Problem Set 4 - Hints

November 10, 2016

1 DoG, LoG and Zero Crossings

- Zero-crossings of the second derivative are more robust than small-size gradient detectors and can be calculated as a Laplacian of Gaussians (LoG) or as a difference of Gaussians (DoG)
- Read through text book section 5.3.3 (3rd edition) on *Zero-crossings of the second derivative*
- You can use Python scipy's convolve2d to perform convolution of the image and give filters
- Avoid using OpenCV convolve fuction as it scales the values between 0 and 255. You will not be able to find zero crossings
- A simple detector may identify a zero crossing in a moving 2 x 2 window, assigning an edge label to any one corner pixel, say the upper left, if LoG/DoG image values of both polarities occur in the 2 x 2 window
- No edge label would be given if values within the window are either all positive or all negative. Read through textbook section for more insight on additional criteria - first derivative support

2 Region Merging Segmentation

2.1 Region Merging via Boundary melting

- Get familiar with crack edge image data structure - You need to double the size of the image and substitute the extra rows and column with the magnitude of difference of adjacent pixels of the original image (Thus it is called a crack edge, which retains the edge strength)
- Crack edge image will contain the gradient information of 4 or 8 neighbors of entire image. stick to 4 neighbor for simplicity.
- Set a threshold T1 and make all the crack edge pixels below that value to zero
- Finding optimal methods for common boundary detection, region perimeter computation and threshold selection will be an intellectual exercise!

- Lets say we decided T2 to be 0.5 . In the first iteration every pixel is considered as a separate region. The length of common boundary between pixels is 1. If there is a weak edge between the pixels, the value of $\frac{W}{min(l_i,l_j)}$ will be 1 and both the pixels will be merged. If there is a strong edge, it will be zero and no merge will happen

- You keep track of all these unique regions and recursively merge them till you cannot do anymore.

- $\frac{W}{l}$ criterion removes the edge if the common boundary of length l has many weak edges. Say T3 is set to be 0.5, this criterion will delete an edge if more than half of the crack edge pixels are zero in the common boundary

- The difference between T2 and T3 is that T2 considers the region perimeter and T3 only considers common boundary while doing merging. Can you think of a valid reason why we need two of these thresholds?