Assignment 3 Grab Cut and MeanShift

Liu Chang

Part I :Grab_Cut

Step1 Input image

We download a dataset called PASCAL, which contains 850 images. Use all images from the dataset for segmentation, and evaluate grab cut by adjusting different parameters.

```
%%%Grab_Cut
%%%此文件完成 Grab_cut 中的批量读入图片、显著性检测和图像二值化的部分
clear all;
%Step 1 Input Image
file_path= ='C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\PASCAL';% 图像文件夹路径
img_path_list = dir(strcat(file_path,'*.jpg'));%获取该文件夹中所有 jpg 格式的图像
img_num = length(img_path_list);%获取图像总数量
mkdir imgSignature;%创建 imgSignature 文件夹，用来存储经过 saliency 并二值化后的图像
if img_num > 0 %有满足条件的图像
        for j = 1:img_num %逐一读取图像
            image_name = img_path_list(j).name;% 图像名
            image =   imread(strcat(file_path,image_name));
            fprintf('%d %d %s\n',j,strcat(file_path,image_name));% 显示正在处理的图像名
```

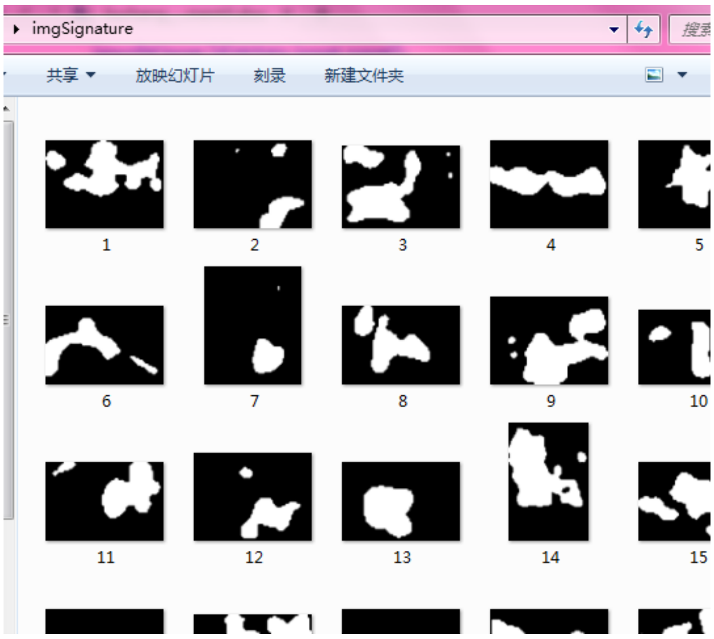
Step 2 Use Image Signature

The color image (m × n × 3 matrix) is regarded as the input for the system. The saliency map is obtained by image signature.

```
            %%%step 2 use image signature
            map=signatureSal(image);% 完成 image signature
            %%%step 3 get the binary image and store the images
            tt=graythresh(map);%自动确定二值化阈值
            image2=im2bw(map,tt);%对图像二值化
            smap = mat2gray( imresize(image2,[size(image,1) size(image,2)]) );
                                                  %将生成的 map 图像调整为与原图同样大小的图像
           directory=[cd,'.\imgSignature\'];%将生成的图片存入 imgSignature 文件夹中
            imwrite(smap,[directory,image_name]);
        end
end
```

After we got the saliency map, we transforms them to binary map with the help of thresholding. Then , store all the 850 images into a file named “imgSignature”.

The “imgSignature” file is ready for the next step ,which is drawing the rectangle.



