Assignment 3

Xinhui Shao

Part 1: Grab Cut

1. Input image

In step 1, I read images by batch. Next, I show the program as an example.

```
clear;
close all;
clc;

img_path = 'E:\GrabCut\grabcut_1\gc\PASCAL\' ;
result_path = 'E:\GrabCut\grabcut_1\gc\PASCAL_SALENCY_0.3\' ;

picstr = dir(strcat(img_path,'*.jpg'));
m = length(picstr);
th=0.3;
if m > 0
      for i = 1:m
          img_name = picstr(i).name;
          if img_name(1)=='.'
          continue;
          end
          filename1=strcat(img_path,img_name);
          img = imread(filename1);
          [l,w]=size(img);
          param = default_signature_param(w);
          salency_image = signatureSal( img,param);
          salency = imresize(salency_image , [ size(img,1)
          size(img,2) ] );
          salencyoutput=im2bw(salency,th);
          imwrite(salencyoutput,strcat(result_path,img_name));
        end
end
```

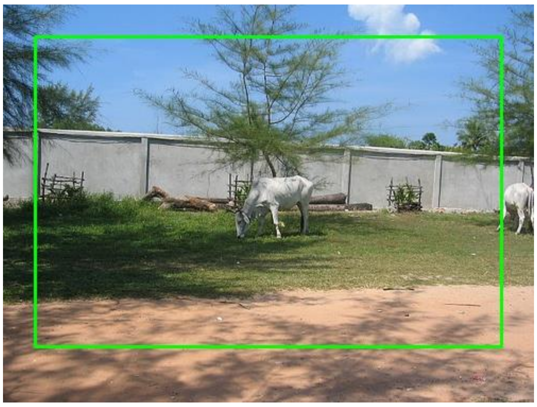
2. Use image signature

In this process, I adjust threshold to transform saliency map for different binary images. As we all know, the image is saved in x. Then I can call functions of ‘signatureSal’ , the result is the saliency map. The saliency map is obtained by image signature and the matlab code of “signatureSal”. The code is as follow:

```
         param = default_signature_param;
          salency_image = signatureSal( img,param);
          salency = imresize(salency_image , [ size(img,1)
          size(img,2) ] );
          salencyoutput=im2bw(salency,th);
          imwrite(salencyoutput,strcat(result_path,img_name));
```

3. Draw the rectangle

I input saliency map image from step 2. The rectangle is used to locate the most probable position of object and initialize mask in grab cut. I use threshold to transform saliency map to binary image and draw a rectangle according to the binary image with different size of rectangle.



4. Implement grab cut

I input image (m × n × 3 matrix) from step 1 and the rectangle from step 3. The codes in C is showing:

