**北京邮电大学软件学院**

**2019-2020学年第二学期实验报告**

**课程名称：** 多媒体技术与应用

**项目名称： 图像分割功能**

**项目完成人：**

**姓名：\_\_王衔飞\_\_学号：\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**指导教师：**\_\_\_\_\_\_\_\_\_\_\_李朝晖\_\_\_\_\_\_\_\_\_\_\_

**日 期： 2020年 6 月 9 日**

1. **实验目的**

掌握图像分割相关原理及方法。

1. **实验内容**

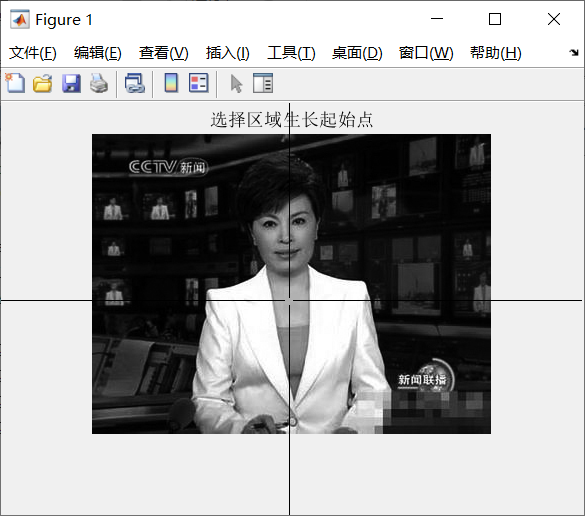
编制程序，尽可能用多种方法实现图像分割功能。

1. **实验环境**

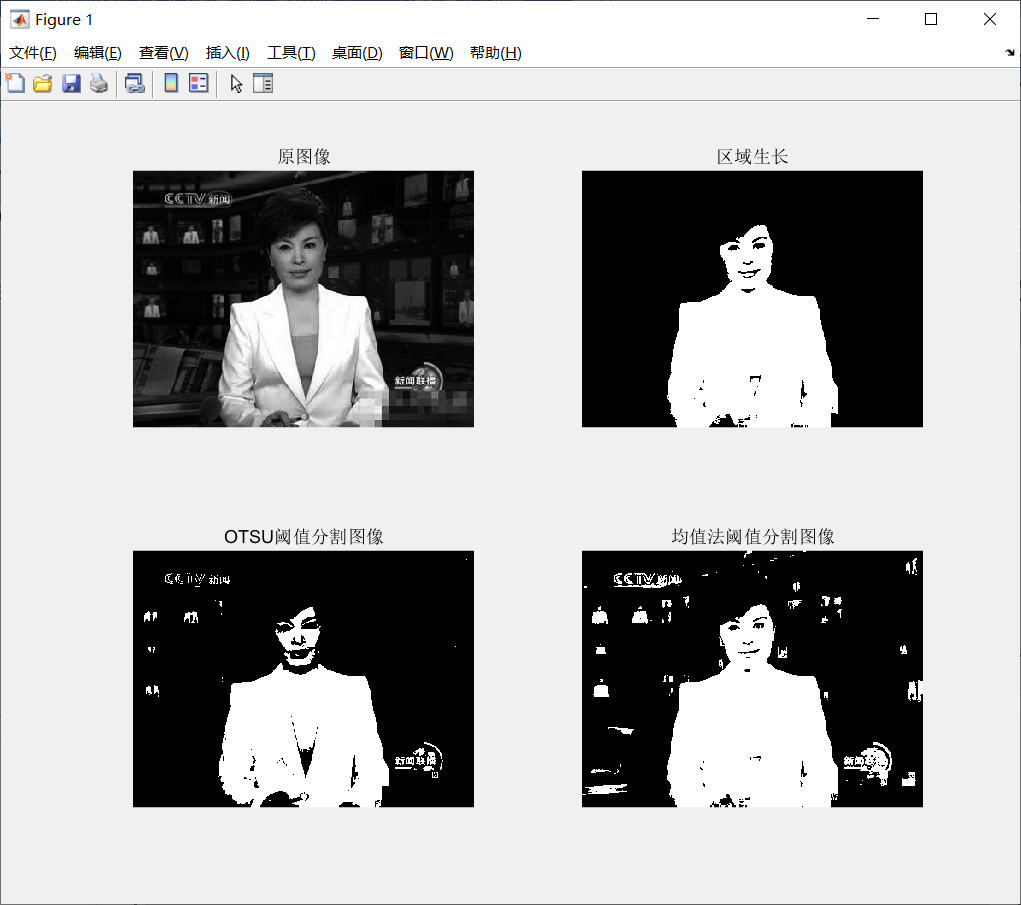
MATLAB2020a

1. **实验结果**

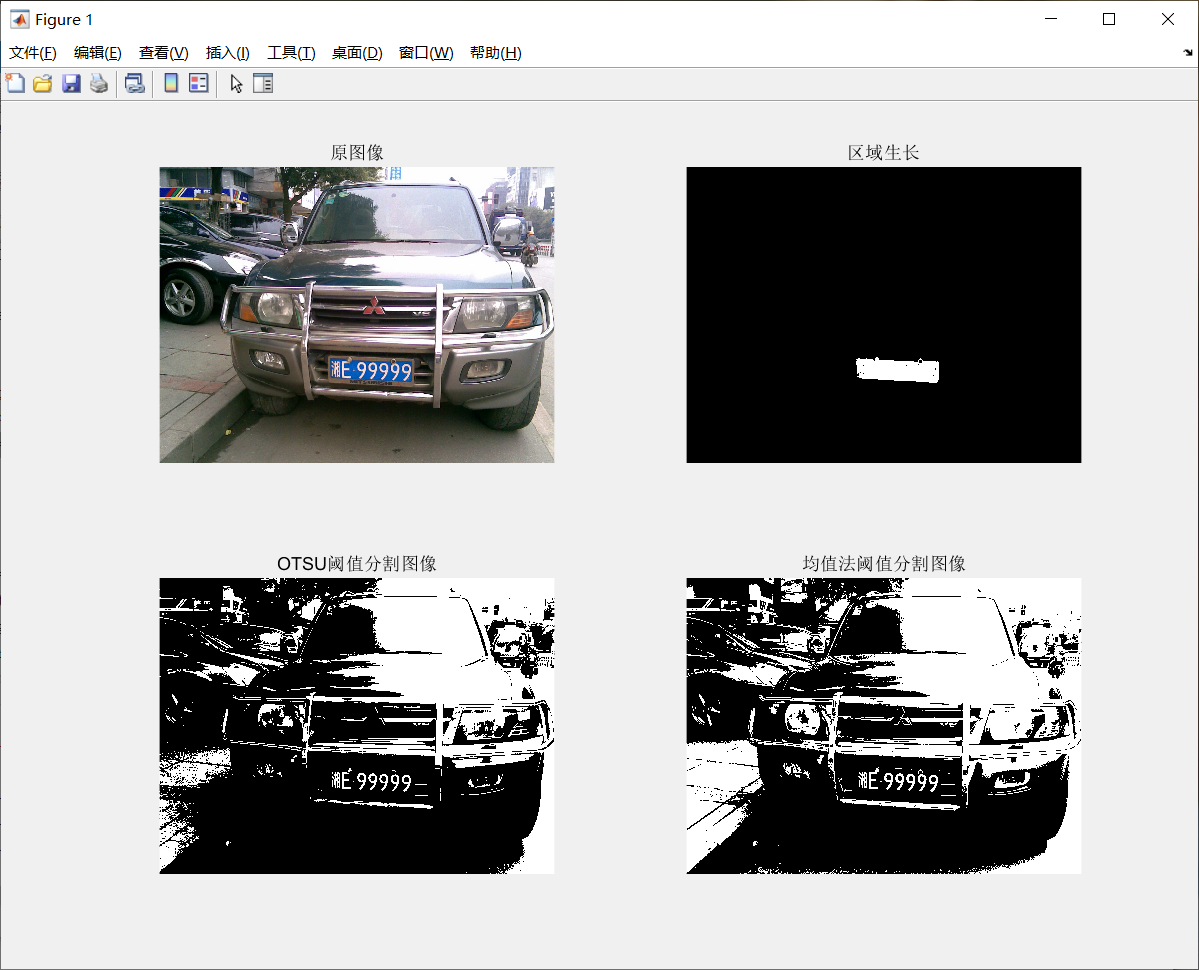
对于区域生长算法，需要在使用前给出种子点位置。对于下文中的实验结果，选择的是播音员的颈部和车牌的中部。选择如图：

