|  |
| --- |
| **数字图像处理实验报告** |
| **实验二：图像压缩编码、形态学处理和分割** |
| |  |  | | --- | --- | | 姓　 名： |  | | 学 科 专 业： |  | | 指 导 教 师： |  | | 班 级： |  | | 学 号： |  | |
| 二○二二年十二月 |

**目录**

[一、 第一题 1](#_Toc15910)

[1、 题目 1](#_Toc19921)

[2、 实验原理及结果 1](#_Toc13341)

[二、 第二题 2](#_Toc3647)

[1、 题目 2](#_Toc12795)

[2、 实验原理及结果 2](#_Toc25807)

[三、 第三题 5](#_Toc5352)

[1、 题目 5](#_Toc719)

[2、 实验原理及结果 5](#_Toc30392)

[四、 实验代码 6](#_Toc31119)

# 第一题

### 题目

读入一幅灰色图像，利用 MATLAB 编程实现采用前值预测（）进一阶无损预测编码，设置=0.8，并显示出解码图像。

### 实验原理及结果

预测编码是建立在信号（语音、图像等）数据的相关性之上，根据某一模型利用以往的样本值对新样本进行预测，减少数据在时间和空间上的相关性，以达到压缩数据的目的。预测方法有多种，本节将着重介绍差分脉冲编码调制的基本原理、最佳线性预测及其自适应编码方法。

一维预测：即用的同一扫描行中的前面已知的几个采样值预测，其预测公式为。实验中设置为0.8

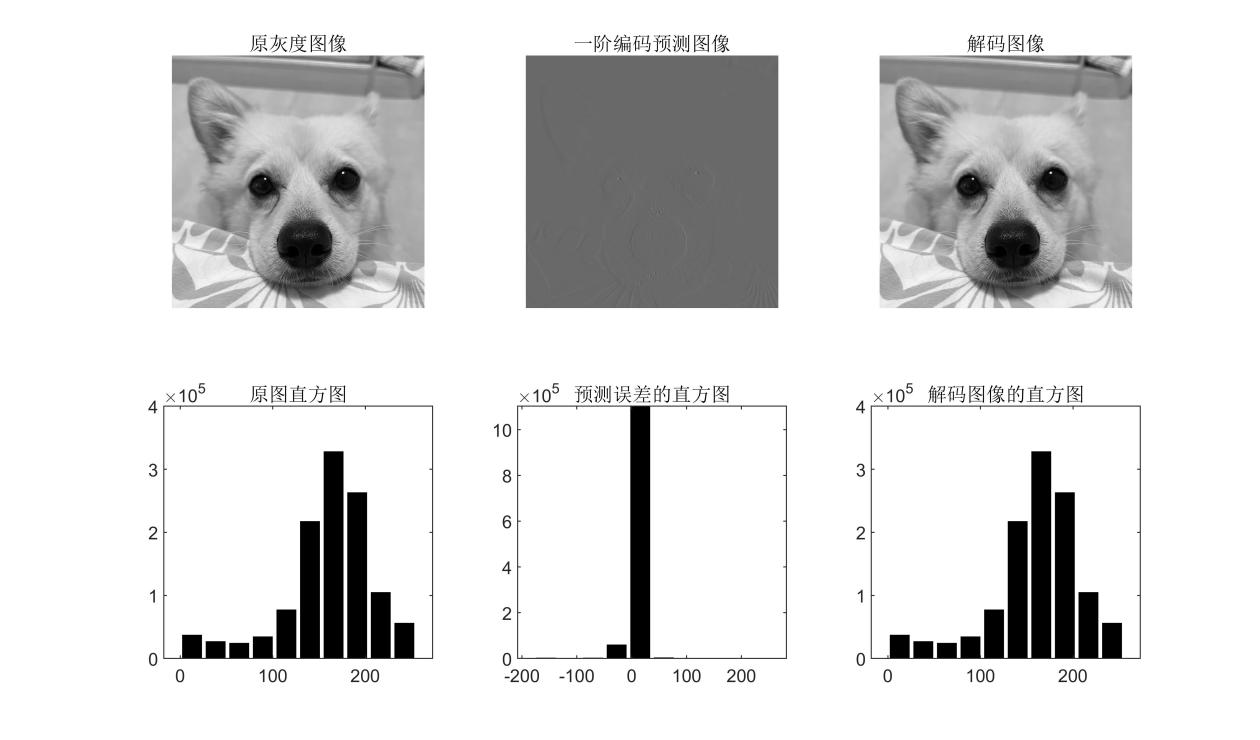


图 1 实验一结果

# 第二题

### 题目

读入一幅带有椒盐噪声的图像，使用 MATLAB 编程进行开和闭运算，要求：1）用二阶单位矩阵的结构元素进行开、闭运算；2）用半径为 1 的平坦圆盘形结构元素进行开、闭运算；3）使用开和闭运算实现形态学滤波；4）显示所有开、闭运算和形态学滤波的结果。