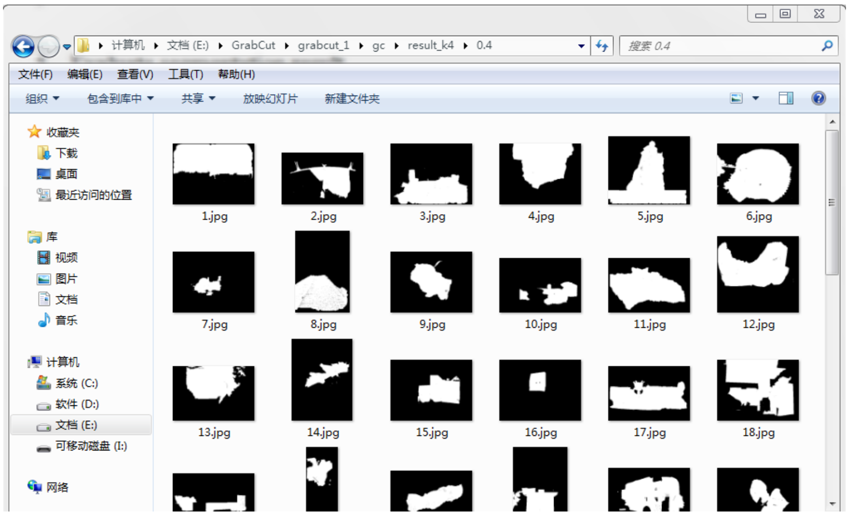
```
#include <iostream>
#include <cv.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/core/core.hpp>
#include<string>
#include<contrib/contrib.hpp>

using namespace std;
using namespace cv;

int main()
{
   for(int threshold=2;threshold<9;threshold++)
```

```
{

    Directory dir1,dir2;
    char threch[1];
    itoa(threshold,threch,10);
    string threst=threch;
    string path1="E:/class/computer vision/assignment3/PASCAL";
    string path2="E:/class/computer
    vision/assignment3/signmap0."+threst;
    string exten="*.jpg";
    string extenout=".jpg";
    string filenames3;
    bool addPath1 = true;
    string output="E:/output/0."+threst+"/";
    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);
        Mat mSaMap=imread(filenames2[n],0);
        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);

//~~~~~~~~~~~~~~~~Implement grab cut~~~~~~~~~~~~~~~~~~~~//
Rect rect(pRecLeftUp,pRecRightDown);
Mat OutMask(mOriginImg.size(), CV_8UC1);
Mat BgdModel, FgdModel;

int nIteration=3;
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);
```

```
    //~~~~~~~~~~~~~~~~Show segmentation result~~~~~~~~~~~~~~~~~~~~//
          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;
}
```



5、Evaluate segmentation result

I input segmentation result and groundtruth from dataset. Adjust threshold to obtain different segmentation results and draw a bar graph to evaluate grab cut.

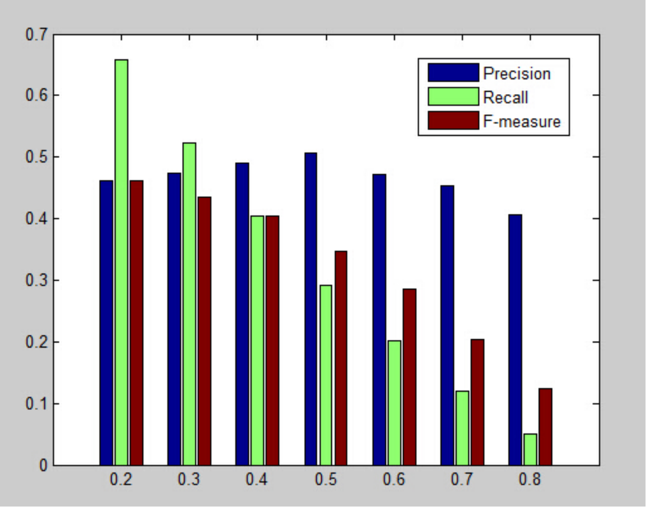


Figure　PRF bar graph k=3

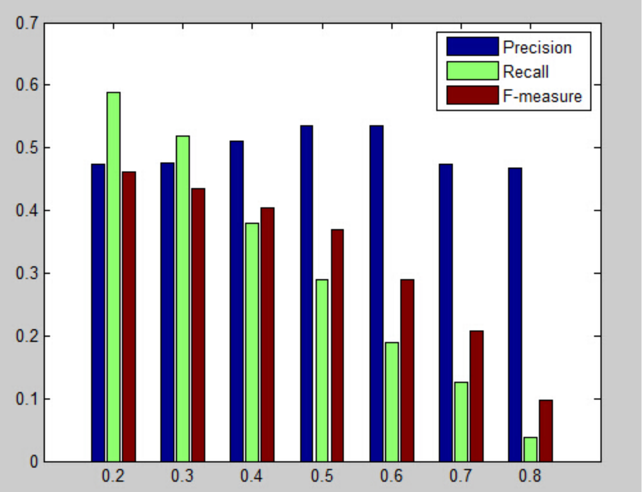


Figure　PRF bar graph k=4

Part 2: Meanshift

1. Input all images by batch processing

I input the image by beach, and show the codes:

```
clear all;
clc;
close all;

picstr=dir('E:\meanshift\BSDS500\data\images\train\*.jpg');
[m,n]=size(picstr);
for i=1:m
    if picstr(i).name(1)=='.'
        continue;
    end
```

```
filename=strcat('E:\meanshift\BSDS500\data\images\train\',picstr(i).n
ame);
end
```

