区域形状分析

2018年4月16日

目录

1 题目介绍

本题为B类题，题目要求为：图像中含有大小不同的两类目标，请将各个目标分割出来，然后对分割的目标，利用合适的特征，将其进行分类。图片如图1所示。

图 1: 题目图片

1.1 分水岭算法简介

分水岭分割方法，是一种基于拓扑理论的数学形态学的分割方法，其基本思想是把图像看作是测地学上的拓扑地貌，图像中每一点像素的灰度值表示该点的海拔高度，每一个局部极小值及其影响区域称为集水盆，而集水盆的边界则形成分水岭。分水岭的概念和形成可以通过模拟浸入过程来说明。在每一个局部极小值表面，刺穿一个小孔，然后把整个模型慢慢浸入水中，随着浸入的加深，每一个局部极小值的影响域慢慢向外扩展，在两个集水盆汇合处构筑大坝，即形成分水岭。

分水岭的计算过程是一个迭代标注过程。分水岭比较经典的计算方法是L. Vincent提出的。在该算法中，分水岭计算分两个步骤，一个是排序过程，一个是淹没过程。首先对每个像素的灰度级进行从低到高排序，然后在从低到高实现淹没过程中，对每一个局部极小值在h阶高度的影响域采用先进先出(FIFO)结构进行判断及标注。

1.2 题目分析

本题的做法是：基于marker-controlled watershed进行分割,然后对分割的目标，选取特征进行Kmean聚类。marker-controlled watershed，即标记控制的分水岭算法，是一种主流的分割算法，其思想是利用一些附加知识，在原图中寻找一些内部标记和外部标记来引导算法进行分割，防止过分割。算法要求每一个内部标记要处在感兴趣物体的内部，而外部标记符要包含在背景中。然后利用找到的内外标记来改进梯度图像，在改进的梯度图像上应用分水岭变换，得到最终的分割结果。

这种算法的主要流程如下：

计算分割函数。分割函数是一个图像，这个图像的较暗的区域是你希望分割的目标。

计算前景标记。前景标注是每一个目标中的一个连接的区域。

计算背景标记。背景是一个不属于任何目标的部分。

修改分割函数。分割函数仅仅在前景和背景标注位置具有较小值。

算修改后分割函数的分水岭变换。

2 算法实现及结果

在MATLAB提供的marker-controlled watershed例程的基础上，经过不断修改和完善，最终完成了这个题目。

2.1 读取图像

首先，调取MATLAB的imread函数读取图像。再调取rgb2gray函数将rgb图像转换为灰度图，结果如图2。由图可知，图像中还有很多小的连通区域，这些区域的存在会使图像产生严重的过分割。为此，笔者利用高帽变换和低帽变换叠加使用来增强对比度。高帽变换和低帽变换可直接调用MATLAB的imtophat函数和imbothat函数。如图3，得到经过高帽变换和低帽变换的灰度图像。