Step 3: Draw the rectangle&Step 4: Implement grab cut

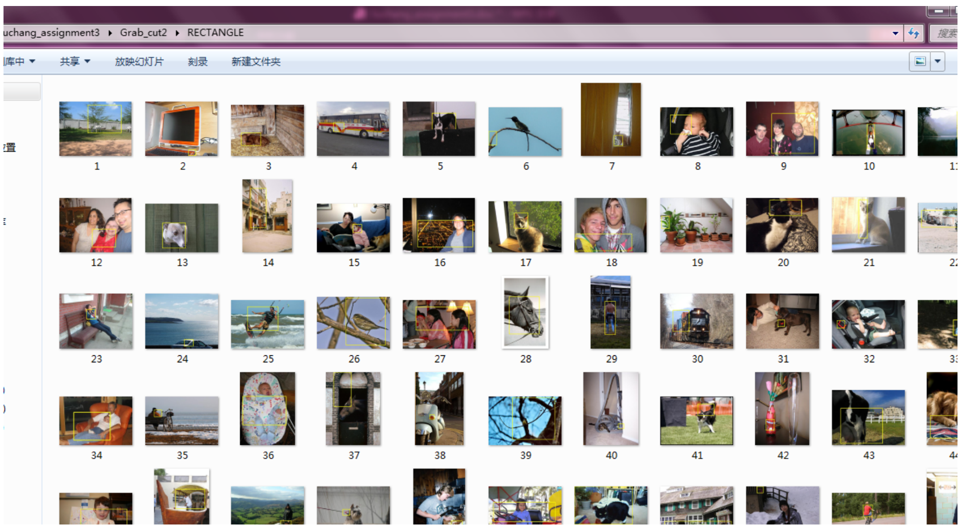
Put the image signatures into Visual Studio 2008(with Open CV inside) to implement this step. The input for this "cpp file" is the saliency map image(m×n matrix).The output is a rectangle that is used to initialize the grab cut. The rectangle is used to locate the most probable position of object and initialize mask in grab cut. We use thresholding to transform saliency map to binary image and draw the rectangle according to the binary image. The size of rectangle is also adjustable.