where , the purpose of the program is to skip the hidden files. It is asking for help Ruchen Wang.

```
if picstr(i).name(1)=='.'
        continue;
    end
```

2. Segment via mean shift

I download the source code from the website and adjust the parameters (hs, hr) from 10 to 80. Every result is saved as different folders named by relative parameter.

```
for hs=10:5:40
    for hr=10:10:90
        for i=1:m
            if picstr(350).name(1)=='.'
            continue;
            end

            filename=strcat('E:\meanshift\BSDS500\data\images\train\',p
            icstr(350).name);
            x=imread(filename);
            th=0.1;
            iteration=5;
            [y, MS] = meanShiftPixCluster(x,hs,hr,th,iteration);
            img_ms=uint8(y);

            fname_img=strcat('E:\meanshift\BSDS500\data\images\train_im
            g','\','hs_',num2str(hs),'\','hr_',num2str(hr),'\');

            fname_lab=strcat('E:\meanshift\BSDS500\data\images\train_la
            b','\','hs_',num2str(hs),'\','hr_',num2str(hr),'\');

            fname_img1=['E:\meanshift\BSDS500\data\images\train_img\','
            hs_',num2str(hs),'\','hr_',num2str(hr),'\'];

            fname_lab1=['E:\meanshift\BSDS500\data\images\train_lab\','
            hs_',num2str(hs),'\','hr_',num2str(hr),'\'];
            mkdir(fname_img);
            mkdir(fname_lab);
            fname_img1=strcat(fname_img,picstr(350).name);

            fname_lab1=strcat(fname_lab,picstr(350).name(1:end-4),'.mat
');

            imwrite(img_ms,fname_img1);

            imgInf = processSuperpixelImage(y);
            label_result = double(imgInf.segimage);
            fid_lab=fopen(fname_lab1,'wt');
            fwrite(fid_lab,label_result);
            fclose(fid_lab);
        end
    end
 end
```

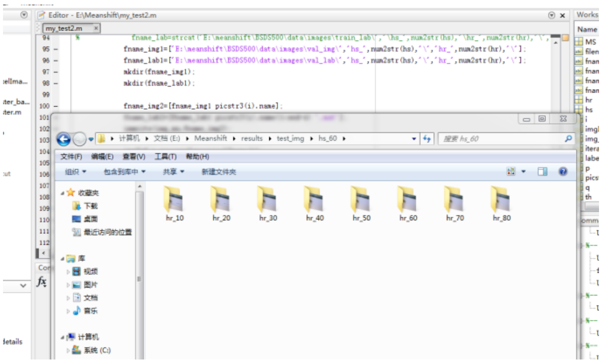
where “`[y, MS] = meanShiftPixCluster(x,hs,hr,th,iteration);`
`img_ms=uint8(y);`” is meanshift’s main function.

Where “`imgInf = processSuperpixelImage(y);`
`label_result = double(imgInf.segimage);`
`fid_lab=fopen(fname_lab1,'wt');`” is making labels.

