Assignment

Liu Hongkun

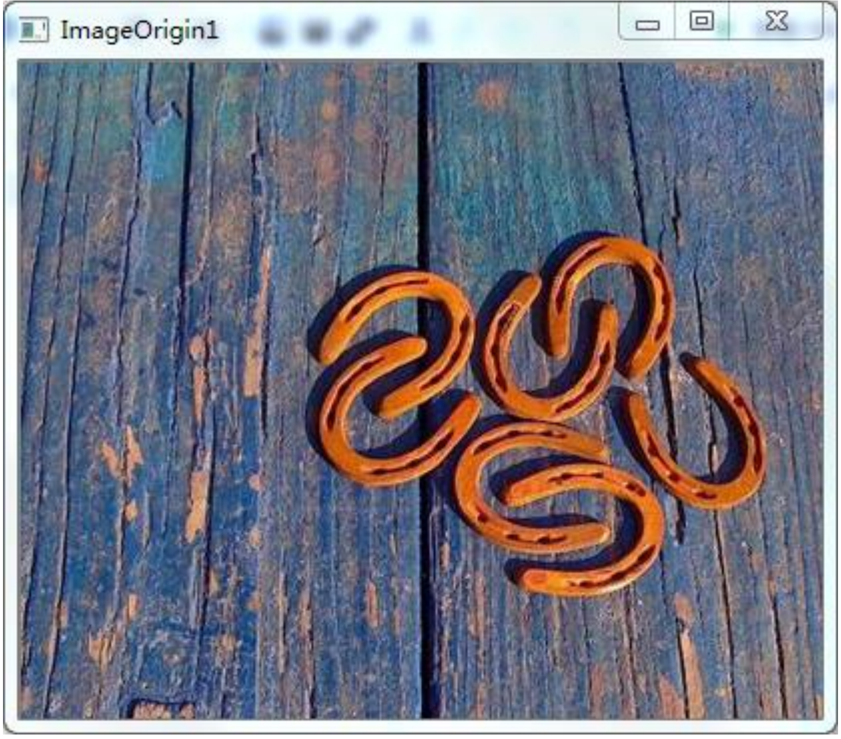
Apil 16,2015

Step 1-1 Extract image information and Segment the input image to superpixels

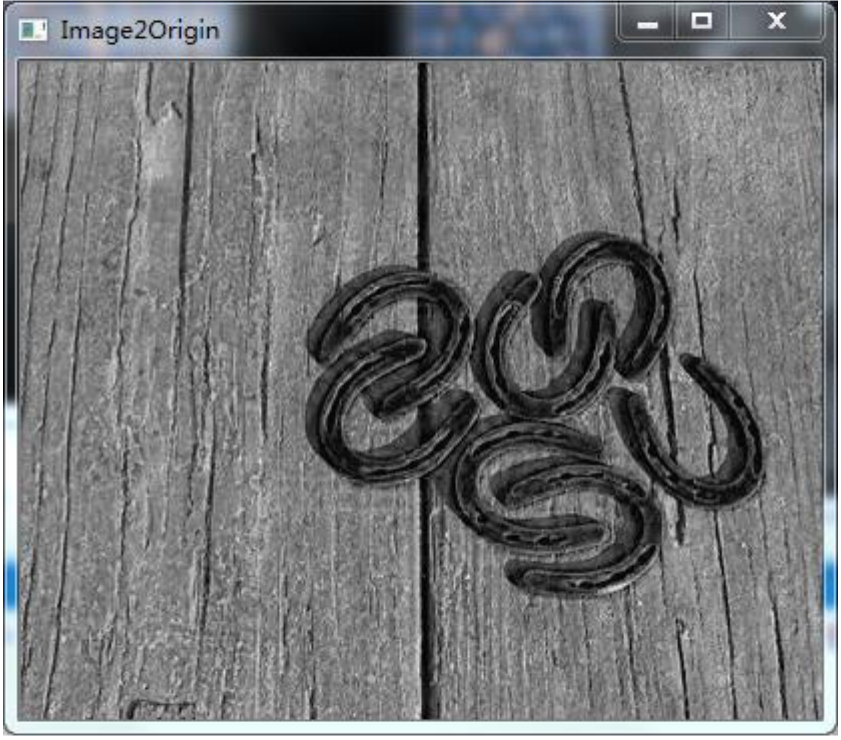
Input: The input color image (m*n*3 matrix), m is the width of the input image, n is the height of the input image.
Output: m*n matrix of color information, m*n matrix of superpixel segmentation matrix.

Read the image to the Mat variable img, then extract the color information of the origin image, use the function InitColExtrMatrix quantizing each color channel, create a one channel matrix saving the color information using the numbers between 0 and 4095.