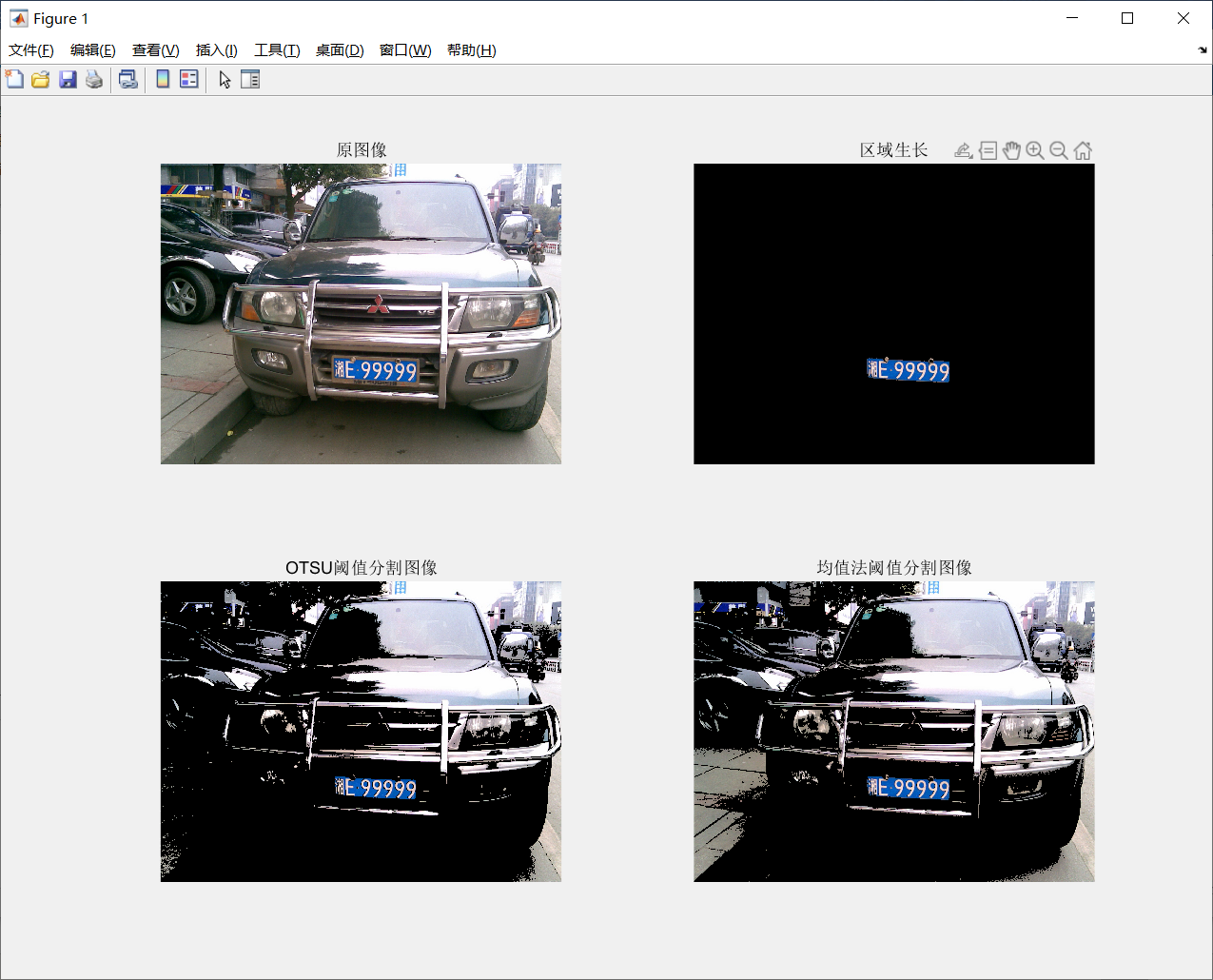
 

使用测试案例 播音员（gray）.jpg 输出截图：

使用测试案例 车牌图像.jpg 输出截图：





其中对于播音员图片，这几种算法均不能较好的扣出其头发（和边缘颜色过于接近），对于车牌图像，用区域生长算法可以较好的抠出车牌部分。（转灰度算法对车牌进行了优化，使用蓝色通道代替直接将图像转为灰度图）

1. **附录**

备注：两测试用例使用的代码相同，唯一区别在于首行的文件名。

代码分为两个版本，一个是输出抠像区域，一个输出抠像结果，分别对应上面截图中的两种截图。

**输出二值图的MATLAB代码：**

f=imread('车牌图像.jpg');

%f=imread('播音员（gray）.jpg');

I1 = f;

I2 = area\_growing(f,100);

I3 = OTSU\_threshold\_segmentation(f);

I4 = average\_threshlod\_segmentation(f);

figure;% 一个2x2显示区域

subplot(2, 2, 1); imshow(I1); title('原图像');

subplot(2, 2, 2); imshow(I2); title('区域生长');

subplot(2, 2, 3); imshow(I3, []); title('OTSU阈值分割图像');

subplot(2, 2, 4); imshow(I4, []); title('均值法阈值分割图像');

grid on

function [outputArg1] = OTSU\_threshold\_segmentation(inputArg1)

