**HW#4 (Watershed segmentation)**

* Implement the simplified Vincent-Soille marker-based watershed segmentation algorithm.
  + The basic algorithm involves three steps:  (1) Compute the magnitude of the image gradient, quantized; (2) Construct a data structure allowing fast access to all the pixels with a certain value; (3) Apply breadth-first search to flood the pixels one gray level at a time, starting with the minimum value, assigning each pixel to either the nearest existing catchment basin or to a new catchment basin.
  + The algorithm is considered "simplified" because it does not explicitly use dams. Although dams are often included in descriptions of the Watershed algorithm, they are unnecessary, having almost no effect on the final result.
  + Define the watershed (boundary) pixels as those which occur at a transition between basins.
  + Your code should include the marker-based modifications to reduce oversegmentation.
  + \*Unlike previous assignments, care should be taken to ensure that your algorithm properly handles the pixels along the image order, so that objects touching the image border are segmented correctly.
* In separate figure windows, display the result of the algorithm at the various stages of the computation as shown in the slide in class (threshold, chamfer, initial watershed, edges of initial watershed, gradient magnitude, markers, and final result).
* The grader will test your code on the images:  [holes.pgm](https://cecas.clemson.edu/~stb/ece847/fall2012/holes.pgm) and [cells\_small.pgm](https://cecas.clemson.edu/~stb/ece847/fall2012/cells_small.pgm).  Due to the difficulty of thresholding these images, it is okay for your code to have a command-line switch to select the appropriate parameter(s).
* For this assignment, you may not use any Blepo code in Watershed.cpp.
* No report is due for this assignment.