The origin image:



Quantized image:



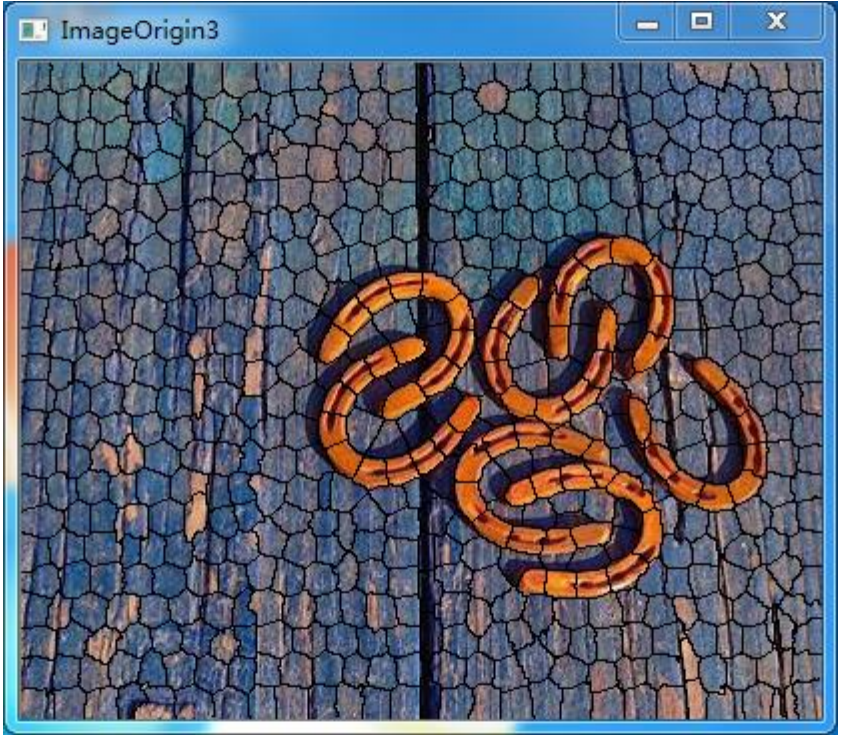
Step 1-2: Segment the input image to superpixels

Input: The input color image (m*n*3 matrix), m is the width of the input image, n is the length of the input image.

Output: Superpixel segmentation matrix (m*n matrix), the value of the pixel in this matrix is just a label to indicate the superpixel it belongs to, and pixels of the same superpixel are labeled the same value.

Create a Mat variable SupSegmatrix whose height and width is as same as the origin image’s height and width, save the super segmentation label values of each pixel.

The superpixel segmentation image:



Step 2: Compute features of each superpixel

Input: Image information matrix, superpixel segmentation matrix.
Output: h histograms, h is the number of superpixels after segmentation.

Vector<Mat> HistTemp whose size is the number of labels, saves the quantized color information, in which HistTemp[i] saving the quantized color information of the pixels with the label i. The vector<Mat> HistSave saves the histograms of HistTemp.

Step 3: Compute superpixel feature contrast

Input: h histograms.
Output: h values, each value indicates the global feature contrast of a superpixel.

Compute the sums of the histogram distance between the histogram with label value i and other label values, save the values as the global regional contrast in the array HistValue.

Step 4: Convert superpixel saliency to pixel saliency

Input: h values, superpixel segmentation matrix.
Output: an initial saliency map (m*n matrix).

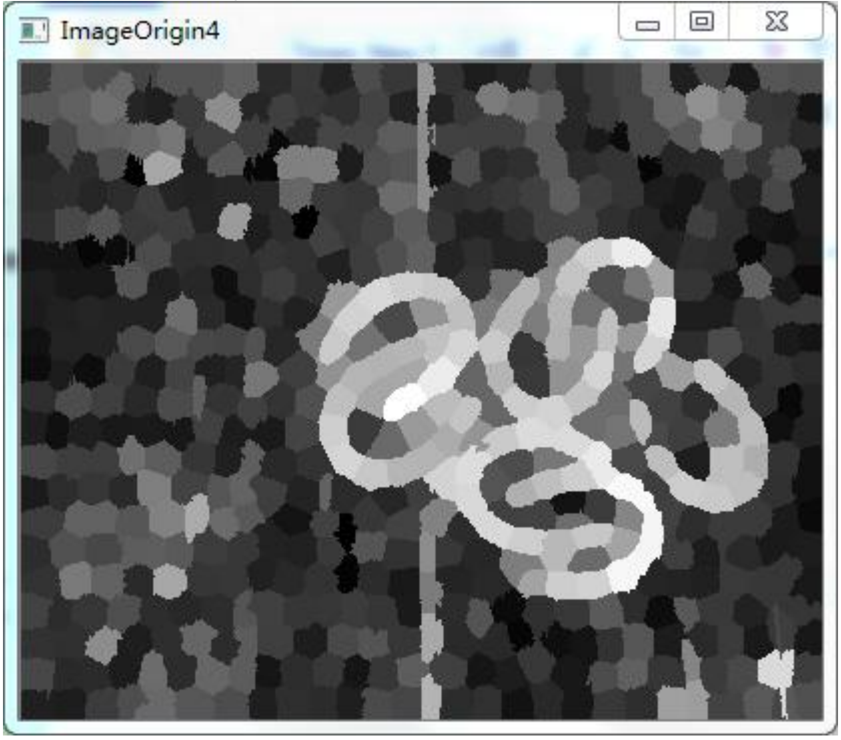
Assign all the pixels of the same superpixel the same saliency value.