2.2 分割函数

该算法使用梯度幅值作为分割函数。使用Sobel边缘检测算子，imfilter函数和一些简单的

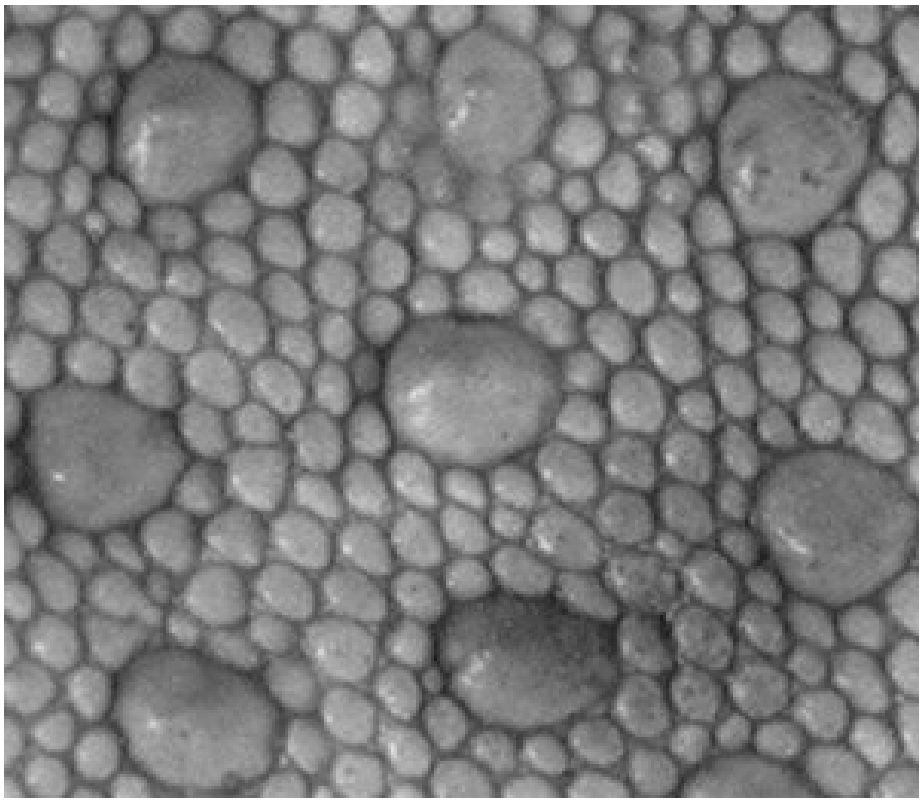

图 2: 灰度图像

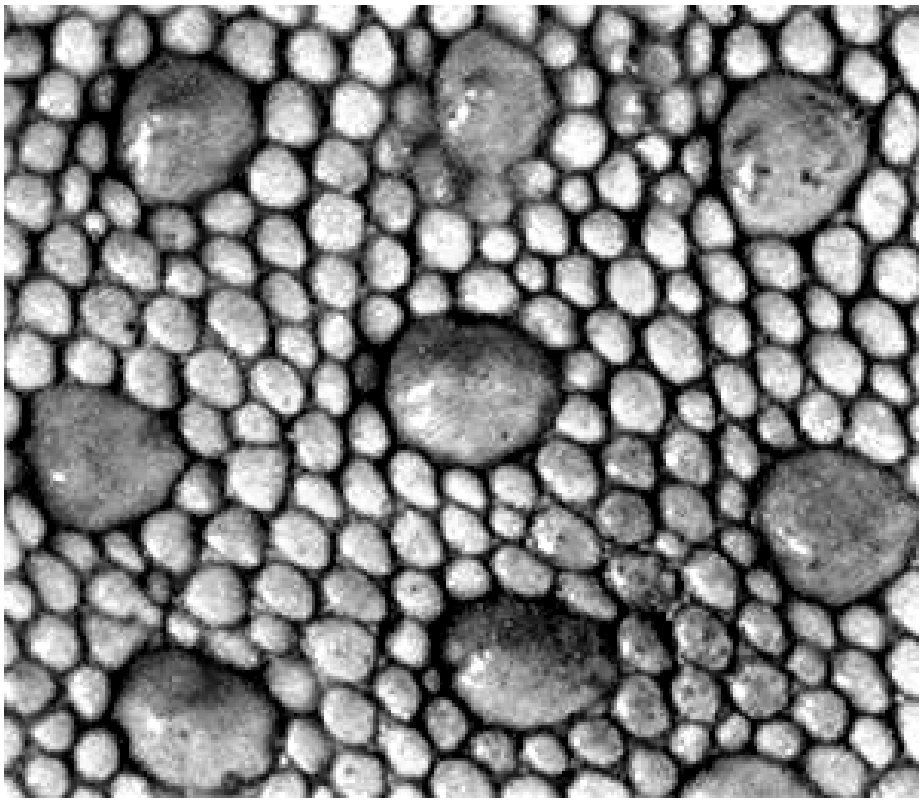

图 3: 高帽变换和低帽变换后的灰度图像

四则运算来计算幅值。梯度总值总是在物体的边缘处较高，在物体的内部较低。如图4为梯度幅值图。值得注意的是，如果在进行梯度幅值运算后不进行额外的预处理，直接进行分水岭变换来分割图像，则会导致严重的过分割，如图5所示。

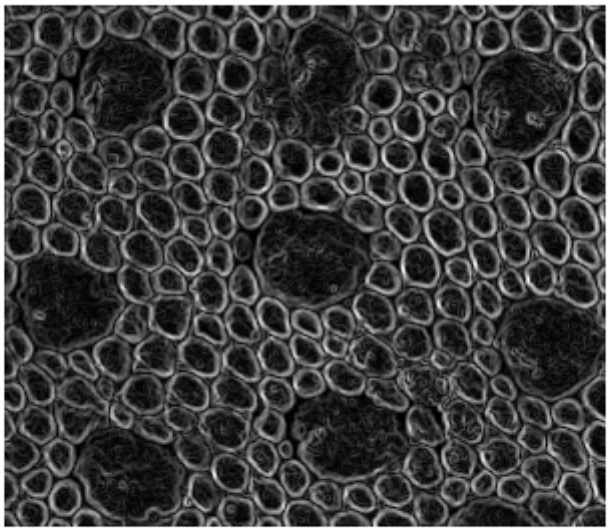

图 4: 梯度图像

2.3 标记前景物体

对标记符的选择，本算法有两个基本的假设，基于这两个基本假设，可以设计出相应的寻找标识符的方法：

假设一： 目标物体是相对较亮的物体，背景对应暗的区域。如果背景区域比较暗，就可以利用二值化，找到背景大致对应的区域，从而确保外部标记符包含在背景区域中。

假设二： 目标物体或其一部分在其领域内是极大区域。由于前景物体经常位于其他物体前面处于突出位置，其很可能是局部区域内像素值最大的区域。基于此，可以寻找图像中局部极大值区域作为内部标记符。对前景进行标记有多种方法，本题中使用的是形态学中“基于重建的开启操作”和“基于重建的关闭操作”，其作用是清理图像，在每个对象内部创建平坦的极大值小斑块。先腐蚀后膨胀为开启操作，可以把比

