双边滤波说明：

这个双边滤波代码通过预先读入待处理图像的先验图，然后利用MATLAB库函数imhist（）读出直方图，这样我们就可以看到不同灰度值上像素点的数量，然后利用直方图设定参数，将图像划分成三部分，灰度值较高、适中、较低的三部分。

saliencyMap = double(imread('SaliencyMap2.jpg'))/255;

figure;

imhist(saliencyMap);

[x,y,v] = find(saliencyMap< 0.35); //找出灰度值低于0.35的部分

[x,y,v] = find(saliencyMap>0.7); //找出灰度值高于0.7的部分

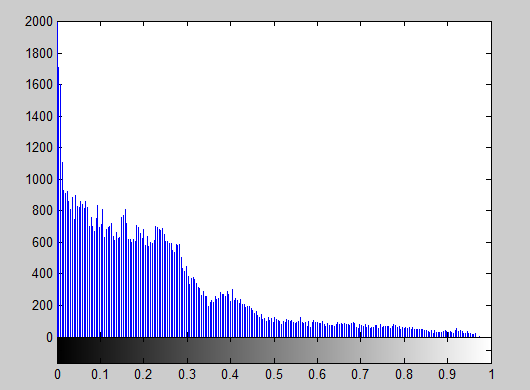
[x,y,v] = find(0.35<saliencyMap); //找出灰度值介于0.35-0.7的部分

C1=zeros(size(saliencyMap));

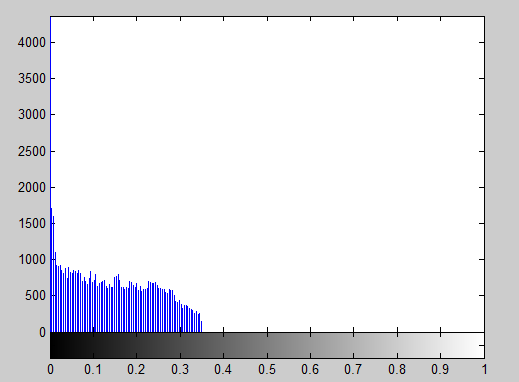
C1(sub2ind(size(C1), x, y))=1;

Cimage1 = C1.\*saliencyMap;

[x,y,v] = find(Cimage1<0.7);



[x,y,v] = find(saliencyMap< 0.35);



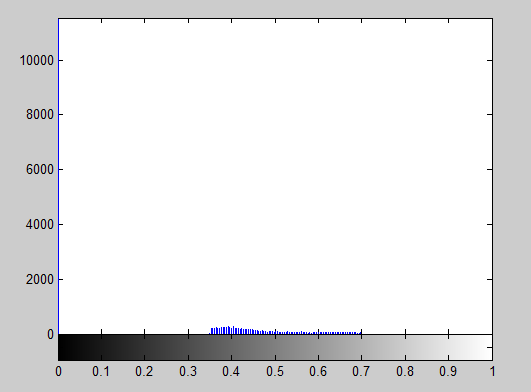
[x,y,v] = find(0.35<saliencyMap);

C1=zeros(size(saliencyMap));

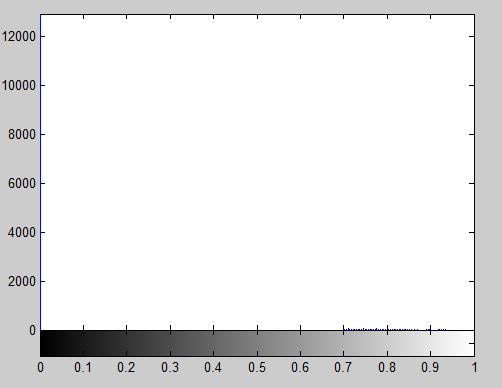
C1(sub2ind(size(C1), x, y))=1;

Cimage1 = C1.\*saliencyMap;

[x,y,v] = find(Cimage1<0.7);



[x,y,v] = find(saliencyMap>0.7);



然后利用矩阵方法，将图像分开，显示

Firstlevel(:,:,1) = A .\* Itemp(:,:,1); //找出第一部分

Firstlevel(:,:,2) = A .\* Itemp(:,:,2);

Firstlevel(:,:,3) = A .\* Itemp(:,:,3);

Secondlevel(:,:,1) = C .\* Itemp(:,:,1); //找出第二部分

Secondlevel(:,:,2) = C .\* Itemp(:,:,2);

Secondlevel(:,:,3) = C .\* Itemp(:,:,3);

Thirdlevel(:,:,1) = B .\* Itemp(:,:,1); //找出第三部分

Thirdlevel(:,:,2) = B .\* Itemp(:,:,2);

Thirdlevel(:,:,3) = B .\* Itemp(:,:,3);

分别拿双边滤波进行处理，最后进行叠加

Firstlevel = double(Firstlevel)/255; //处理第一部分

Firstlevel = Firstlevel +0.03\*randn(size(Firstlevel));

Firstlevel(Firstlevel<0) = 0;Firstlevel(Firstlevel>1) = 1;

FirstlevelTemp= bfilter2(Firstlevel,5,[3 0.1]); //bfilter2是一个自定义的双边滤波函数

Secondlevel = double(Secondlevel)/255; //处理第二部分

Secondlevel =Secondlevel +0.03\*randn(size(Secondlevel ));

Secondlevel(Secondlevel<0) = 0; Secondlevel(Secondlevel>1) = 1;

SecondlevelTemp = bfilter2(Secondlevel,5,[ 3, 0.1]);

Thirdlevel = double(Thirdlevel)/255; //处理第三部分

Thirdlevel = Thirdlevel +0.03\*randn(size(Thirdlevel ));

Thirdlevel (Thirdlevel <0) = 0; Thirdlevel (Thirdlevel>1) = 1;

ThirdlevelTemp = bfilter2(Thirdlevel,2.5,[0.6,0.02]);

Result = FirstlevelTemp + SecondlevelTemp + ThirdlevelTemp ;

导向滤波说明：

导向滤波方法，主要是将一张RGB图分成R、G、B三个部分，

I= imread('forrest\_small.jpg');

Itemp = double(I)/255;

I\_R = Itemp(:,:,1); //分出R



I\_G = Itemp(:,:,2); //分出G



I\_B = Itemp(:,:,3); //分出B

****

对于不同的部分，分别进行处理，

I\_R\_G = abs(Ibf\_R - Ibf\_G);

I\_G\_B = abs(Ibf\_G -Ibf\_B);

I\_B\_R = abs(Ibf\_B- Ibf\_R);

Imean = (I\_R\_G + I\_G\_B + I\_B\_R)./3;

q(:, :, 1) = guidedfilter(Iguidance, Itemp(:, :, 1), r, eps);

q(:, :, 2) = guidedfilter(Iguidance, Itemp(:, :, 2), r, eps);

q(:, :, 3) = guidedfilter(Iguidance, Itemp(:, :, 3), r, eps);

//guidefilter是一个自定义的导向滤波函数

最后显示合起来的部分