Step 5: Use priors to enhance the result

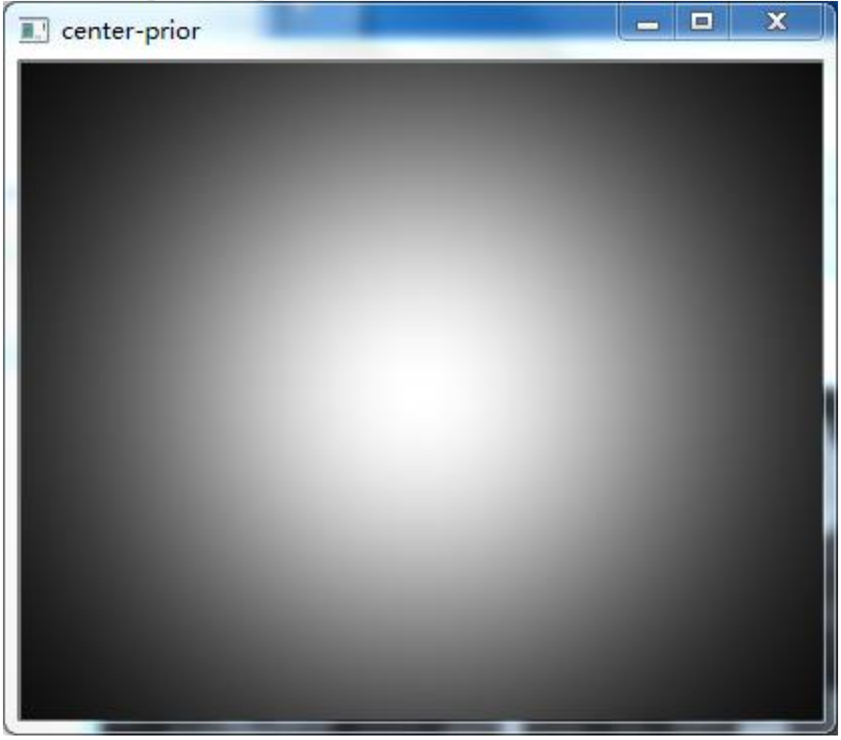
Input: Initial saliency map.
Output: Final saliency map.

Multiple the initial saliency map with the center prior, obtain the final saliency map.

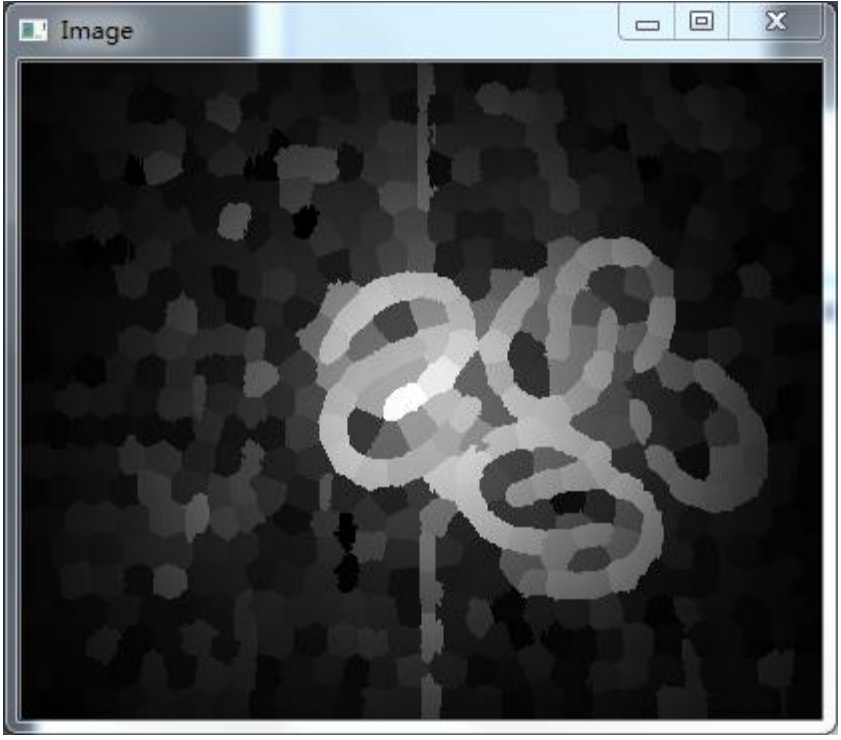
The initial saliency map:



The center prior map:



The final saliency map:



Tip: You could enter the main program from thetest_slic.cpp.