Firstly, I use a for loop to process images of seven thresholds and use two strings to store the path of origin images and saliency maps. Then I use the function GetListFiles of class Directory to get all of the images stored in these paths and store their information in two string vectors. Thirdly, I use one for loop to process images of each threshold with the iterations of 3,4,5 and 6 in Grab Cut. I read in the origin image and saliency map and their length and height. Then I improve the program of drawing rectangle through find the edges from four directions in order to void traversing all the pixels and reducing the processing time. After that, I use the program which is given to proceed Grab Cut ,save the final map and show which step is proceeding in the screen.

```
#include <iostream>
#include <cv.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/core/core.hpp>
```

```
#include<string>
#include<contrib/contrib.hpp>
using namespace std;
using namespace cv;
int main()
{       for(int threshold=3;threshold<4;threshold++)
        {       Directory dir1,dir2;
                char threch[1];
                itoa(threshold,threch,10);
                string threst=threch;
                string path1=
'C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\PASCAL\';
                string path2=
'C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\Grab_cut1\imgSignature ';
                string exten="*.jpg";
                string extenout=".jpg";
                string filenames3;
                bool addPath1 = true;
                string output=
"C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\Grab_cut2\RECTANGLE";
                char numch[1];
                vector<string> filenames1 = dir1.GetListFiles(path1, exten, addPath1);
                vector<string> filenames2 = dir2.GetListFiles(path2, exten, addPath1);
                for(int n=0;n<filenames1.size();n++)
                {       itoa(n+1,numch,10);
                        filenames3=output+numch+extenout;
                        Mat mOriginImg=imread(filenames1[n],1);                         //输入原始图像
                      Mat mSaMap=imread(filenames2[n],0);             //输入 saliency map 的二值图像
                      const int nRow=mOriginImg.rows;
                      const int nCol=mOriginImg.cols;
                    %%%%Draw Rectangle
                      int i=0,j=0,c=0;
                      int nRowMin=0, nRowMax=0, nColMin=0, nColMax=0;
                        for(i=0;i<nRow;i++)
                        {       for(j=0;j<nCol;j++)
                                {       if(mSaMap.at<uchar>(i,j)==255)
                                        {       nColMin=j;
                                                nColMax=j;
                                                nRowMin=i;
                                                nRowMax=i;
                                                c=1;
                                                break;}}
                                if(c==1)
                                        break;}
```

```
   for(c=0,j=0;j<nColMin;j++)
   {       for(i=nRowMin;i<nRow;i++)
           {       if(mSaMap.at<uchar>(i,j)==255)
                   {       nRowMax=i;
                           nColMin=j;
                           c=1;
                         break;}         }
           if(c==1)
                   break;         }
   for(c=0,i=nRow-1;i>nRowMax;i--)
   {       for(j=nColMin;j<nCol;j++)
           {       if(mSaMap.at<uchar>(i,j)==255)
                   {       nRowMax=i;
                           if(j>nColMax)
                                   nColMax=j;
                           c=1;
                           break;}}
           if(c==1)
                   break;}
   for(c=0,j=nCol-1;j>nColMax;j--)
   {for(i=nRowMin;i<nRowMax;i++)
           { if(mSaMap.at<uchar>(i,j)==255)
                   {       nColMax=j;
                           c=1;
                           break;}}
           if(c==1)
                   break;}
Mat mDrawRec=mOriginImg.clone();
Point2f pRecLeftUp,pRecRightDown;
pRecLeftUp=cvPoint(nColMin,nRowMin);
pRecRightDown=cvPoint(nColMax,nRowMax);
rectangle( mDrawRec, pRecLeftUp,pRecRightDown, Scalar(0, 255, 0), 2);
```



```
%%%%%%grab cut
                    Rect rect(pRecLeftUp,pRecRightDown);
                    Mat OutMask(mOriginImg.size(), CV_8UC1);
                    Mat BgdModel, FgdModel;
                    int nIteration=4;
                    Mat mask;
                    Mat obj;
                    bool isInitialized=false;
                    for(i=0; i<nIteration; i++)
                    {             if(!isInitialized)
                    { grabCut(mOriginImg, OutMask, rect, BgdModel, FgdModel, 1, GC_INIT_WITH_RECT);
                                         isInitialized=true;}
                               else { grabCut(mOriginImg, OutMask, rect, BgdModel, FgdModel, 1); }}
                    compare(OutMask, GC_PR_FGD, mask, CMP_EQ);
                    mOriginImg.copyTo(obj, mask);
      %%%%segmentation results
                    Mat    mSegBinary=obj.clone();
                    cvtColor(mSegBinary,mSegBinary,CV_RGB2GRAY);
                    for(i=0;i<nRow;i++)
                          for(j=0;j<nCol;j++)
                          {    if(mSegBinary.at<uchar>(i,j)>0)
                                                       mSegBinary.at<uchar>(i,j)=255;                    }
                             imwrite(filenames3,mSegBinary);
                      cout<<n+1<<endl;              }      }
       return 0;}
```

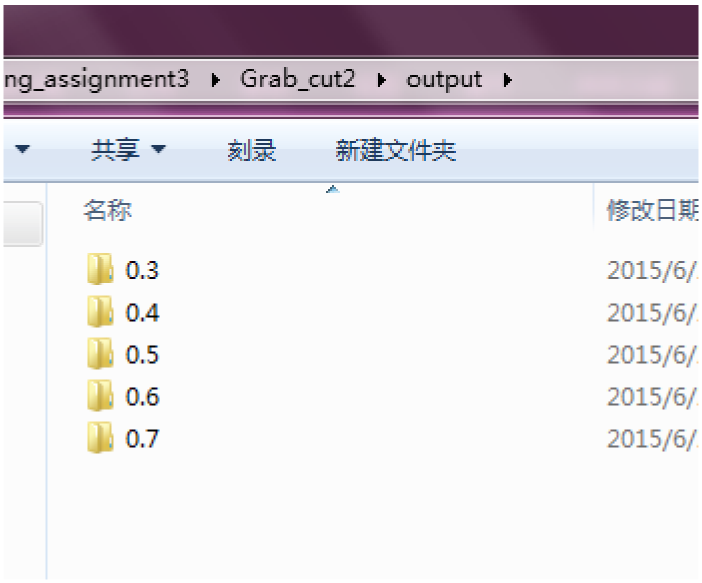
Step 5: Evaluate segmentation result

Input Segmentation result (m × n × 3 matrix) and groundtruth (m × n matrix) from dataset.