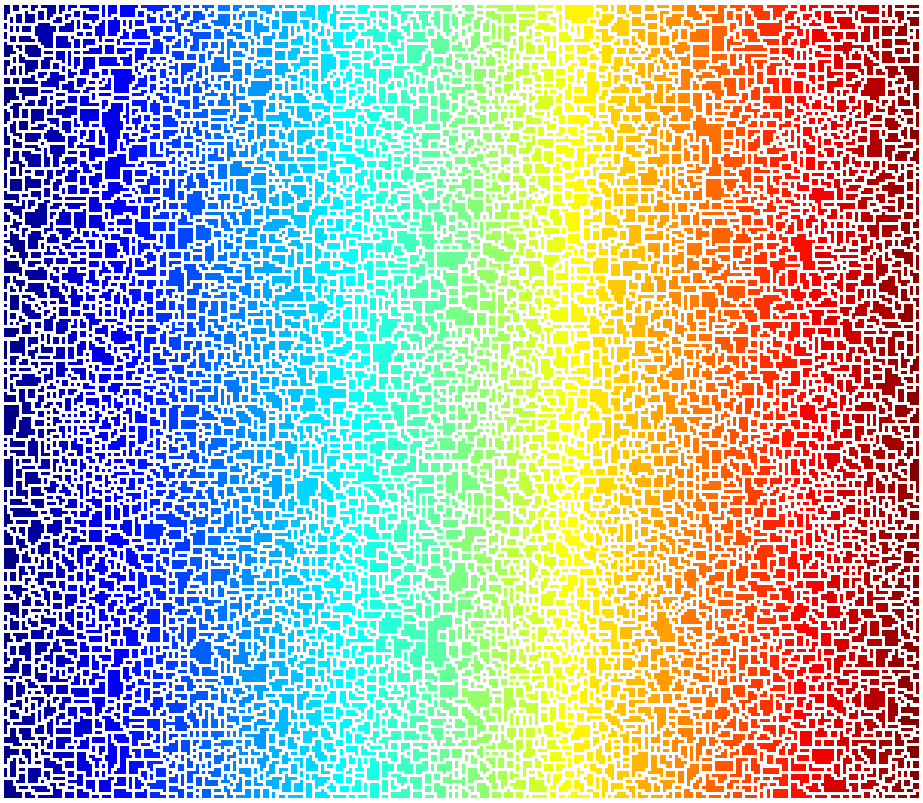

图 5: 过分割

结构元素小的突刺滤掉，切断细长搭接而起到分离作用；先膨胀后腐蚀为关闭操作，可以把比结构元素小的缺口或孔填充上，搭接短的间隔而起到连接作用。开启和关闭可以去除比结构元素小的特定图像细节，同时保证不产生全局几何失真。开启操作的第一步腐蚀能去除白色小物体，随后的膨胀趋向于恢复保留下来的物体的形状，这种恢复是不精确的，其精确度取决于形状和结构体之间的相似性。基于重建进行的开启操作能够准确的恢复腐蚀之后的物体形状。所以基于重建的开启操作与单纯的开启操作功能类似，但更好地保留了原物体的形状。关闭操作同理。首先，利用strel函数构建结构元素，即模板。经过不断尝试，模板的参数最终选为'disk'圆形，大小为5。调用imerode函数进行腐蚀，再将腐蚀后的结果利用imreconstruct函数进行重建。得到开启的结果后再将此结果进行关闭操作。调用imdilate函数进行膨胀，再将膨胀的结果进行重建，最后将重建结果取补。如图6所示，即得到实际基于重建的开启-关闭操作的结果。