### 实验原理及结果

开运算为先腐蚀后膨胀，对于图像X及结构元素S，用符号X○S表示S对图像X作开运算。

闭运算为先膨胀后腐蚀，对于图像X及结构元素S，用符号XS表示S对图像X作开运算。

如图2所示。开运算可看作将b贴着f 的下沿从一端滚到另一端。对所有比b的直径小的山峰其高度和尖锐度都减弱了。开运算操作消除与结构元素相比尺寸较小的亮细节，而保持图像整体灰度值和大的亮区域基本不受影响。腐蚀去除了小的亮细节并同时减弱了图像亮度，膨胀增加了图像亮度，但又不重新引入前面去除的细节。

闭运算可看作将b贴着f 的上沿从一端滚到另一端。所有比b的直径小的山谷得到了“填充”。闭运算操作消除与结构元 素相比尺寸较小的暗细节，而保持图像整体灰度值和大的暗区域基本不受影响；膨胀去除了小的暗细节并同时增强了图像亮度，腐蚀减弱了图像亮度但又不重新引入前面去除的细节。

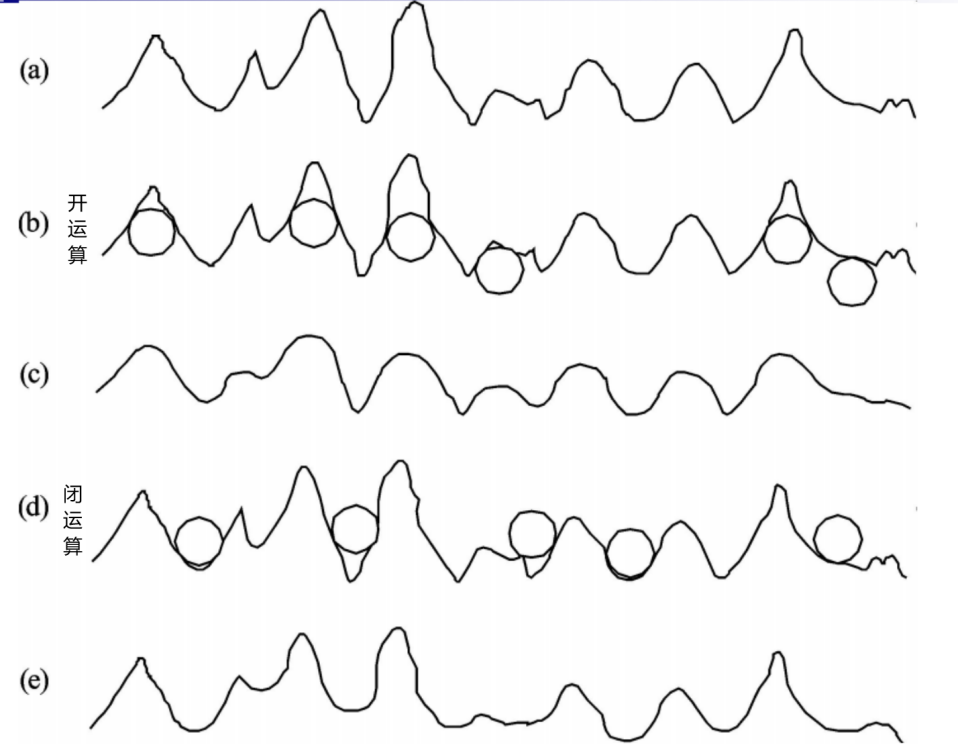


图 2 开运算与闭运算

