Readme of Salient Object Segmentation code

This code is used to perform ranked salient object segmentation given a color image.

The components included are:

1. MRF segmentation (graph-cut or several other optimization algorithms)
2. Composition cost code for segment (given image and mask)

###### Composition cost code

Converted from ICCV/CVPR “Salient object detection by composition” code and modified to deal with arbitrary segment composition instead of window.

###### MRF segmentation

Based on code from PAMI08 (R.Szeliski et.al) which contains different MRF optimization algorithm to call.

Major concerns:

1. Seed generation: compute foreground weight map by computing shortest path cost to image boundary for patches and compute map for superpixels (average over pixels and normalize). Superpixels whose weight is above certain threshold (e.g. 0.9) are successively used as seed to perform MRF optimization. (note: currently, composition cost computation and seed generation share same superpixel segmentation results)
2. MRF energy function contains two terms: data term and smooth term, E\_D + lamda\*E\_S. (fg:1 bg:0)
   1. Data term: based on superpixel foreground weight map; currently may cause negative energy due to bias and lead to –INF after optimization
      1. 0 l is 1, x is seed
      2. inf l is 0, x is seed
      3. 0 l is 0, x isn’t seed
      4. –( log(fg(x)/(1-fg(x)) + bias ) l is 1, x isn’t seed
   2. Smooth term: based color affinity between adjacent pixels; this term is always non-negative
      1. 0 l(x\_i) == l(x\_j)
      2. –log(|C(x\_i)-C(x\_j)|)\*tradeoff l(x\_i) != l(x\_j)
   3. Parameters: bias and tradeoff. They are selected according to some criterion (not very sure what now), currently use values from ECCV10 proposal paper; this issue may not be very important since we are generating pool of segments as object candidates
3. After computing CC score for all candidates, we do post-processing to filter overlapped segments with lower scores. This procedure follows window-like non-maximum suppression based on mask intersection/union criterion.