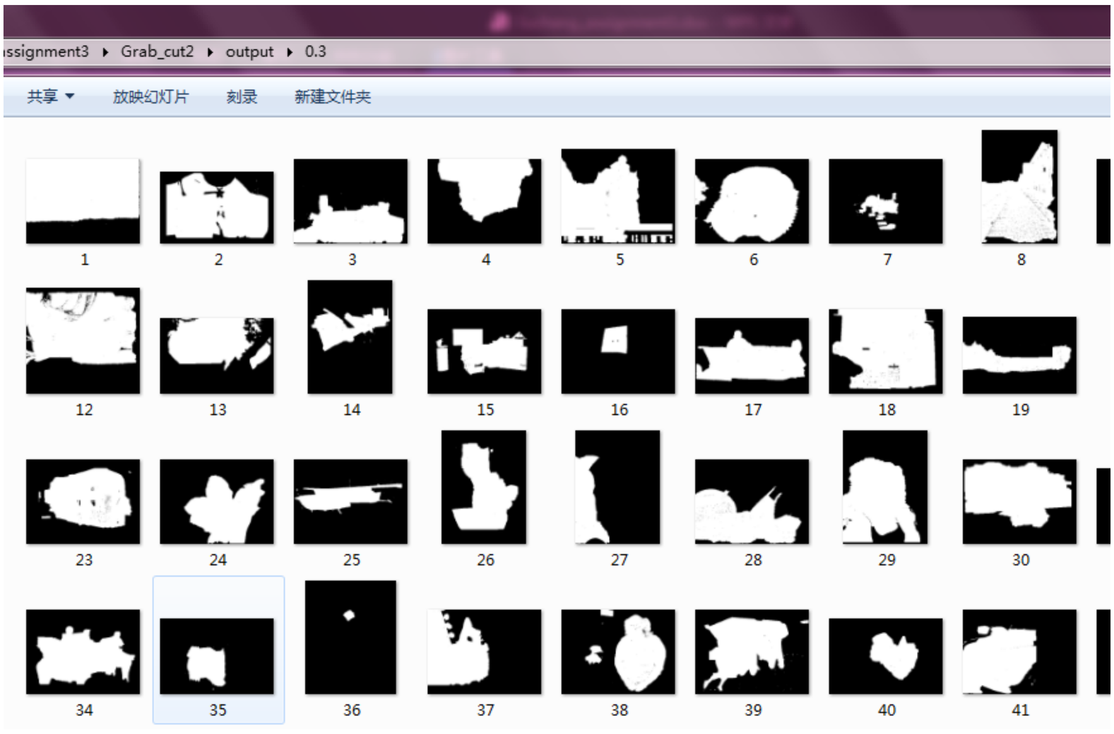
Output A figure that indicates evaluation results (The horizontal axis represents the parameter, the vertical axis represents the evaluation results.).

```
clear;
close all;
clc;
file_path = 'C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\PASCAL_GT' ;
binary_map_path_f='C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\Grab_cut1\i
mgSignature';
groundtruth_path_list = dir(strcat(file_path,'*.png'));
groundtruth_num = length(groundtruth_path_list);
saveall=[];
for threshold=0.3:0.1:0.7
     binary_map_path=strcat(binary_map_path_f,num2str(threshold),'\');
     binary_map_path_list = dir(strcat(binary_map_path,'*.jpg'));
     binary_map_num = length(binary_map_path_list);
```

