CV2015Spring—Assignment #1

Due: Apr 12, 2015 (10:00AM)

Dai Jialun

Apr 12, 2015

My assignment is programmed by C++ based OpenCV 3.0 and the IDE is Code::Blocks.

1. The Arrange of This Assigment

1.1 Input image

I pick one image shown in Figure 1 from the dataset for test.

Figure 1: Origin Image

1.2 Step 1-1: Extract image information

In this step, I choose color information to extract image information and I quantize each color channel (RGB) to reduce the number, then the number of color is reduced to $16 \times 16 \times 16 = 4096$ of colors. The quantized image is shown in Figure 2.

Figure 2: Quantizied Image

1.3 Step 1-2: Segment the input image to superpixels

In this step, the input color image is the source image. And I adopt the "SuperpixelSEEDS" class in the "ximgproc" module. This "ximgproc" module is built by the Opencv_Contrib, which owns the the extra modules compared with Opencv 3.0. The functions in the "SuperpixelSEEDS" class can complete the superpixel segmentation quite well and quickly. The result after superpixel segmentation is shown in Figure 3.

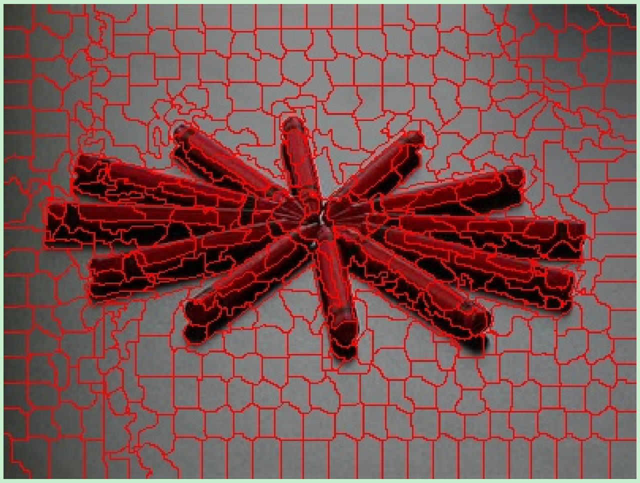

Figure 3: Superpixel Segmentation

1.4 Step 2: Compute features of each superpixel

From the step 1-2, I have known the number of superpixel is 192. And I compute color histogram feature of each superpixel and I store 192 histograms. In this step, the "calcHist" function can easily calculate the color histogram of each superpixel.

1.5 Step 3: Compute superpixel feature contrast

I have obtained 192 histograms in step 2. In this step, I get 192 values stored in a array and each value shows the global contrast of each superpixel. The "compareHist" function is used to calculate the distance of two histograms and it returns a value of "double" type. A parameter in the "compareHist" function can make it execut, based on the formulation for histogram distance, which is as follows:

$$\chi^2(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{b} \frac{2(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}}$$

1.6 Step 4: Convert superpixel saliency to pixel saliency

In this step, I assign 192 values, whose each value indicates a saliency value in a superpixel, to all 192 superpixels in a blank image. Then the image assigned values is exactly a initial saliency image. The result is displayed in Figure 4.

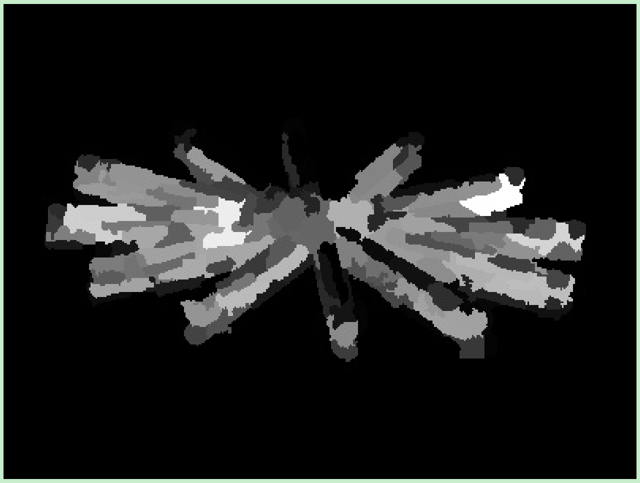

Figure 4: Initial Saliency

1.7 Step 5:The priors

I choose center prior as priors to enhance the consequence of initial saliency. The image of center prior is displayed in Figuire 5.

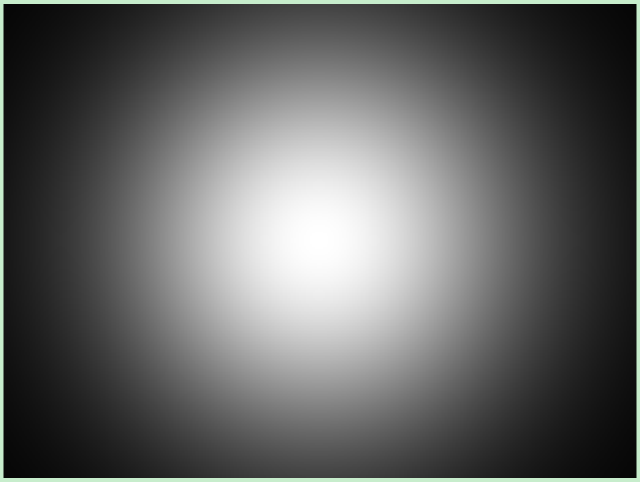

Figure 5: Center Prior

1.8 Step 6: Use priors to enhance the result

Multiply the initial saliency map with the center prior, I obtain the final saliency map. The final saliency map is shown in the Figure 6.

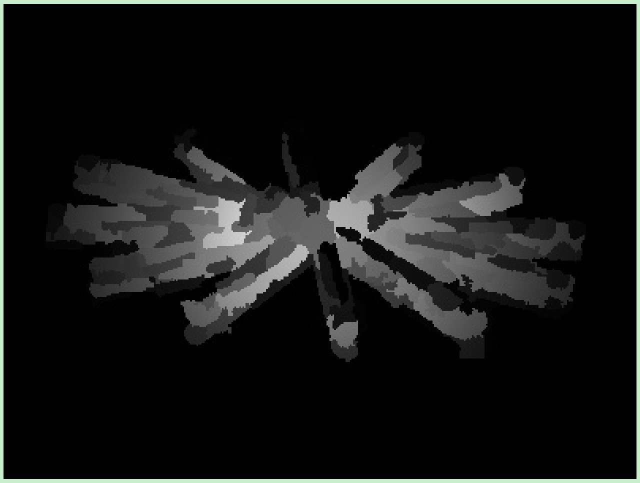

Figure 6: Final Saliency