1.k-means threshold selection

Input: rice.png from MATLAB

a) k-means threshold selection

take all image corners as the initial background and the rest as the initial object with the corner size of 10

calculate the mean values of both regions

estimate the threshold of two clusters by the mean of the two mean values

cluster the pixels according to the estimated threshold

iteratively calculate the mean values, cluster the pixels and estimate the threshold

 stop the loop when the new threshold doesn't change by more than 1

b) display

display 2 axes in one figure, including the original image and the thresholded image

show the final threshold and the number of loops in the axes title

2.livewire

input: pout.tif from MATLAB

a) gradient approximation

approximate the gradient magnitude G by Sobel filter

b) livewire

take two pixel locations as inputs, one as the start point and one as the end point

calculate the local edge weight between neighbor pixels (p, q): $f(p, q) = (\max(G) - G(q))/(\max(G) - \min(G)) \cdot Distance(p, q)/\sqrt{2}$

search the path on the 8-connected neighborhood per pixel

find the path between the selected two pixel locations with the minimum sum of edge weights using the Dijkstra's algorithm.

c) display

display the results in 2 different figures

display 4 axes in the first figure, including the original image, the gradient image, the image showing the start and end points, and the image showing the result path

display 4 axes in the second figure, including the image showing the result path, the distance map corresponding to the start point, the binary image with the visited pixels as 1, and the image showing the index map of previous visited pixels

d) Notes for the Dijkstra's algorithm

implement: http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm#Pseudocode

use images to store the help variables of the algorithm: including dist, visited, previous, Q