At first, I writed the code “

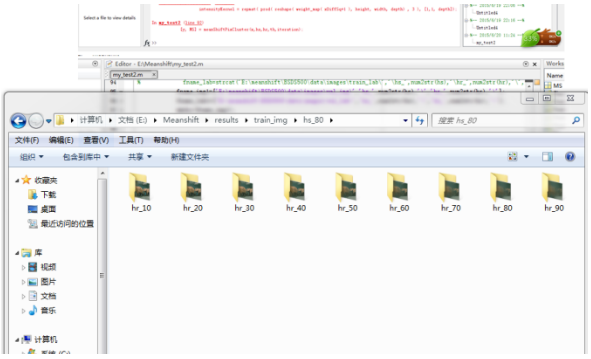
```
for i=1:m
    if picstr(350).name(1)=='.'
    continue;
    end
    for hs=10:5:40
        for hr=10:10:90”
            ...............
         end
     end
 end
```

The results show as follow:



```
%           fname_lab=strcat('E:\meanshift\BSDS500\data\images\train_lab\','\hs_',num2str(hs),'\hr_',num2str(hr),'\',
            fname_img1=['E:\meanshift\BSDS500\data\images\val_img\','hs_',num2str(hs),'\','hr_',num2str(hr),'\'];
            fname_lab1=['E:\meanshift\BSDS500\data\images\val_lab\','hs_',num2str(hs),'\','hr_',num2str(hr),'\'];
            mkdir(fname_img1);
            mkdir(fname_lab1);

            fname_img2=[fname_img1 picstr3(i).name];
```

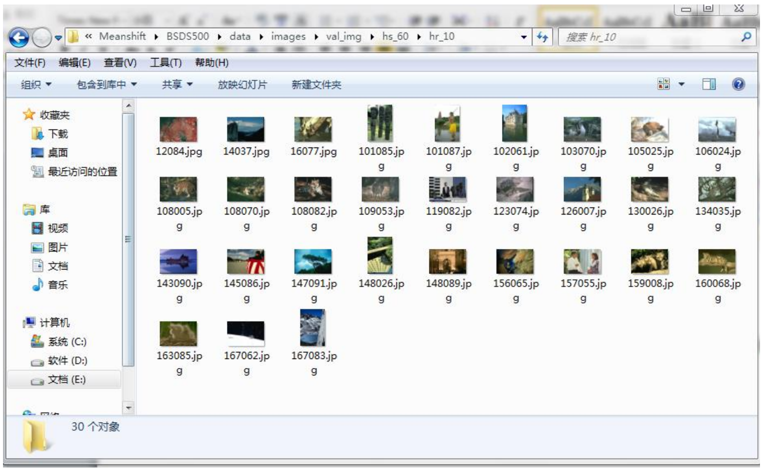
Editor - E:\Meanshift\my_test2.m
my_test2.m
计算机 ▸ 文档 (E:) ▸ Meanshift ▸ results ▸ test_img ▸ hs_60
搜索 hs_60
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 包含到库中 共享 新建文件夹
收藏夹
下载
桌面
最近访问的位置
库
视频
图片
文档
音乐
计算机
系统 (C:)
hr_10
hr_20
hr_30
hr_40
hr_50
hr_60
hr_70
hr_80
Works
Name
MS
hr
hs
i

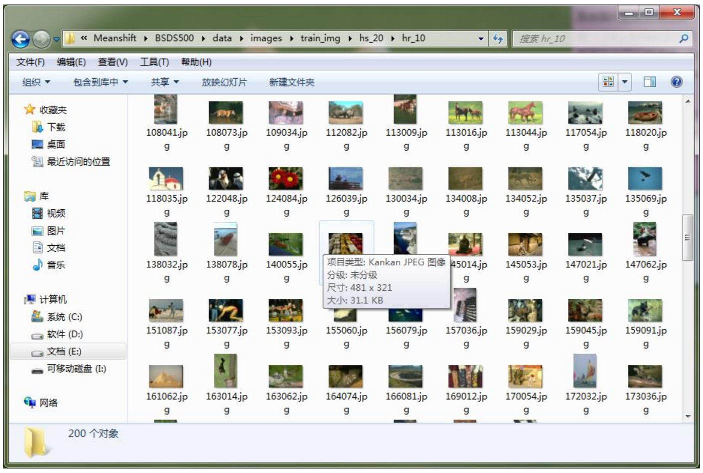


```
intensityKernel = repmat( prod( reshape( weight_map( xDiffSq+1 ), height, width, depth) , 3 ), [1,1, depth]);

In my_test2 (line 92)
    [y, MS] = meanShiftPixCluster(x,hs,hr,th,iteration);
```