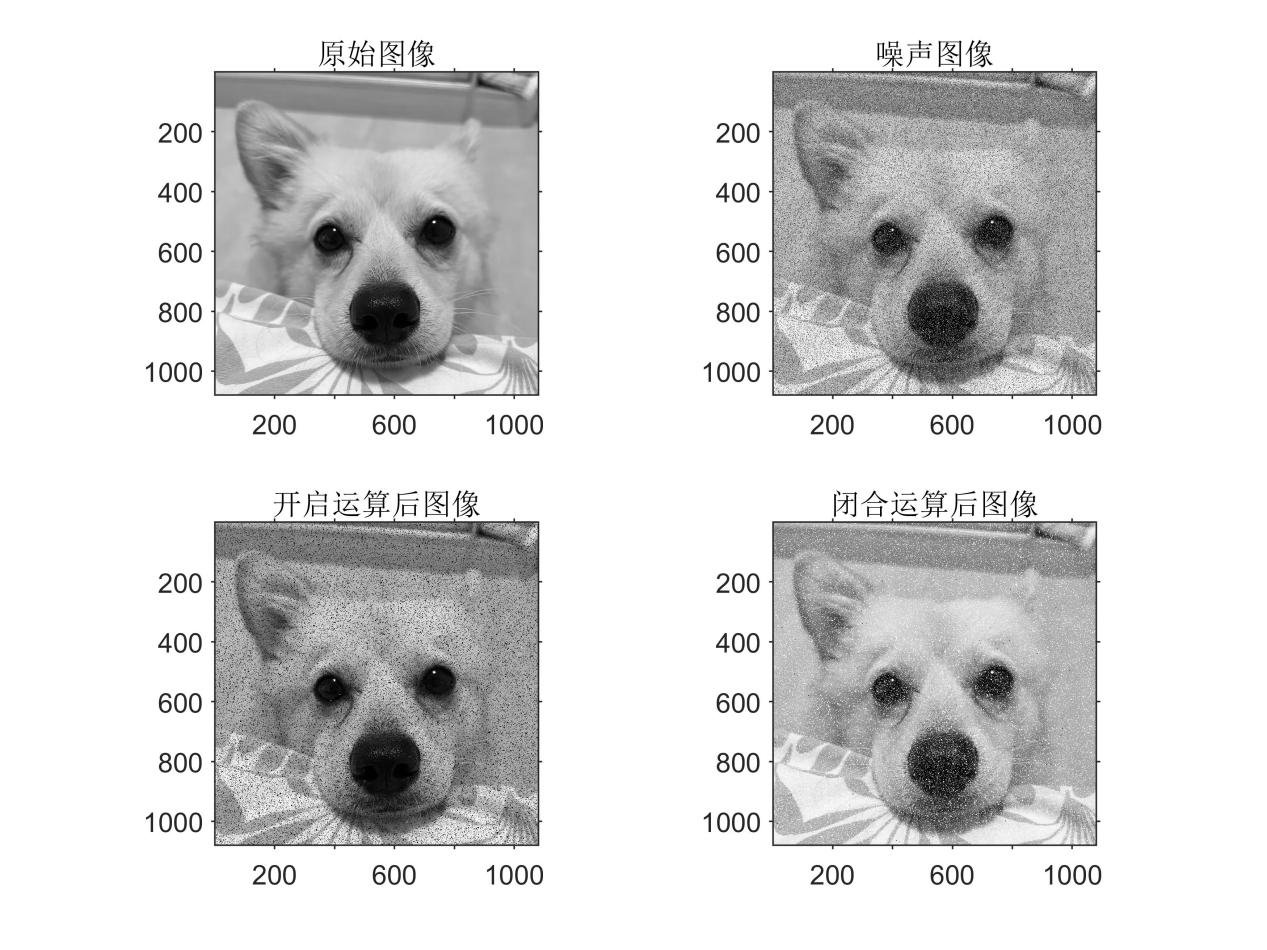


图 3 原始图像、噪声图像、开运算图像以及闭运算图像

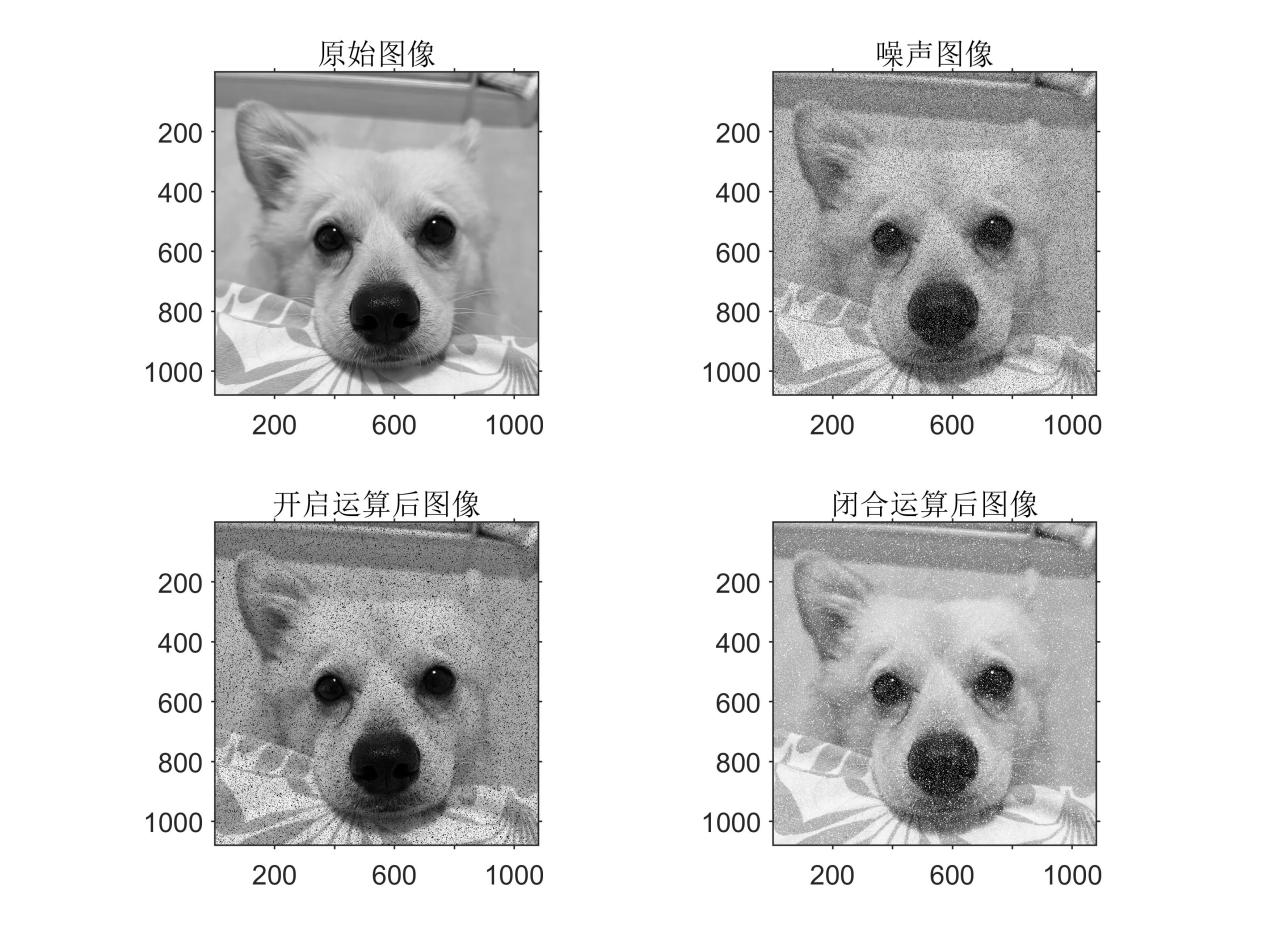


图 4 用半径为1的平坦圆盘形结构元素进行开、闭运算

由于开、闭运算所处理的信息分别与图像的凸、凹处相关， 因此，它们本身都是单边算子，可以利用开、闭运算去除图像的噪声、恢复图像，也可交替使用开、闭运算以达到双边滤波目的。一般，可以将开、闭运算结合起来构成形态