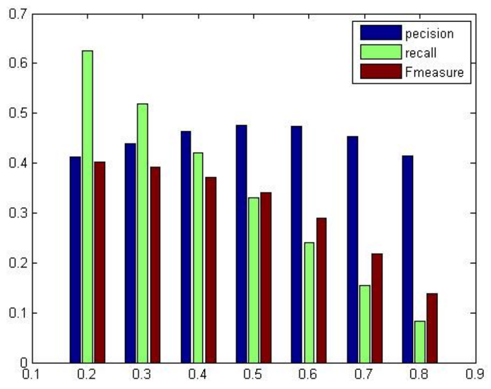
```
        savedata=[];
        for i=1:50
            groundtruth_name=groundtruth_path_list(i).name;
            binary_map_name= binary_map_path_list(i).name;
            groundtruth=imread(strcat(file_path,groundtruth_name));
            salencymap=imread(strcat(salencymap_path,salencymap_name));
            [temp1 temp2 temp3]=prfCount(im2double(groundtruth), im2double(salencymap));
            temp=[temp1,temp2,temp3];
            savedata=[savedata;temp];
        end
            savedata_mean=mean(savedata);
            saveall=[saveall;savedata_mean];
    end
    bar([0.3,0.4,0.5,0.6,0.7],saveall);
    set(gca,'XTick',[0.3:0.1:0.7]);
    legend('Precision','Recall','F-measure',4);
```



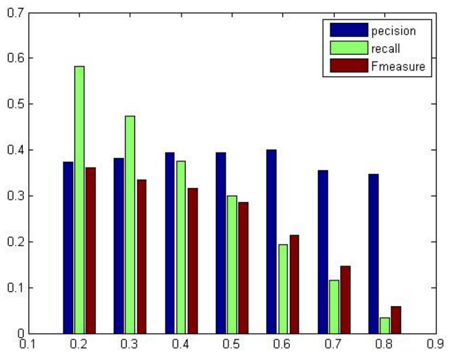
名称	修改日期
0.3	2015/6/
0.4	2015/6/
0.5	2015/6/
0.6	2015/6/
0.7	2015/6/



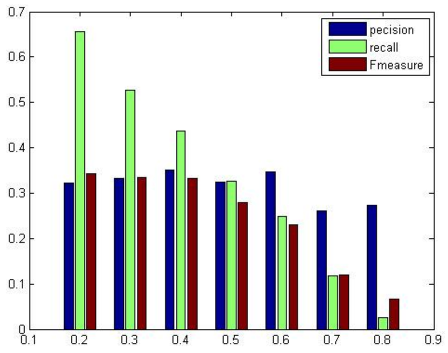
To make the system better, we change the threshold from 0.3 to 0.7. That is to say, threshold is set as 0.3, 0.4, 0.5, 0.6 and 0.7. All the results are stored in the file named as the number of threshold affiliated to the "output"file .



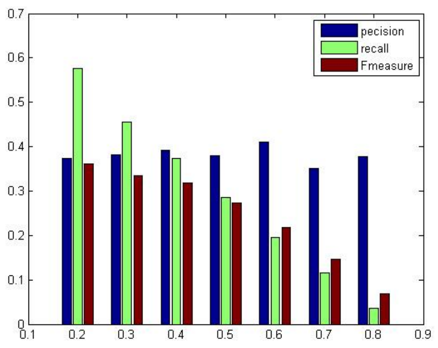
k=3



k=4



k=5



k=6

Part II : MeanShift

Step 1: Input image

We download a dataset, which contains 850 images. Use all images from the dataset for segmentation, and evaluate grab cut by adjusting different parameters.