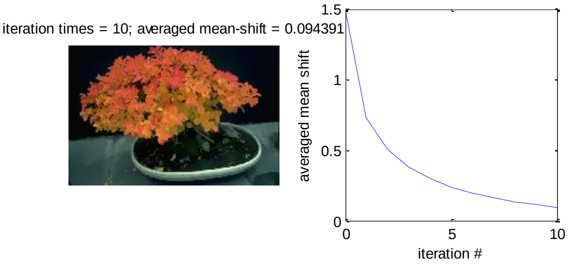
Select a file to view details
Editor - E:\Meanshift\my_test2.m
my_test2.m
计算机 ▸ 文档 (E:) ▸ Meanshift ▸ results ▸ train_img ▸ hs_80
搜索 hs_80
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 包含到库中 共享 新建文件夹
hr_10
hr_20
hr_30
hr_40
hr_50
hr_60
hr_70
hr_80
hr_90

Each folder is saved one map. So modify the program, the results is showing:





The part processing image is showing:

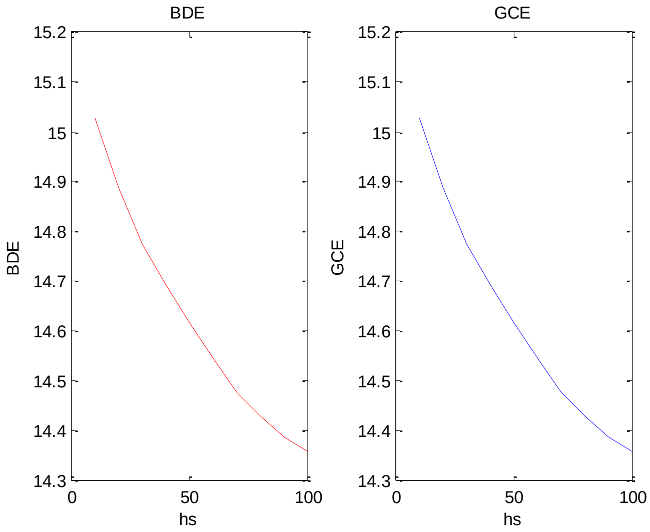


3、Evaluate segmentation result with groundtruth

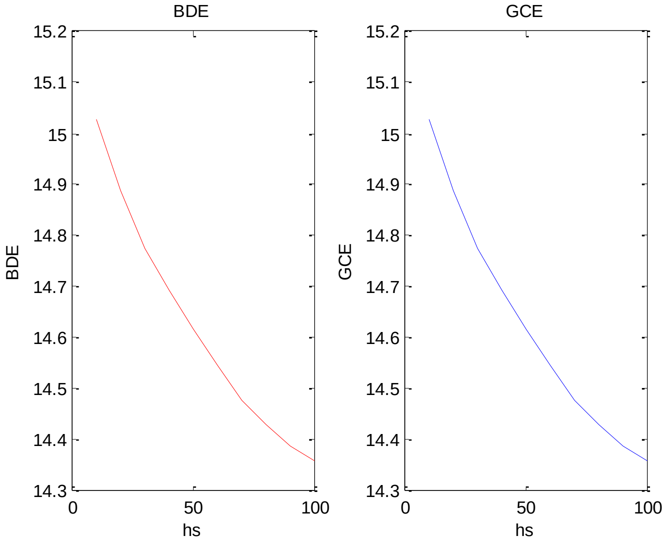
Input the label matrixes (m×n matrix) from step 2, the groundtruth matrixes (m×n matrix) from dataset.

```
clear;
clc;
benchPath = 'E:\meanshift\BSDS500\data\groundTruth\train\';
testPath = 'E:\Meanshift\BSDS500\data\images\train_lab\hs_10';

testList = dir(testPath);
numX = 0;
for i = 1:length(testList)
    if testList(i).name(1)=='.'
        continue;
    end

        numX = numX+1;
        x(1,numX) = str2double(testList(i).name);

        averageBoundaryError = 0;
        averageRI = 0;
        averageVOI = 0;
        averageGCE = 0;
        imageCount = 0;
        testImgPath = [testPath '\' testList(i).name];
        testImgList = dir(testImgPath);
        for j = 1:length(testImgList)
            if testImgList(j).name(1)=='.'
                continue;
            end
            benchImgName = [benchPath testImgList(j).name];
            testImgName = [testImgPath '\' testImgList(j).name];
            load(benchImgName);
            load(testImgName);
            imageCount = imageCount + 1;
            imageLabelCell=groundTruth;
            sampleLabels = label_result;
             % Comparison script
            totalBoundaryError = 0;
            sumRI = 0;
            sumVOI = 0;
            sumGCE = 0;

            [imageX, imageY] = size(sampleLabels);
            [benchX, benchY] = size(imageLabelCell{1}.Segmentation);
            for benchIndex=1:length(imageLabelCell)
                benchLabels = imageLabelCell{benchIndex}.Segmentation;