学噪声滤波器。例如或等。

整个过程是先做开运算再做闭运算，可以写为：

O

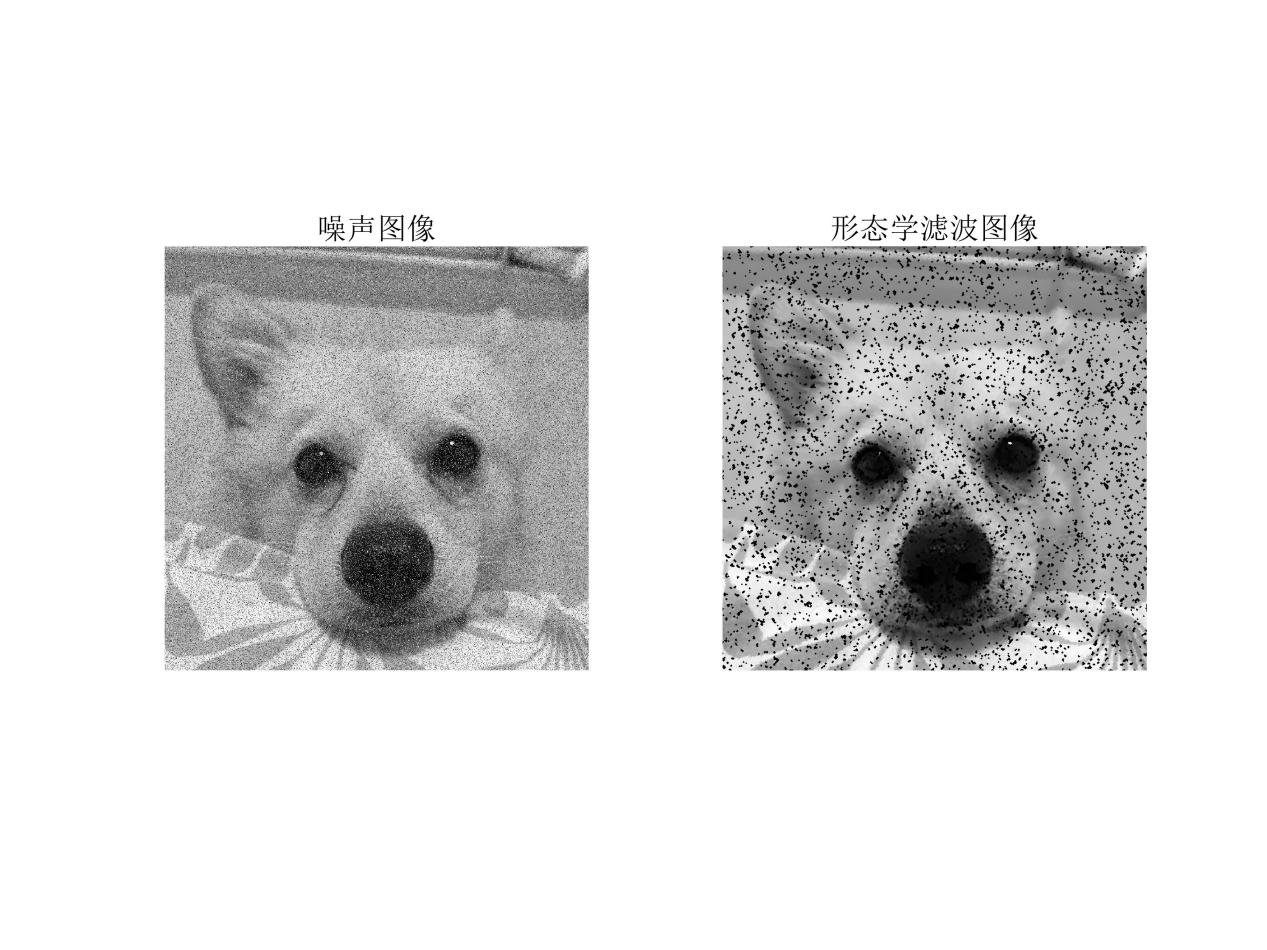


图 5 使用开运算和闭运算实现形态学滤波

# 第三题

### 题目

读入一幅RGB图像，使用MATLAB编程实现最大信息熵阈值分割，并显示图像分割处理后的结果。

### 实验原理及结果

一维最大熵阈值分割。目标区域O的概率分布和背景区域B的概率分布分别是：

背景区域B的概率灰度分布为：

式中：，目标区域和背景区域熵的定义为、

由目标区域和背景区域熵、得到熵函数定义为：

当熵函数取得最大值时，对应的灰度值就是所求的最佳阈值：



图 6 图像分割处理后的结果

# 实验代码

%% 实验一：一阶无损预测编码

% 下面对大小为512\*512像素、灰度级为256的标准lena图像进行无损的一阶预测编码，其matlab程序如下：

rgb\_image = imread('51.png');%原始图片

gray\_image = rgb2gray(rgb\_image);%灰度图片

x=double(gray\_image);

y=LPCencode(x);

xx=LPCdecode(y);

%显示预测误差值

figure(1);

subplot(2,3,1);

imshow(gray\_image);

title('原灰度图像');

subplot(2,3,2);

imshow(mat2gray(y));

title('一阶编码预测图像');