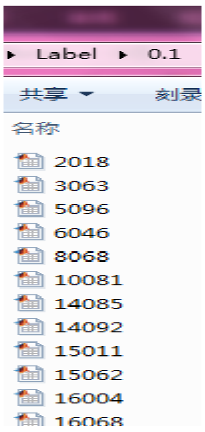
```
clear all;
clc;
%设置 meanShift  参数%
hs = 20 ; % the bandwidth of spatial kernel
%hr = 10 ; % the bandwidth of feature kernel
th = 0.05 ; %    the threshod of the convergence criterion (default = .25)
plotOn = 1; % switch on/off the image display of intermediate results (default = 1)
%  设置文件路径%
img_file='C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\BSDS500\groundTruth\
train' ;
%ground_truth_file='C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\BSDS500\g
roundTruth\train';
store_clust_file='C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\MeanShift\Clust
erImage\clustd_img';
store_seg_file='C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\MeanShift\SegIm
age\seg_img' ;    %%
%store_MS_evalu='C:\Users\Administrator.NTPVB2AO09XFWFK\Desktop\liuchang_assignment3\MeanShift\ev
aluation';
%读入所有图片名字%
dir_img = dir( strcat(img_file, '*.jpg') );
img_cell = struct2cell(dir_img);
img_num = size(img_cell, 2);
%对每张图片进行处理%
%for hs = 20 : 10 : 40
```

```
    for hr = 10 : 10 : 40
        store_seg_path = strcat( store_seg_file, num2str(hs), num2str(hr), '\' );
        store_clust_path = strcat( store_clust_file, num2str(hs), num2str(hr), '\' );
        for i = 2 : img_num
            img_name = img_cell{1, i};
            img_path = strcat( img_file, img_name );
            orig_img = double( imread( img_path ) );
            [clusted_img, aver_MS] = meanShiftPixCluster( orig_img,hs,hr,th,plotOn );
            clusted_img = uint8(clusted_img);
            imwrite( clusted_img,strcat(store_clust_path,img_name) );
            seg_img = processSuperpixelImage(clusted_img);
            save( strcat(store_seg_path,strrep(img_name,'jpg','mat')),'seg_img'  );
        end
    end
%end
```

Step 2: Segment via mean shift

In this step, we need to adjust the parameters of mean shift to get different segmentation results. Input is the color image (m×n × 3 matrix). Output is the segmentation results (m×n×3matrix) and label matrixes (m×n matrix)of different parameters.

```
function imsegs = processSuperpixelImage(fn)
% imsegs = processSuperpixelImage(fn). Creates the imsegs structure from a segmentation image.
% INPUT:    fn - filenames of segmentation images. Use '/' (not '\') to separate directories.
%Segments are denoted by different RGB colors.
% OUTPUT: imsegs - image segmentation data.
fn = {fn};
imsegs(length(fn)) = struct('imname', '', 'imsize', [0 0]);
for f = 1:length(fn)
    im = double(fn{f});
    imsegs(f).imname = 'empty';
    imsegs(f).imsize = size(im);
    imsegs(f).imsize = imsegs(f).imsize(1:2);
    im = im(:, :, 1) + im(:, :, 2)*256 + im(:, :, 3)*256^2;
    [gid, gn] = grp2idx(im(:));
    imsegs(f).segimage = uint16(reshape(gid, imsegs(f).imsize));
     imsegs(f).nseg = length(gn);
end
```



