Implementation Strategy

Name: Xinhui Shao

(1) Input an image

```
image_name='1.jpg';
image = imread(image_name );
```

(2) Extract the image information

I choose the color informationwith three channels.

```
r = image(:,:,1);
g = image(:,:,2);
b = image(:,:,3);
```

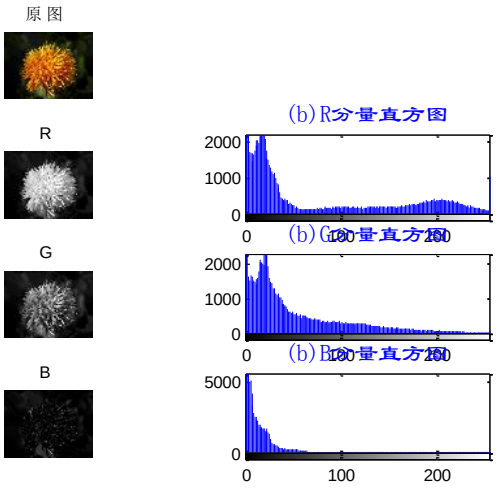

Figure 1 images showing

(3) Quantize each channel

I use “if-end” to achieve it. The idea is that setting a certain range of pixel value as one value, finally the result is the values of 0-15.

For example:

```
%R通道
for i=1:M
    for j=1:N
        if r(i,j)>=0&&r(i,j)<=15
            rr(i,j)=0;
        end
        if r(i,j)>=16&&r(i,j)<=31
            rr(i,j)=1;
        end
        if r(i,j)>=32&&r(i,j)<=47
            rr(i,j)=2;
        end
        if r(i,j)>=48&&r(i,j)<=63
```

```
            rr(i,j)=3;
        end
        if r(i,j)>=64&&r(i,j)<=79
            rr(i,j)=4;
        end
        if r(i,j)>=80&&r(i,j)<=95
            rr(i,j)=5;
        end
        if r(i,j)>=96&&r(i,j)<=111
            rr(i,j)=6;
        end
        if r(i,j)>=112&&r(i,j)<=127
            rr(i,j)=7;
        end
        if r(i,j)>=128&&r(i,j)<=143
            rr(i,j)=8;
        end
        if r(i,j)>=144&&r(i,j)<=159
            rr(i,j)=9;
        end
        if r(i,j)>=160&&r(i,j)<=175
            rr(i,j)=10;
        end
        if r(i,j)>=176&&r(i,j)<=191
            rr(i,j)=11;
        end
        if r(i,j)>=192&&r(i,j)<=207
            rr(i,j)=12;
        end
        if r(i,j)>=208&&r(i,j)<=223
            rr(i,j)=13;
        end
        if r(i,j)>=224&&r(i,j)<=239
            rr(i,j)=14;
        end
        if r(i,j)>=240&&r(i,j)<=255
            rr(i,j)=15;
        end
      end
    end
```

(4) Segment the input image to the superpixels.

I load a vlfeat-library, and convert into lab channel by "`vl_xyz2lab`", then segment by "`vl_slic`". At the beginning, the code is:

```
imlab = vl_xyz2lab(vl_rgb2xyz(image)) ;
```

```
segments = vl_slic(I_single,50, 0.1) ;
```

but the result is wrong. So I help "`vl_slic`", SEGMENTS is a UINT32 array containing the superpixel identifier for each image pixel. I use "`single()`" to achieve it. Finally, I choose the first one.

Figure 2 superpixel image

Figure 3 superpixel image

(5) Compute features of each superpixel.

I created the matrixes to store the pixel of every sengment. Then draw each histogram.

Wrong:

1) At the beginning, I initialize each array which resulted in a lot of 0 in the array, and affected histogram.

2) when I ran the program with resulting 48 pieces of histograms. By creating cells, finally put them into an array.

image_combine_hist0	<1x80 double>	16	273
image_combine_hist1	<1x75 double>	0	16
image_combine_hist10	<1x128 double>	0	3426
image_combine_hist11	<1x121 double>	0	272
image_combine_hist12	<1x175 double>	0	3543
image_combine_hist13	<1x160 double>	0	4091
image_combine_hist14	<1x187 double>	0	4032
image_combine_hist15	<1x184 double>	0	3972
image_combine_hist16	<1x179 double>	0	3201
image_combine_hist17	<1x175 double>	0	1107
image_combine_hist18	<1x225 double>	0	3816
image_combine_hist19	<1x231 double>	0	4071
image_combine_hist2	<1x75 double>	0	273
image_combine_hist20	<1x234 double>	0	4035
image_combine_hist21	<1x219 double>	0	3750
image_combine_hist22	<1x229 double>	0	4009
image_combine_hist23	<1x225 double>	0	1635
image_combine_hist24	<1x280 double>	0	1074
image_combine_hist25	<1x275 double>	0	4071
image_combine_hist26	<1x270 double>	0	4068
image_combine_hist27	<1x283 double>	0	4019
image_combine_hist28	<1x273 double>	0	4087
image_combine_hist29	<1x275 double>	0	4073
image_combine_hist3	<1x75 double>	0	0
image_combine_hist30	<1x310 double>	0	2980
image_combine_hist31	<1x320 double>	0	4089
image_combine_hist32	<1x310 double>	0	4050
image_combine_hist33	<1x332 double>	0	4054

Figure 4 the matrix of superpixel

(6) Compute the distance histogram.

```
segments_marix=cell(max_segments+1,1);
for k=1:(max_segments+1)

segments_marix{k,1}=hist(eval(['image_combine_hist',num2str(k-1)]
),128);
end
```

```
distance_hist=zeros(1,max_segments+1);
for i=1:(max_segments+1)
    for j=1:(max_segments+1)
     distance_hist(i) = 2 * sum( (segments_marix{i,1} -
segments_marix{j,1}).^2 ./ (segments_marix{i,1} +
segments_marix{j,1} + eps) );
    end
 end
```

(7) Quantitze.

```
distance_hist_quantity=zeros(1,max_segments+1);
for i=1:(max_segments+1)
distance_hist_quantity(i)=(distance_hist(i)-min(distance_hist))/(
max(distance_hist)-min(distance_hist))*255;
end
```

(8) Convert superpixel saliency to pixel saliency. I crate a new matrix to story the picture.

```
 image_saliency=zeros(M,N);
 for i=1:M
     for j=1:N
     image_saliency(i,j)=distance_hist_quantity(segments(i,j)+1);
     end
 end
```

Figure 5 sallency figure

(9) Use priors to enhance the result.

```
sigmaD = 400;
[rows, cols, junk] = size(image);
coordinateMtx = zeros(rows, cols, 2);
coordinateMtx(:,:,1) = repmat((1:1:rows)', 1, cols);
coordinateMtx(:,:,2) = repmat(1:1:cols, rows, 1);

centerY = rows / 2;
```

```
centerX = cols / 2;
centerMtx(:,:,1) = ones(rows, cols) * centerY;
centerMtx(:,:,2) = ones(rows, cols) * centerX;
SDMap = exp(-sum((coordinateMtx - centerMtx).^2,3) / sigmaD^2);
VSMap = image3.* SDMap;
```

Figure 6 using priors