Homework #2 (Part B): Multi Scale Contour-Detection

Due Thursday, March 4\textsuperscript{th}, 2010.

Professor Davi Geiger

A Multiscale solution:

Construction of the coarser grid data:
Say we replace each 2 by 2 pixels by one pixel, and store for each direction the highest contrast among the four pixels as “the contrast” of this direction for the coarser grid pixel. In this way, the new coarser grid (smaller grid) will have for each pixel eight responses to contrast at eight angles (actually only four different magnitude values since increments by $\pi$ have the same magnitude) and preserve the high contrast information from the original grid.

Contour solution at the coarser grid:
Dynamic programming as in homework-2, part A.

How to bring this solution back to the finer scale grid:
Once the dynamic programming solution is obtained, at the coarser grid, we would need to bring this solution to the larger-original grid, effectively creating the final solution. How to bring a contour solution from the coarser grid to the finer (and larger) grid?

One needs a strategy to convert each contour pixel and orientation from the smaller (and coarser) grid into contour pixels in the finer grid. Let us refer to the neighbors of a pixel $P_i$ as 1 to 8 as follow:

\[ \text{PATH}_P = \{a, (a,b), (a,c), (a,d), (a,b,c), (a,d,c), b, (b,c), (b,d), (b,a), (b,a,d), (b,c,d), c, (c,d), (c,b), (c,a), (c,b,a), (c,d,a), d, (d,a), (d,b), (d,c), (d,a,b), (d,c,b) \} \]

Note that a path such as $(a,c)$ is different from $(c,a)$ since a successor pixel to each of these paths is different, i.e., this different order will influence the remaining of the path.

We want to assure that the contour is a sequence of contiguous pixels, i.e., for each contour element, there is always another one within the 8 pixels neighbors. Our strategy constructs the path at the finer grid, step by step, in a greedy way. More precisely, let us focus on a contour point $P_i$ at the coarse scale, shown in blue in figure 1. There are in total the following twenty four (24) different possible path that could be the final solution at the finer grid: $\text{PATH}_P = \{a, (a,b), (a,c), (a,d), (a,b,c), (a,d,c), b, (b,c), (b,d), (b,a), (b,a,d), (b,c,d), c, (c,d), (c,b), (c,a), (c,b,a), (c,d,a), d, (d,a), (d,b), (d,c), (d,a,b), (d,c,b) \}$. Note that a path such as $(a,c)$ is different from $(c,a)$ since a successor pixel to each of these paths is different, i.e., this different order will influence the remaining of the path.
Let us say we have already resolved the best path at the fine scale up to \( P_{t-1} \), the predecessor of \( P_t \). The last node (pixel and angle) of the fine scale solution up to the predecessor \( P_{t-1} \) must point to one of the four fine scale pixel at \( P_t \), either \( a \), \( b \), \( c \), or \( d \), as show in figure 1. To be more precise, we may refer to them as \( P_{t,a} \), \( P_{t,b} \), \( P_{t,c} \), or \( P_{t,d} \) to distinguish from the other pixels such as the four pixels \( P_{t-1,a} \), \( P_{t-1,b} \), \( P_{t-1,c} \), or \( P_{t-1,d} \), for example. Moreover, the best fine scale path in \( P_t \) must end pointing to one of the four pixels \( P_{t+1,a} \), \( P_{t+1,b} \), \( P_{t+1,c} \), or \( P_{t+1,d} \) where \( P_{t+1} \) is the successor of \( P_t \).

Thus the set of possible path in \( P \) from \( \text{PATH}_P \) must be limited to these two constraints. Then, one can evaluate among the paths from the set \( \text{PATH}_P \) that satisfy these constraints, which one is the best one (minimum cost) according to the average of the sum of the costs

\[
E(v, u) = f_s(v) + \gamma |\theta_v - \theta_u|_{\text{circular}}
\]

over each of the paths. We use as starting point the last element from the fine scale path at \( P_{t-1} \). This is a greedy method to estimate best path.

This defines a strategy to solve a problem at a coarser grid and bring the solution to the finer grid. One can then define a multiscale algorithm by creating a (small) grid several scales coarser, solving at this large scale using the dynamic programming solution and then bringing the solution down to the finer scale, scale by scale, with the strategy above, until reaching the fine scale.

A. Homework, test this strategy using the same images and try for two scales and then for three scales (including the original image as the fine scale). Set \( T \) constant, for example \( T=255 \) (or \( T=1 \)), and consider varying only \( \gamma \).

B. Write a pseudo-Code for the multiscale algorithm (where one parameter input is the number of scales)