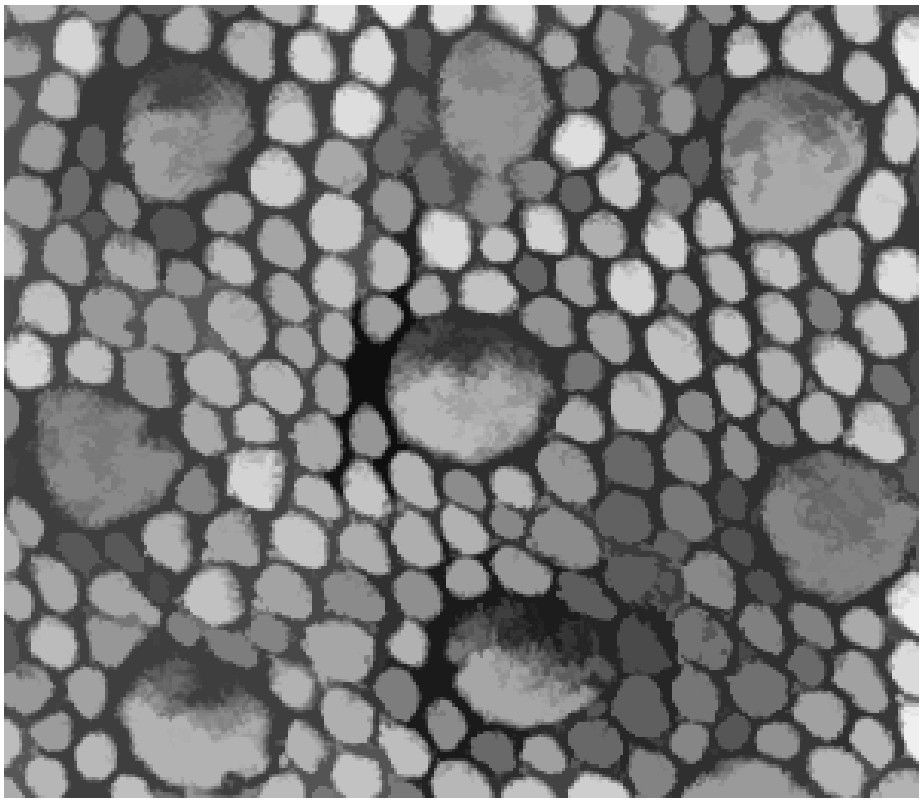

图 6: 开启-关闭操作结果

为了标记出前景，再构建一个圆形，但大小为2的模板。注意此处的模板大小会影响到

最终结果中小石头的识别效果，此问题会在后文单独讨论。利用此模板对开启-关闭的结果使用imregionalmax取极大值，再进行关闭操作，最终得到前景标记的二值图，如图7所示。 注

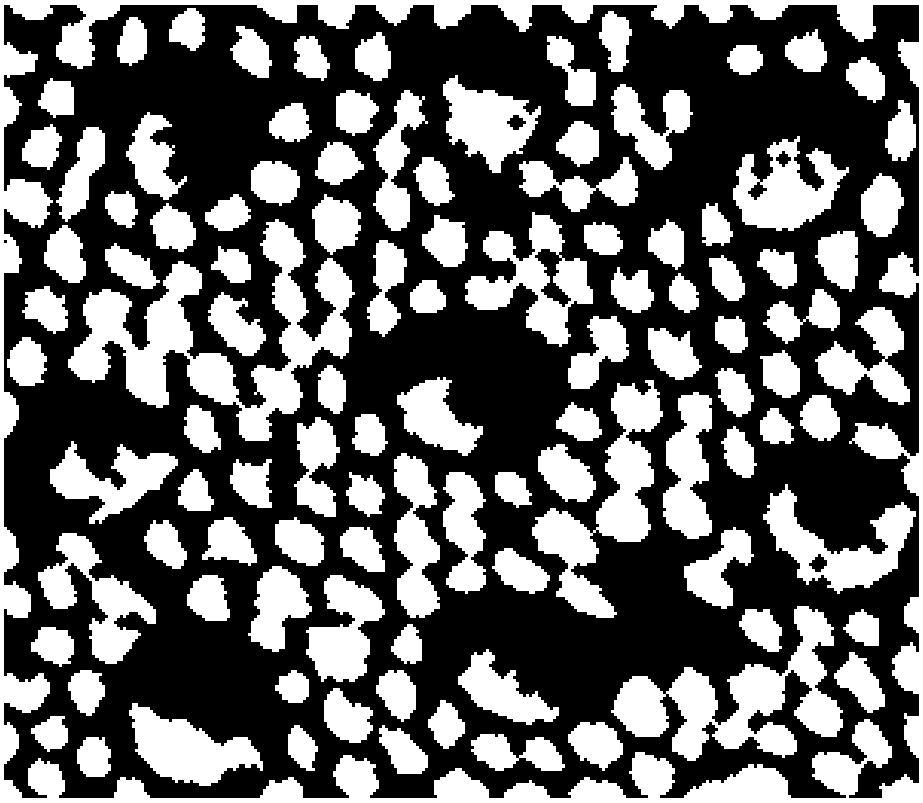

图 7: 前景标记

意到一些大部分重合或被阴影遮挡的物体没有被标记出来。这意味着这些物体最终可能不会被正确的分割出来。并且，有些物体中前景标记正确的到达了物体的边缘。应该清除掉标记斑块的边缘，向内收缩一点。可以通过先闭操作，再腐蚀做到这点。这个操作会导致遗留下一些离群的孤立点，这些是需要被移除的。可以通过bwareaopen做到这点，这个函数将移除那些包含像素点个数少于指定值的区域。

2.4 计算背景标记

对背景标记同样进行如计算前景时的操作，先开启重建，再关闭重建。得到结果后通过大津二值法，即OTSU法进行二值化。然后，对二值化的结果进行中值滤波和开启操作，最后调用imfill填充小的连通区域。代码如下，结果如图8。

```
bw1 = im2bw(Iobrcbrbw, graythresh(Iobrcbrbw));
bw2 = medfilt2(bw1);
bw = imopen(bw2, se);
bw = imfill(bw,'holes');
```