%   inputArg1 图像 gray

% outputArg1 = inputArg1;

if size(inputArg1,3)==3

    f = rgb2gray(inputArg1);

else

    f = inputArg1;

end

[M,N] = size(f);

T = 0;

result\_T = 0;

gmax = 0;

while (T<=255)

    g=f>=T;

    w0 = numel(f(g))/(M\*N);

    w1 = numel(f(~g))/(M\*N);

    p0 = mean(f(g));

    p1 = mean(f(~g));

    tmp = (w0 \* w1) \* (p0 - p1)^2;

    if (tmp > gmax)

        gmax = tmp;

        result\_T =T;

    end

    T = T+1;

end

outputArg1 = im2bw(f,result\_T/255);

end

function [outputArg1] = average\_threshlod\_segmentation(input\_args)

% inputArg1 image gray

if size(input\_args,3)==3

    f = rgb2gray(input\_args);

else

    f = input\_args;

end

T = mean(mean(f));

outputArg1 = im2bw(f,T/255);

end

function [ output\_args ] = area\_growing( input\_args, thredshold\_input)

%   input\_args raw image as gray format

%   thredshold\_input input thredshold

%   output\_args convert image as gray format

if size(input\_args,3)==3

    % 取blue通道，便于分隔车牌

    I = input\_args(:,:,3);

else

    I = input\_args;

end

% 显示原始图片

figure,imshow(input\_args),title('选择区域生长起始点');

I=double(I);

[height,width]=size(I);

[y\_double,x\_double]=ginput(1);  %获得区域生长起始点

x=round(x\_double);            %横坐标取整

y=round(y\_double);            %纵坐标取整

seed=I(x,y);           %将生长起始点灰度值存入seed中

Tmp = zeros([height,width]);

Tmp(x,y) = 1;

sum = seed;

seed\_save=1;

seed\_count = 1;

thredshold = thredshold\_input;

close all;

while seed\_count>0

    current\_seed\_gray\_level = 0;

    seed\_count = 0;

    for i=1:height

        for j=1:width

            if Tmp(i,j) == 1

                if (i-1)>0 && (i+1)<(height+1) ...

                        && (j-1)>0 && (j+1)<(width+1)%判断该点不在边缘

                    for m=-1:1

                        for n=-1:1

                            if Tmp(i+m,j+n)==0 && ...%首先该点周围的点应该没有被纳入生成的内容中

                                    abs(I(i+m,j+n)-seed) <= thredshold ... %其次，该点的灰度值需要小于我们设置的阈值

                                Tmp(i+m,j+n) = 1; % 成功判定，将内容输出到输出矩阵中

                                seed\_count= seed\_count+1;%种子数量加一

                                current\_seed\_gray\_level = current\_seed\_gray\_level + I(i+m,j+n); %累加灰度，最后求均值

                            end

                        end

                    end

                end

            end

        end

    end

end

seed\_save = seed\_save + seed\_count;%获取所有种子的个数

sum = sum +current\_seed\_gray\_level;

seed = sum/seed\_save;%并计算灰度

output\_args = Tmp;

end

**输出抠像结果的MATLAB代码：**

f=imread('车牌图像.jpg');

%f=imread('播音员（gray）.jpg');

I1 = f;

I2 = area\_growing(f,100);

I3 = OTSU\_threshold\_segmentation(f);

I4 = average\_threshlod\_segmentation(f);

figure;% 一个2x2显示区域

subplot(2, 2, 1); imshow(I1); title('原图像');

subplot(2, 2, 2); imshow(I2); title('区域生长');

subplot(2, 2, 3); imshow(I3, []); title('OTSU阈值分割图像');

subplot(2, 2, 4); imshow(I4, []); title('均值法阈值分割图像');

grid on

function [outputArg1] = OTSU\_threshold\_segmentation(inputArg1)

%   inputArg1 图像 gray

% outputArg1 = inputArg1;

if size(inputArg1,3)==3

    % 取blue通道，便于分隔车牌

    f = inputArg1(:,:,3);

else

    f = inputArg1;

end

[M,N] = size(f);

T = 0;

result\_T = 0;

gmax = 0;

while (T<=255)

    g=f>=T;

    w0 = numel(f(g))/(M\*N);

    w1 = numel(f(~g))/(M\*N);

    p0 = mean(f(g));

    p1 = mean(f(~g));

    tmp = (w0 \* w1) \* (p0 - p1)^2;

    if (tmp > gmax)

        gmax = tmp;

        result\_T =T;

    end

    T = T+1;

end

[height,width]=size(f);

