Homework #3 B Image Statistics

Due Wednesday, April 14\textsuperscript{th}, 2010.

Professor Davi Geiger

Extracting image statistics using ground truth data:

You have now for each image a set of say 5 contours (I will provide 4 contours from the other students and each student uses its own contour as well. If you are one of the 4 selected contours, I will provide you a fifth one):

For each image do the following (item 4 will repeat the first three items for different scales, so when you program this, try to make one program where the scale $s$ can be varied):

1. For each pixel in the image, extract the largest magnitude, $| D_\rho \hat{I}(p_\rho, \theta, \frac{\pi}{2}, s = 3)|$, among all directions. Separate these pixels into two sets, the pixels on the given contours (color red) (there are five given contours) and the pixels outside (let us give blue color to them). Plot a histogram for the “red” pixels and for the “blue” pixels. Plot both histograms in the same graph. More precisely, this histogram has the “x-axis” as the possible values of the magnitude of the derivative (so $x$ vary from 0 to 255) and the “y-axis” gives how many pixels such a magnitude was found. As Quenton suggested we should divide this number by the total number of pixels in the respective area, blue and red. In this way the histogram is normalized, i.e., the area of the histogram will be 1. The two histograms, for red pixels and blue pixels are to be shown together as a red curve and a blue curve. A possible graph histogram is shown below.

![Histogram example](image.png)

Number of pixels/total pixels (area)

Display one such histogram per image.

Also display the same histogram but for the function

$$g(x,y) = \frac{1}{I(x, y)+1} \max_\theta | D_\rho \hat{I}(p_\rho, \theta, \frac{\pi}{2}, s = 3)|$$

Computing the max $\theta$ for the magnitude of the derivative is already done, so one just needs to divide by the grey value of the pixel. Now the values vary from 0 to 255, but have continuous values (for each integer value of contrast one can vary $I(x,y)$ and obtain fractions of integer values). So one may create bins separated by intervals of 0.25, i.e., have 4 times more bins. If we use the floor function, then $g(x,y)=3.3$ goes to the bin 3.25. If $g(x,y)=3.74$, it goes to bin 3.5. Most likely the first half of the bins will be filled but not the second half. One can even
create integer bins of space 4 after the value 128, i.e., a bin for 128, a bin for 132, 136, ..., up to 252. So if \( g(x,y) = 139 \), it goes to bin 136. Again, make sure to normalize the histogram (divide the count by the total count). Note that the total count of each area is the same for all these cases. These histograms will clarify if one should take into account the grey value, if it will provide a better separation between red and blue curve.

We now examine the angle along the contour.

2. For each given contour pixel (you have five contours on each image) extract the true angle \( \theta_{\text{True}} \). Each true angle \( \theta_{\text{True}} \) on the contour is the angle that brings a contour pixel to its neighbor contour pixel. Exclude the \( \theta_{\text{True}} \) associated to the last contour pixel from these computations since it does not have a next neighbor.

Now, consider the four angles \( 0, \pi/4, \pi/2, 3\pi/4 \) and assume the other four angles \( 0=\pi, 5\pi/4, 3\pi/2, 7\pi/4 \) have the same magnitude \( D_{\rho} \hat{I}(p_{\rho}, \theta_{\text{true}} + \frac{\pi}{2}, s = 3) \) (i.e., adding \( \pi \) does not change the magnitude). For each contour pixel verify the ranking of the true contour angle \( \theta_{\text{True}} \) as it can be ranked 1,2,3,4. The rank of \( \theta_{\text{True}} \) is based on the rank of the four quantities \( D_{\rho} \hat{I}(p_{\rho}, \theta_{\text{true}} + \frac{\pi}{2}, s = 3) \), \( D_{\rho} \hat{I}(p_{\rho}, \frac{\pi}{4} + \frac{\pi}{2}, s = 3) \), \( D_{\rho} \hat{I}(p_{\rho}, \frac{3\pi}{4} + \frac{\pi}{2}, s = 3) \), \( D_{\rho} \hat{I}(p_{\rho}, \frac{\pi}{2} + \frac{\pi}{2}, s = 3) \) as 1,2,3,4 where 1 is the largest one. The quantity \( D_{\rho} \hat{I}(p_{\rho}, \theta_{\text{true}} + \frac{\pi}{2}, s = 3) \) will then fall somewhere in this rank. For each contour pixel record the rank and create a histogram, where the “x-axis” is the rank, 1,2,3,4 (only four values) and the “y-axis” is the number of pixels the magnitude of the true contour contrast along the true angle was ranked accordingly. We hope that most pixels will have the true angle to be ranked 1, but this should not be necessarily true for all pixels. Let us verify this histogram. Display one such histogram per image.

\[
\begin{align*}
n(i) & \text{- Number of contour pixels} \\
1 & \quad 2 \quad 3 \quad 4 \quad \text{rank of the true value } |D_{\rho} \hat{I}(p_{\rho}, \theta_{\text{true}} + \frac{\pi}{2}, s = 3)|
\end{align*}
\]

3. For the four ranks above, \( i=1,2,3,4 \) return the average entropy \( H(i) \), where

\[
f_{s=3}(p_{\rho}, \theta) = e^{-1 \left[ |D_{\rho} \hat{I}(p_{\rho}, \theta_{\text{true}} + \frac{\pi}{2}, s = 3)| + 1 \right]}
\]

\( n(i) \) – number of contour pixels ranked \( i \) (from the histogram above)
\[ P(i, \theta) = \frac{1}{n(i)} \sum_{\text{pixels ranked } i} \frac{f_i(p_r, \theta)}{f_3(p_r, 0) + f_3(p_r, \frac{\pi}{4}) + f_3(p_r, \frac{\pi}{2}) + f_3(p_r, \frac{3\pi}{4})} \]

\[ H(i) = -\left[ P(i, 0) \log P(i, 0) + P(i, \frac{\pi}{4}) \log P(i, \frac{\pi}{4}) + P(i, \frac{\pi}{2}) \log P(i, \frac{\pi}{2}) + P(i, \frac{3\pi}{4}) \log P(i, \frac{3\pi}{4}) \right] \]

4. Repeat 1,2,3 for the scales s=5, 7.