2018
3063
5096
6046
8068
10081
14085
14092
15011
15062
16004
16068

Step 3: Evaluate segmentation result with groundtruth

In this step, function “ meanShiftPixCluster” implements the classic mean shift pixel. Inputs of this function are shown as below: 1. x = an input image (either gray or rgb； 2. hs = the bandwidth of spatial kernel；3. hr = the bandwidth of feature kernel；4. th = the threshod of the convergence criterion； 5. plotOn = switch on/off the image display of intermediate results (default = 1) . Outputs of this function are “y = the output pixel clustered image” and “MS = the output of averaged mean shift”.

```
function [y, MS] = meanShiftPixCluster(x,hs,hr,th,plotOn)
if nargin<3
    error('please type help for function syntax')
elseif nargin == 3
    th = 1/100; plotOn = 1;
elseif nargin == 4
    if th<0 || th >255
        error('threshold should be in [0,255]')
    else
        plotOn = 1;
    end
elseif nargin == 5
    if sum(ismember(plotOn,[0,1])) == 0
        error('plotOn option has to be 0 or 1')
    end
elseif nargin>5
    error('too many input arguments')
end
%% initialization
x = double(x);
[height,width,depth] = size(x);
y = x;
done = 0;
```

```
iter = 0;
if plotOn
    figure(randi(1000)+1000);
end
% padding image to deal with pixels on borders
xPad = padarray(x,[height,width,0],'symmetric');
% build up look up table to boost computation speed
weight_map = exp( -(0:255^2)/hr^2 );
MS = [];
%% main loop
while ~done
    weightAccum = 0;
    yAccum = 0;
    % only 99.75% area (3sigma) of the entire non-zero Gaussian kernel is considered
    for i = -hs:hs
        for j = -hs:hs
            if ( i~=0 || j~=0 )
                % spatial kernel weight
                spatialKernel = 1;
                % uncomment the following line to active Gausian kernel
                %spatialKernel = exp(-(i^2+j^2)/(hs/3)^2/2);
                xThis =   xPad(height+i:2*height+i-1, width+j:2*width+j-1, 1:depth);
                xDiffSq = (y-xThis).^2;
                % feature kernel weight
                intensityKernel = repmat( prod( reshape( weight_map( xDiffSq+1 ), height, width, depth) , 3 ),
[1,1, depth]);
                % mixed kernel weight
                weightThis = spatialKernel.*intensityKernel;
                % update accumulated weights
                weightAccum = weightAccum+ weightThis;
                % update accumulated estimated ys from xs
                yAccum = yAccum+xThis.*weightThis;
            end
        end
    end
    % normalized y (see Eq.(20) in the cited paper)
    yThis = yAccum./(weightAccum+eps);
    % convergence criterion
    yMS = mean(abs(round(yThis(:))-round(y(:))));
    y = round(yThis);
    MS(iter+1) = yMS;
    %xPad = padarray(y,[height,width,0],'symmetric');
    if plotOn
        subplot(121), imshow(uint8(y)),axis image, title(['iteration times = ' num2str(iter) '; averaged mean-shift
= ' num2str(yMS)]);
            subplot(122), plot(0:iter, MS ), xlabel('iteration #'), ylabel('averaged mean shift');axis square
            drawnow
      end
      if yMS <= th || iter>5 % exit if converge
            done = 1;
      else % otherwise update estimated y and repeat mean shift
            iter = iter+1;
      end
end
```

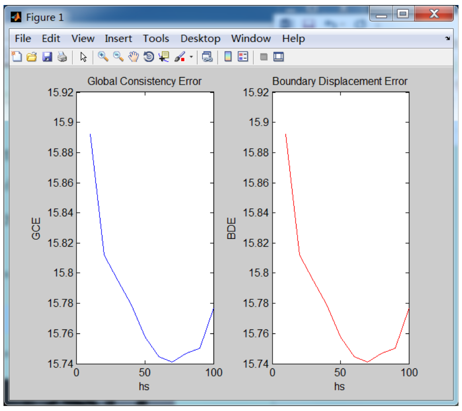
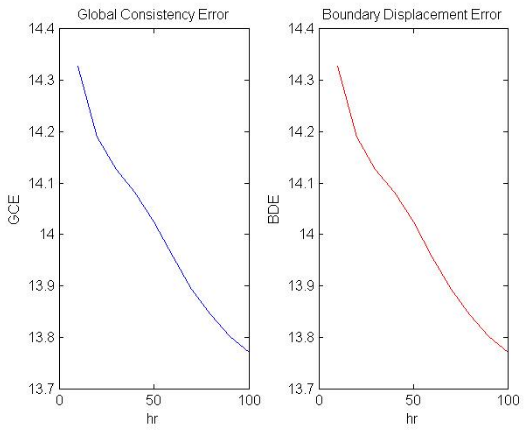


Figure 1
File Edit View Insert Tools Desktop Window Help
Global Consistency Error
Boundary Displacement Error
GCE
BDE
hs
15.92
15.74
0
50
100



From the charts we can get that: we will get the best outcome when the hs is about 70,and when hs is under 70,we will have the better outcome if hs is larger, and if hs is larger than 70, the outcome will become worse.