Tmp = im2bw(f,result\_T/255);

Tmp2 = inputArg1;

if size(inputArg1,3)==3

    for i2 = 1:height

        for j2 = 1:width

            if Tmp(i2,j2) == 0

                % 用于彩色抠像，保留彩色图片

                Tmp2(i2,j2,1)=0;

                Tmp2(i2,j2,2)=0;

                Tmp2(i2,j2,3)=0;

            end

        end

    end

else

    for i2 = 1:height

        for j2 = 1:width

            if Tmp(i2,j2) == 0

                % 用于黑白抠像，保留原始图片

                Tmp2(i2,j2)=0;

            end

        end

    end

end

outputArg1 = Tmp2;

end

function [outputArg1] = average\_threshlod\_segmentation(input\_args)

% inputArg1 image gray

if size(input\_args,3)==3

    % 取blue通道，便于分隔车牌

    f = input\_args(:,:,3);

else

    f = input\_args;

end

T = mean(mean(f));

[height,width]=size(f);

Tmp = im2bw(f,T/255);

Tmp2 = input\_args;

if size(input\_args,3)==3

    for i2 = 1:height

        for j2 = 1:width

            if Tmp(i2,j2) == 0

                % 用于彩色抠像，保留彩色图片

                Tmp2(i2,j2,1)=0;

                Tmp2(i2,j2,2)=0;

                Tmp2(i2,j2,3)=0;

            end

        end

    end

else

    for i2 = 1:height

        for j2 = 1:width

            if Tmp(i2,j2) == 0

                % 用于黑白抠像，保留原始图片

                Tmp2(i2,j2)=0;

            end

        end

    end

end

outputArg1 = Tmp2;

end

function [ output\_args ] = area\_growing( input\_args, thredshold\_input)

%   input\_args raw image as gray format

%   thredshold\_input input thredshold

%   output\_args convert image as gray format

if size(input\_args,3)==3

    % 取blue通道，便于分隔车牌

    I = input\_args(:,:,3);

else

    I = input\_args;

end

% 显示原始图片

figure,imshow(input\_args),title('选择区域生长起始点');

I=double(I);

[height,width]=size(I);

[y\_double,x\_double]=ginput(1);  %获得区域生长起始点

x=round(x\_double);            %横坐标取整

y=round(y\_double);            %纵坐标取整

seed=I(x,y);           %将生长起始点灰度值存入seed中

Tmp = zeros([height,width]);

Tmp(x,y) = 1;

sum = seed;

seed\_save=1;

seed\_count = 1;

thredshold = thredshold\_input;

close all;

while seed\_count>0

    current\_seed\_gray\_level = 0;

    seed\_count = 0;

    for i=1:height

        for j=1:width

            if Tmp(i,j) == 1

                if (i-1)>0 && (i+1)<(height+1) ...

                        && (j-1)>0 && (j+1)<(width+1)%判断该点不在边缘

                    for m=-1:1

                        for n=-1:1

                            if Tmp(i+m,j+n)==0 && ...%首先该点周围的点应该没有被纳入生成的内容中

                                    abs(I(i+m,j+n)-seed) <= thredshold ... %其次，该点的灰度值需要小于我们设置的阈值

                                Tmp(i+m,j+n) = 1; % 成功判定，将内容输出到输出矩阵中

                                seed\_count= seed\_count+1;%种子数量加一

                                current\_seed\_gray\_level = current\_seed\_gray\_level + I(i+m,j+n); %累加灰度，最后求均值

                            end

                        end

                    end

                end

            end

        end

    end

end

seed\_save = seed\_save + seed\_count;%获取所有种子的个数

sum = sum +current\_seed\_gray\_level;

seed = sum/seed\_save;%并计算灰度

Tmp2 = input\_args;

if size(input\_args,3)==3

    for i2 = 1:height

        for j2 = 1:width

            if Tmp(i2,j2) == 0

                % 用于彩色抠像，保留彩色图片

                Tmp2(i2,j2,1)=0;

                Tmp2(i2,j2,2)=0;

                Tmp2(i2,j2,3)=0;

            end

        end

    end

else

    for i2 = 1:height

        for j2 = 1:width

            if Tmp(i2,j2) == 0

                % 用于黑白抠像，保留原始图片

                Tmp2(i2,j2)=0;

            end

        end

    end

end

output\_args = Tmp2;

end