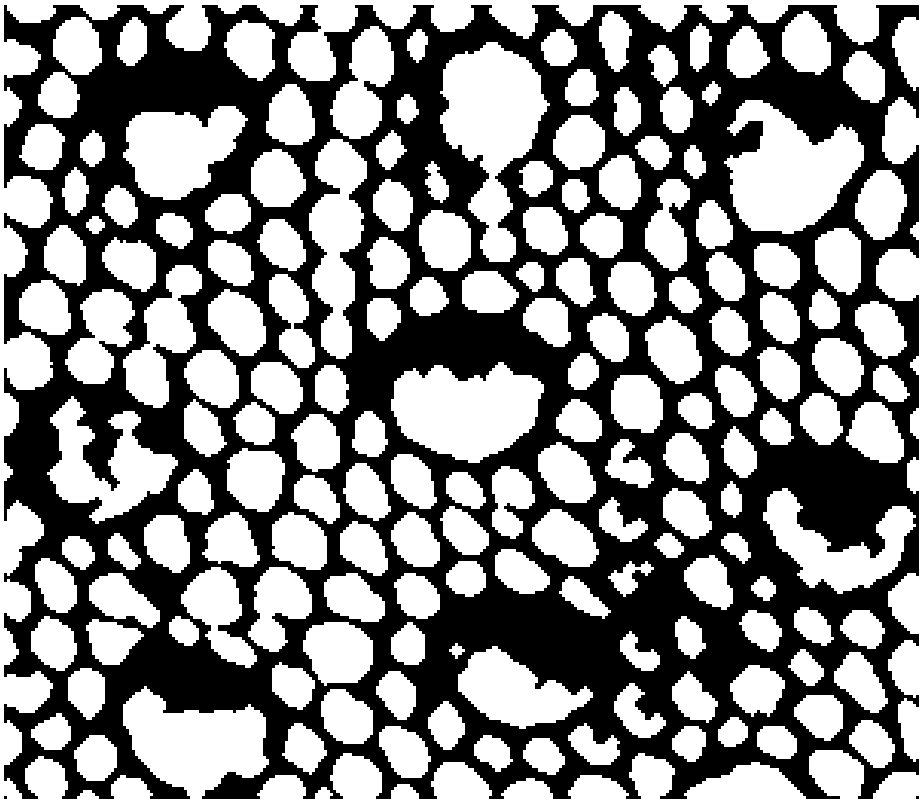

图 8: 背景标记

通过对背景标记的距离变换进行分水岭变换，寻找分水岭脊线，结果如图9。

2.5 分水岭变换

调用imimposemin函数对梯度幅度进行修改，使得其只在指定的位置处取得局部最小值，指定位置即为前景标记和背景标记处。再调用MATLAB提供的分水岭函数watershed得到分割图像。代码如下。

```
gradmag2 = imimposemin(gradmag, bgm | fgm4);
Lrow = watershed(gradmag2);
```

