NEURL-GA 3042.005 – Spring, 2014

Representation and Analysis of Visual Images

Homework 2

Due: 14 April 2014

Your results should be in the form of an executable MATLAB file (typically, the filename should have an extension of `.m`), divided into sections by triple comments (%%%). Any functions you write should be included as separate m-files. Please email your solutions to `eero.simoncelli@nyu.edu`. You'll need a few functions from `www.cns.nyu.edu/~lcv/software.php`. Auxilliary matlab files are in the course homework directory:

`http://www.cns.nyu.edu/~eero/imrep-course/Homework/`

also linked from the course web page.

1. **Gradients.** Matlab provides a function `gradient` that computes the 2D gradient field of an image. The homework directory includes a function, `localOri`, that computes the three components of the orientation tensor introduced in class. We want to use this function to evaluate the quality of the gradient estimates.

   (a) For a set of 30 angles, ranging from 0 to π, generate a sinusoidal image of size 20x20 (using your function `mksine2`) of frequency $pi/3$ radians/pixel at that orientation, compute MATLAB 's gradient, pass this to `localOri`, and extract the estimated orientation in the center of the image. Plot the error in these estimated orientations as a function of the actual angles. How large are the errors? Also, look at the values of the second eigenvalue, divided by the first. What do these indicate?

   (b) Now do the same thing, but compute the gradient using separable filters generated from these two "optimized" filters, as described in class:

   ```
   p = [0.037659 0.249153 0.426375 0.249153 0.037659]
   d = [-0.109604 -0.276691 0.000000 0.276691 0.109604]
   ```

   Plot the error on top of the plot of the errors for matlab's gradient. How much improvement do the "optimized" filters give?

   (c) Try making the same comparison, but for a higher frequency (say, $5\pi/12$), and for a lower frequency (say, $\pi/6$). Where are the errors worst?

2. **Search.** Load the file matchedFilter.mat into MATLAB . It contains: an image to be searched (scene), and a small target image (target), and a binary mask (mask).

   a. **Matched filter.** Find the location of the target, to the nearest pixel, using matched filtering. Specifically, convolve the scene with the (vertically and horizontally flipped) target. For this problem, we do not know the amplitude of the target, and there are also many high-amplitude "distractors" that will produce large filter responses, so you'll need to divide the filtered result, pointwise, by an image containing the L2 norm (sqrt of the sum of squares) of the masked region. Then find the location of the maximum. Construct a blank image, of the same size as the large one, and place the target into it at the found location. Subtract from the original large image, and look at the error image. Did you succeed? If not, what went wrong?

b. **Taylor correction.** Refine your answer to the previous part, by adjusting the subpixel position of the target using differential approximations (i.e., a first order Taylor series). Specifically, compute the gradient of the image. Then solve for the combination of the region found in the previous part, plus the two derivatives from that region that, when multiplied by the mask, best approximates the target. With this correction included, what is your estimate of the target location?