Report of Assignment#1

Wang Jinpeng

April 17, 2015

In this assignment, I used matlab to complete it. But the result is unsatisfactory, I have tried lots of ways to improve it. However, it appears to have little effect. The procedure and effect after modifications are as follows.

1. Strategy

The original image is shown in Figure 1.

Figure 1: original image.

1.1 Extract image information

Here I choose to extract color information. First, using function *imread* () to input the image you want to process. Then quantize each color channel from 0~255 to 0~15. Compress pixel value by means of dividing each pixel by 16, for example：

```
rgb(:,:,1)=image(:,:,1)./16;    %R color channel.
rgb(:,:,2)=image(:,:,2)./16;    %G color channel.
rgb(:,:,3)=image(:,:,3)./16;    %B color channel.
```

After this, translate the three value of each pixel to a number (range:1~4096).

$$value = rgb(x,y,1) + rgb(x,y,2) \times 16 + rgb(x,y,3) \times 16^2 \ .$$

Finally, get a matrix (m×n), it's each value range from 1 to 4096.

1.2 Segment the input image to superpixels

Here I used the VLFeat open source library. First, transform the imput image from RGB to LAB.

Function: $im\_lab \ = \ vl\_xyz2lab \ (\ vl\_rgb2xyz(image) \ )$

Second, use SLIC algorithm segment the image.

Function:$segments = vl\_slic \ (im\_single, region\_size, regularizer)$

The result image is shown in Figure 2.

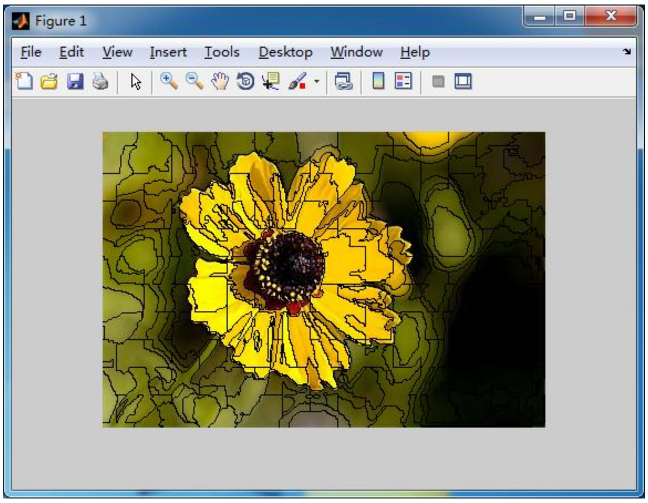

Figure 2: segements image.

1.3 Construct a histogram of these segments.

First, according to segments (get form part 1.2), classify the pixels get from part 1.1. Then use function: *hist* to construct histograms for each segments and store the each result (use a matrix to store them).

Founction: *store_hist(:,m)=hist(double(seg_store(1:seg_count(m),m)),hist_size).*

1.4 Compute superpixel feature contrast

Compute the distances between eachhistogram and others. Here I have done some modifications for the distance founction.

Modified founction: $\chi^2(h_1\ ,h_2) = \sum_{i=1}^{b} \frac{h_{1i}^2 - h_{2i}^2}{h_{1i}}$

where h1 and h2 are color histograms of two distinct regions, h1i and h2i are the ith component of h1 and h2 respectively, b is the number of histogram bins. Moreover, both histograms are normalized, i.e. their entries sum up to one.

Then: $\mathbf{d_1} = [X^2(h_1,h_2)+X^2(h_1,h_3)+\ldots+X^2(h_1,h_n)]$;

$\mathbf{d_2} = [X^2(h_2,h_1)+X^2(h_2,h_3)+\ldots+X^2(h_2,h_n)]$;

………………..

$\mathbf{d_n} = [X^2(h_n,h_1)+X^2(h_n,h_2)+\ldots+X^2(h_n,h_{n-1})]$;

$\mathbf{d_i}$ is the contrast between ***i***thsuperpixel and all the other superpixels in the image.