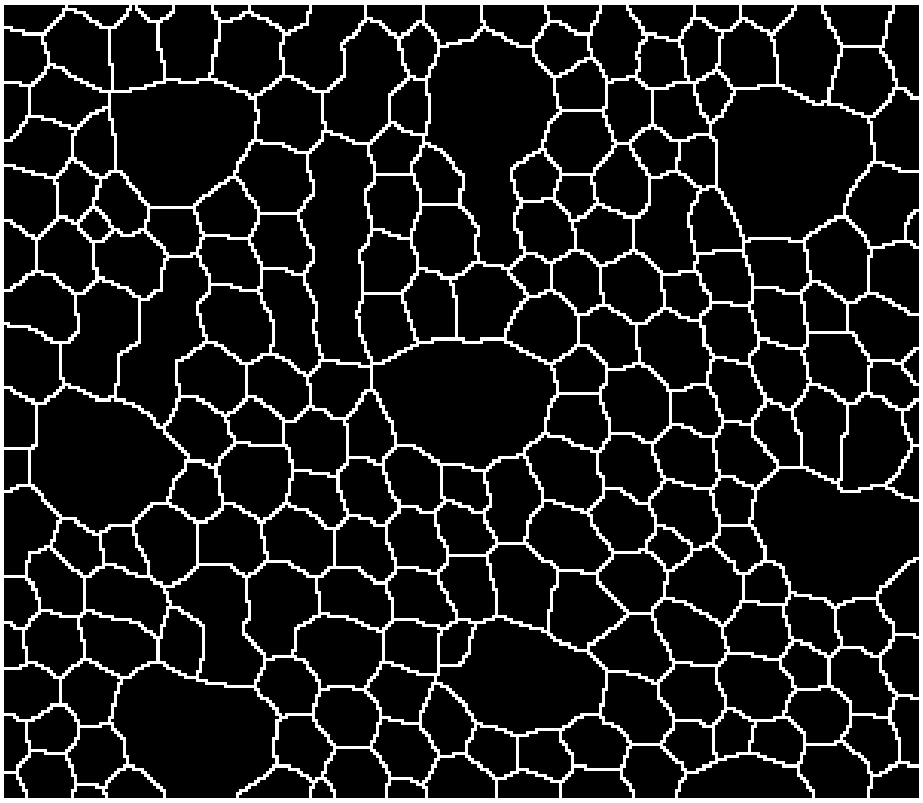

图 9: 分水岭脊线

2.6 k-means聚类

对不同大小的区域进行分类使用的是k-means降类的方法。首先，调用regionprops函数计算被识别出来的不同区域的面积，再对面积做k-means聚类。得到聚类结果后将传入可视化程序中输出为图像。

2.7 数据可视化

利用例程中的可视化程序，可以输出分割和分类后的图像，如图10, 11。

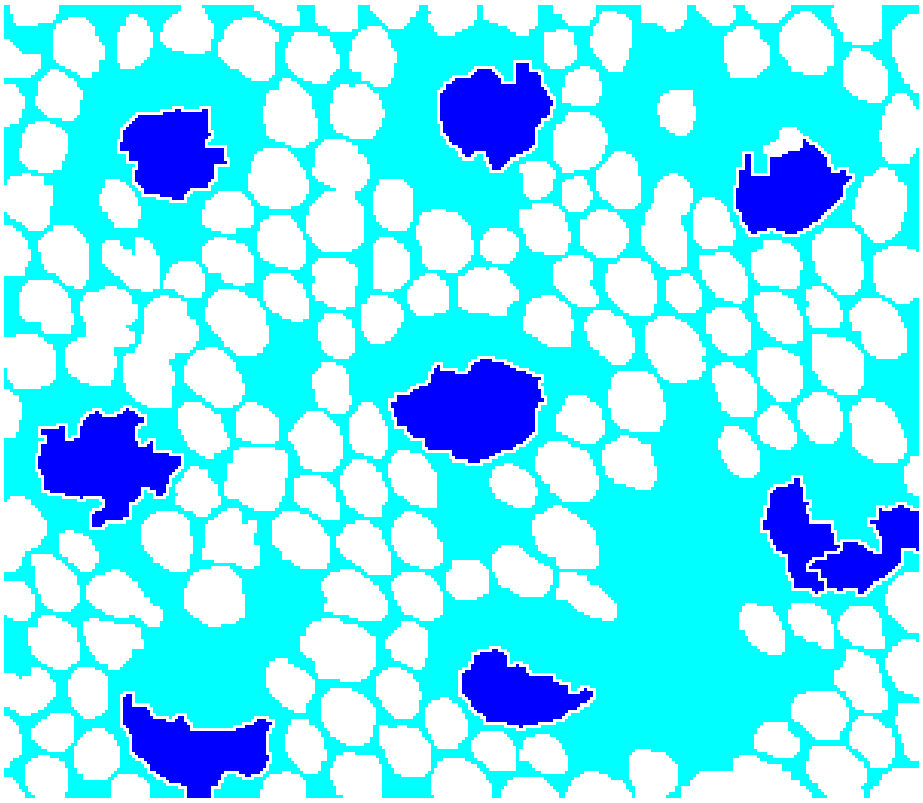

图 10: 分割分类示意

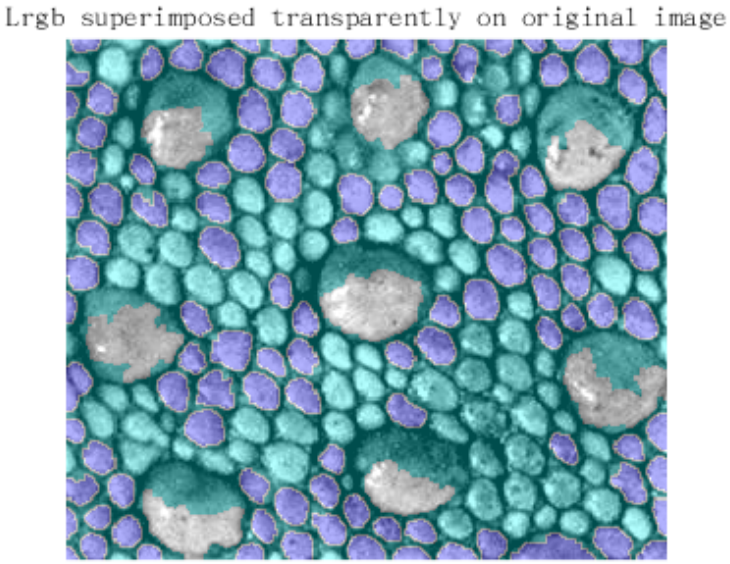

图 11: 分割分割后在原图上的叠加

3 讨论

3.1 前景模板大小

前文提及的在前景标注中，se3的模板大小的选取会对小石头的识别产生影响。当模板大小为2时可识别出比较少的小石头，但大右下侧的大石头正常识别为一个，如图12所示。而当模板大小为1时，可较好识别小石头，但右下侧的大石头会被识别为两个，如图13所示。这主要是因为在前景标注时右下侧的大石头由于光照条件较为不同，二值化时出现了分裂，最终导致在k-means聚类中被分为两类。