                totalBoundaryError = totalBoundaryError +
                compare_image_boundary_error(double(benchLabels),
                double(sampleLabels));
                [curRI,curGCE,curVOI] =
                compare_segmentations(sampleLabels,benchLabels);
                sumRI = sumRI + curRI;
                sumVOI = sumVOI + curVOI;
                sumGCE = sumGCE + curGCE;
            end

            % update the averages... note that sumRI /
            length(imageLabelCell) is
            % equivalent to the PRI.
            averageBoundaryError = averageBoundaryError +
            totalBoundaryError / length(imageLabelCell);
            averageRI = averageRI + sumRI / length(imageLabelCell);
            averageVOI = averageVOI + sumVOI / length(imageLabelCell);
            averageGCE = averageGCE + sumGCE / length(imageLabelCell);

        end
        arrayRI(numX,1) = averageRI/imageCount;
        arrayVOI(numX,1) = averageVOI/imageCount;
        arrayGCE(numX,1) = averageGCE/imageCount;
        arrayBE(numX,1) = averageBoundaryError/imageCount;
end
xNum = 0;
for i =1:length(x)
    if x(1,i)~=100;
        xNum = xNum+1;
        xFinal(1,xNum) = x(1,i);
        arrayRIFinal(xNum,1) = arrayRI(i,1);
        arrayVOIFinal(xNum,1) = arrayVOI(i,1);
        arrayGCEFinal(xNum,1) = arrayGCE(i,1);
        arrayBEFinal(xNum,1) = arrayBE(i,1);
    else
        xFinal(1,length(x)) = x(1,i);
        arrayRIFinal(length(x),1) = arrayRI(i,1);
        arrayVOIFinal(length(x),1) = arrayVOI(i,1);
        arrayGCEFinal(length(x),1) = arrayGCE(i,1);
        arrayBEFinal(length(x),1) = arrayBE(i,1);
    end
end

subplot(221);plot(xFinal,arrayBEFinal,'r'),title('BDE'),xlabel('hs'),
ylabel('BDE');
subplot(222);plot(xFinal,arrayRIFinal,'k'),title('PRI'),xlabel('hs'),
ylabel('PRI');
subplot(223);plot(xFinal,arrayVOIFinal,'b'),title('VOI'),xlabel('hs'),
ylabel('VOI');
subplot(224);plot(xFinal,arrayBEFinal,'y'),title('GCE'),xlabel('hs'),
ylabel('GCE');
```

The result is showing: Th=0.1 iterate=5 hr=40



Th=0.1 iterate=5 hs=10


BDE
GCE
BDE
GCE
hs
hs