1.5 Convert superpixel saliency to pixel saliency

Use $\boldsymbol{d_i}$ (result of step4), replace the pixels of corresponding segments. Then quantize the value to 0~255. The result image as is shown in Figure 3.

1.6 Use central prior to enhance the result

Multiply the initial saliency map with the central prior and obtain the final saliency map. The prior refers to DSDP, downloaded from website. The priors is shown in Figure 4.

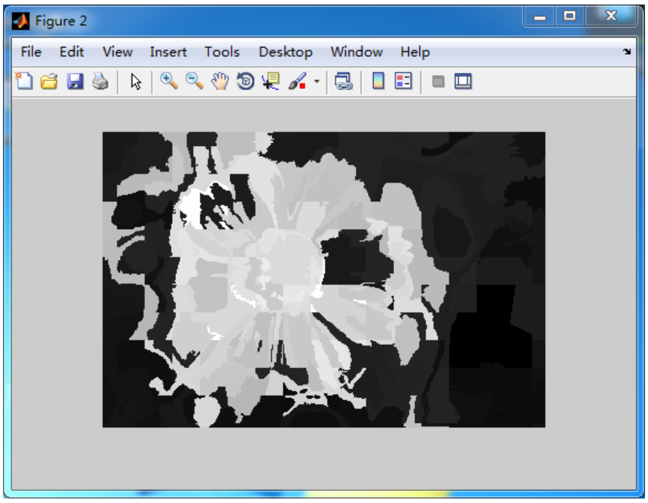

Figure 3: saliency image without prior

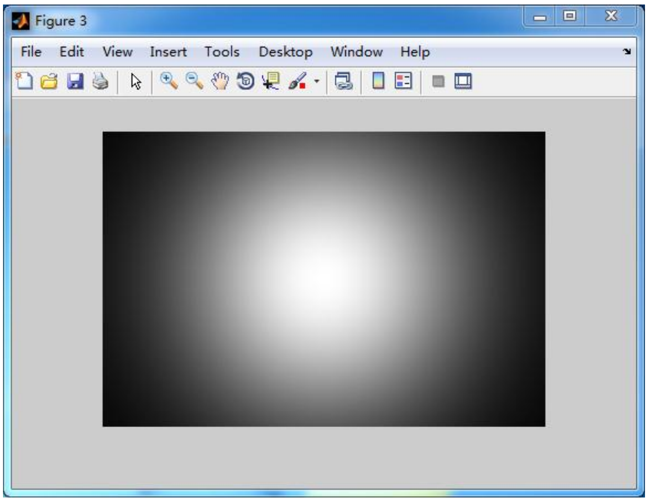

Figure 4: center prior

2. The result

The output map can be seen in Figure 5.

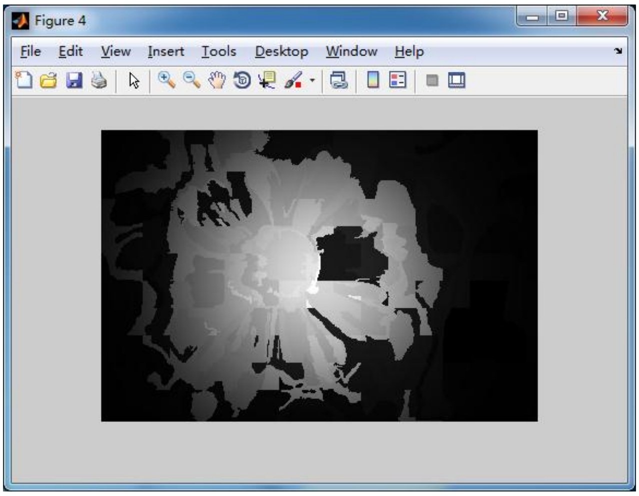

Figure 3: saliency image with center prior