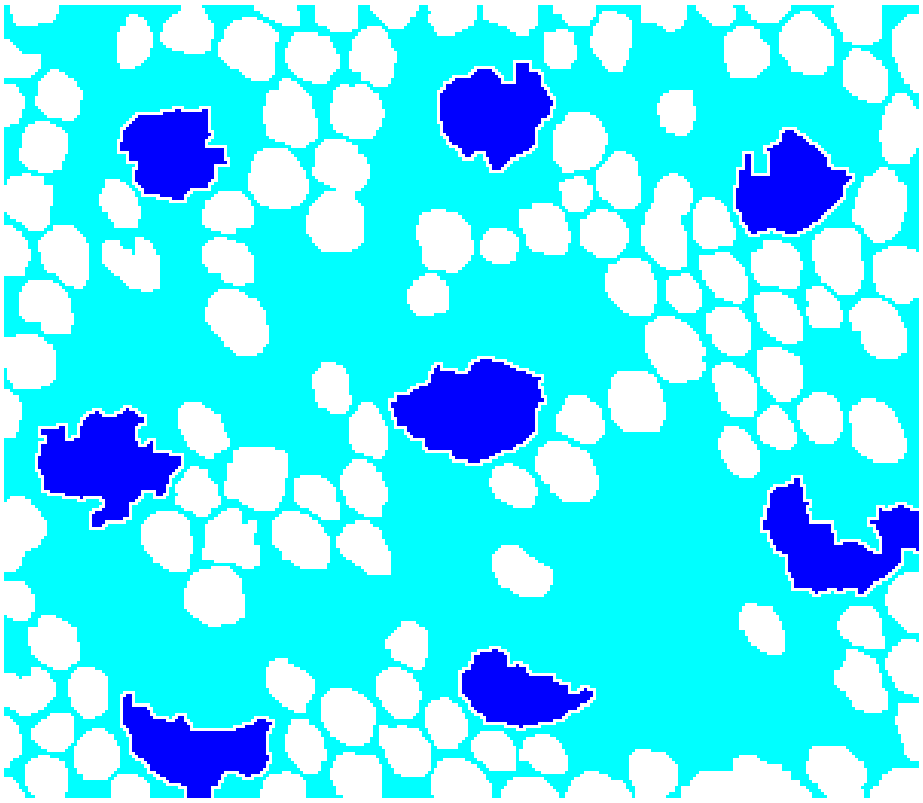

图 12: 多小石头

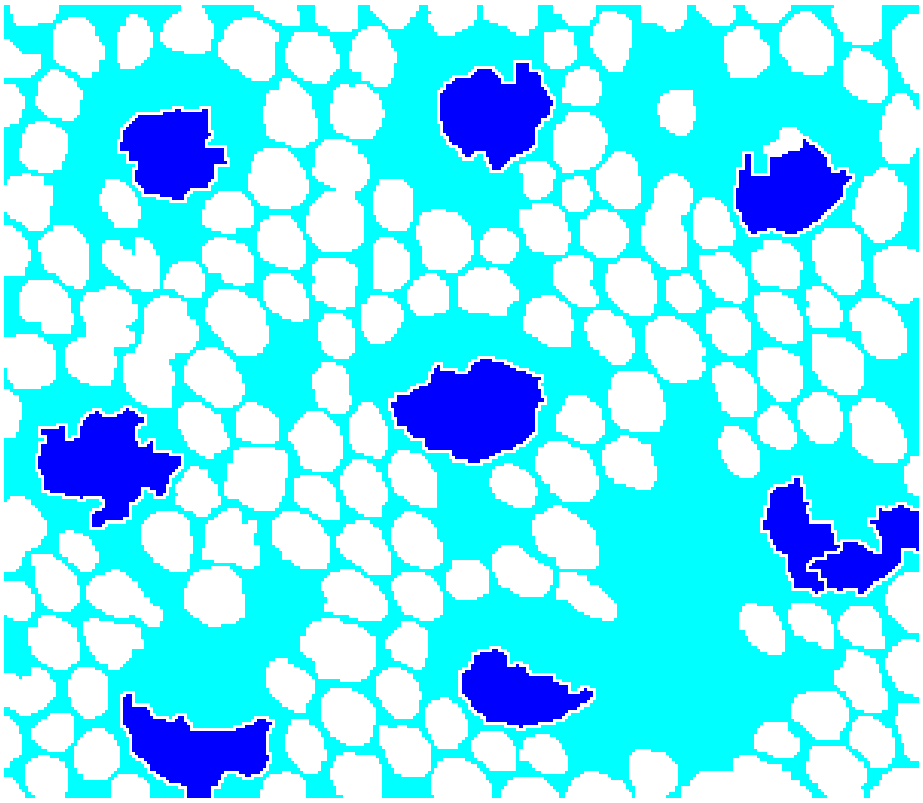

图 13: 大石头分裂

3.2 右下角的半个石头

在原图中，我们可以观察到在下边界靠右侧的位置有半个石头。基于人类的判断，那个位置是被分为大石头的类别，但是k-means聚类的结果显示那个部分被分为了小石头。这主要是由于，那个部分的面积较小，而k-means聚类的主要指标为面积，没有基于石头边界形状聚类。所以，若想使那个位置被识别为大石头，则要增加一种聚类的指标，如石头边缘的曲率。或者将原图像利用插值的方法扩展，使右下的半个大石头变为整个石头，再进行k-means聚类。

4 总结

此次大作业收获颇多，学习了一种图像分割的算法，知道了很多在理论课中的出现的概念在MATLAB中的应用。虽然大作业最终的结果不甚满意，但最终在自己的努力下基本完成要求。最后，感谢两位老师一个学期的教导，和两位助教的付出。

A MCwatershedSEG.m

```
clear;
close all;
clc;图像标题会错位，不知什么原因

%

%% 1 read image rgb2gray

rgb = imread('gecko.bmp');

Ir2g = rgb2gray(rgb);
% Ir2g = medfilt2(rgb2gray(rgb))转化灰度图后进行中值滤波，效果比未滤波有
    明显提升;%
% I = imhmin(Img, 1);
imwrite(Ir2g, 'gray.png');

se1 = strel('disk', 15); 构建多尺度%高帽，低帽变换，增强亮处和暗处
%
Itop = imtophat(Ir2g, se1);
% figure
% imshow(Itop), title('Itop')
Ibot = imbothat(Ir2g, se1);
% figure
% imshow(Ibot), title('Ibot')
I = imsubtract(imadd(Itop, Ir2g),Ibot);
imwrite(I, 'gtb.png');
imshow(I)
% imshow(Img)
text(732,501,'Image courtesy of Corel',...
    'FontSize',7,'HorizontalAlignment','right')

%% 2 use the gradient magnitude as the segmentation function
```

```
hy = fspecial('sobel'); 索贝尔算子，计算纵向梯度%
% hy = fspecial('prewitt');
%'gaussian', 'sobel', 'prewitt', 'laplacian', 'log', 'average',
   'unsharp', 'disk', 'motion'

hx = hy';
Iy = imfilter(double(I), hy, 'replicate');
Ix = imfilter(double(I), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2); %gradient magnitude
