems, Man, and Cybernetics" Volume 8, Edition 6, article "Texture Features Corresponding to Visual Perception", Tamura, et.al.):

Step 1: Take averages at every point over neighborhoods whose sizes are powers of two, e.g., 1 × 1, 2 × 2,… , 32 × 32. The average over the neighborhood of size $2^k \times 2^k$ at the point (x, y) is

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} \frac{f(i,j)}{2^{2k}}$$

where f(i, j) is the gray-level at (i, j).

Step 2: For each point, at each point, take differences between pairs of averages corresponding to pairs of non-overlapping neighborhoods just on opposite sides of the point in both horizontal and vertical orientations. For example, the difference in the horizontal case is

$$E_{k,\mathrm{h}}(x,y) = \left| A_k\left(x + 2^{k-1}, y\right) - A_k\left(x - 2^{k-1}, y\right) \right|$$

Step 3: At each point, pick the best size which gives the highest output value:

$$S_{\mathrm{best}}(x,y) = 2^k$$

where k maximizes E in either direction, i.e.,

$$E_k = E_{\max} = \max\{E_1, E_2, \cdots, E_L\}.$$

Step 4: Finally, take the average of S_{best} over the picture to be a coarseness measure F_{crs}:

$$F_{\mathrm{crs}} = \frac{1}{w \times h} \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} S_{\mathrm{best}}(i,j)$$

where w and h are the effective width and height of the picture, respectively.

Note: Start by transforming the images to gray. Also consider the fact that the provided algorithm, is not specific in the case of k=0, on how to lay the window. There is also an ambiguity case if there are more k values which build the maximum difference for one pixel. Make your decisions in this cases, and try to argue them. Also worth noticing is the fact that there are pixels, at the boundaries of the images, which cannot be integrated in any window of some k values due to the non-overlapping condition.