% 计算均方差误差，因为是无损编码，那么erms应该为０

subplot(2,3,3);

imshow(mat2gray(xx));

title('解码图像');

e=double(x)-double(xx);

[m, n]=size(e);

erms = sqrt(sum(e(:).^2)/(m\*n))

% 显示原图直方图

subplot(2,3,4);

[h, f] = hist(x(:));

bar(f, h, 'k');

title('原图直方图');

%显示预测误差的直方图

subplot(2,3,5);

[h, f] = hist(y(:));

bar(f, h,'k');

title('预测误差的直方图');

subplot(2,3,6);

[h, f] = hist(xx(:));

bar(f, h,'k');

title('解码图像的直方图');

%% 实验二：开和闭运算

% 要求1：用二阶单位矩阵的结构元素进行开、闭运算

I1 = imnoise(gray\_image,'salt & pepper',0.2);

subplot(2,2,1),imshow(gray\_image);

title('原始图像');

axis on; %显示坐标系

subplot(2,2,2),imshow(I1);

title('噪声图像');

axis on; %显示坐标系

se=strel('rectangle',[2 2]); %采用二阶单位矩阵的结构作为结构元素

I2=imopen(I1,se); %开启操作

I3=imclose(I1,se); %闭合操作

subplot(2,2,3),imshow(I2);

title('开启运算后图像');

axis on; %显示坐标系

subplot(2,2,4),imshow(I3);

title('闭合运算后图像');

axis on; %显示坐标系

% 要求2：用半径为1的平坦圆盘形结构元素进行开、闭运算

figure

subplot(2,2,1),imshow(gray\_image);

title('原始图像');

axis on; %显示坐标系

subplot(2,2,2),imshow(I1);

title('噪声图像');

axis on; %显示坐标系

se=strel('disk',1); %采用半径为1的平坦圆盘形结构作为结构元素

I2=imopen(I1,se); %开启操作

I3=imclose(I1,se); %闭合操作

subplot(2,2,3),imshow(I2);

title('开启运算后图像');

axis on; %显示坐标系

subplot(2,2,4),imshow(I3);

title('闭合运算后图像');

axis on; %显示坐标系

% 要求3：使用开和闭运算实现形态学滤波

sca = 5; % 结构元素尺寸

se=strel('diamond',(sca-1)/2); % 形态学结构元素

Io=imopen(I1,se); % 开操作

Ioc=imclose(Io,se); % 闭操作

Iobr=imreconstruct(Io,Ioc); % 结构重建

Iobrd = imdilate(Iobr,se); % 膨胀

Iobrcbr = imreconstruct(imcomplement(Iobrd),imcomplement(Iobr)); % 结构重建

Iobrcbr = imcomplement(Iobrcbr);

% 要求4：显示滤波效果

figure

subplot(121),imshow(I1);title('噪声图像')

subplot(122),imshow(Iobrcbr,[]);title('形态学滤波图像')

%% 实验三：最大信息熵阈值分割

clc;clear;

rgb\_image = imread('51.png');

a = rgb\_image;

count=imhist(a);

[m,n]=size(a);

N=m\*n;

L=256;

count=count/N;%%每一个像素的分布概率

for i=1:L

if count(i)~=0

st=i-1;

break;

end

end

for i=L:-1:1

if count(i)~=0

nd=i-1;

break;

end

end

f=count(st+1:nd+1); %f是每个灰度出现的概率

size(f)

E=[];

for Th=st:nd-1 %%%设定初始分割阈值为Th

av1=0;

av2=0;

Pth=sum(count(1:Th+1));

%%%第一类的平均相对熵为

for i=0:Th

av1=av1-count(i+1)/Pth\*log(count(i+1)/Pth+0.00001);

end

%%%第二类的平均相对熵为

for i=Th+1:L-1

av2=av2-count(i+1)/(1-Pth)\*log(count(i+1)/(1-Pth)+0.00001);

end

E(Th-st+1)=av1+av2;

end

position=find(E==(max(E)));

th=st+position-1

for i=1:m

for j=1:n

if a(i,j)>th

a(i,j)=255;

else

a(i,j)=0;

end

end

end

figure

subplot(1,2,1)

imshow(rgb\_image);

title('原始图像');

subplot(1,2,2)

imshow(a);

title('最大信息熵阈值分割图像');