% imwrite(gradmag, 'gradmag.png');
figure
imshow(gradmag, [])%, title('Gradinet magnitude (gradmag)')

%% oversegmentation 不进行其它处理会过分割

oversegL = watershed(gradmag);
oversegLrgb = label2rgb(oversegL);
% figure, imshow(oversegLrgb), title('watershed transform
   overseg')
imwrite(oversegLrgb,'overseg.png');

%% 3 foreground 标注前景目标

se = strel('disk', 5);构建结构元素，即模板。例子中的参数（


%'disk'）不行，关闭操作图像左半边就没了,20


% 使用%模板，多尺度se1 figure3
% 直接调用开启函数的方法%
% Io = imopen(I, se1);
% figure
```

```
% imshow(Io), title('Opening(Io)')
%
% 使用%模板，多尺度se1
% % 直接调用关闭函数的方法
% Ioc = imclose(Io, se1);
% figure
% imshow(Ioc), title('Opening-closing (Ioc)')




% 基于重建的开启。使用和imerodeimreconstruct
%figure4
% Ie = imerode(Itop, se)腐蚀，重建;%
% Iobr = imreconstruct(Ie, Itop);
Ie = imerode(I, se)腐蚀，重建;%
Iobr = imreconstruct(Ie, I);
figure
imshow(Iobr), title('Opening-by-reconstruction (Iobr)')基于重建的关
    闭



%
Iobrd = imdilate(Iobr, se)膨胀;%
Iobrcbr = imreconstruct(imcomplement(Iobrd), imcomplement(Iobr))
    重
    建;%
Iobrcbr = imcomplement(Iobrcbr)重建结果取补;%
figure
imshow(Iobrcbr), title('Opening-closing by reconstruction (
    Iobrcbr)')
imwrite(Iobrcbr, 'o-c.png');修改此处识别大的（），小的（）
```

```
%21
se3 = strel('disk', 2);
% se3 = strel('disk', 1);计算
%的区域极大值从而得到好的前景标记Iobrcbr
fgm = imregionalmax(Iobrcbr)二值图，白色为前景;%
fgm = imclose(fgm, se3);
% fgm = imdilate(fgm, se4);
figure
imshow(fgm), title('Regional maxima of opening-closing by
    reconstruction (fgm)')
imwrite(fgm, 'fgm.png');将前景标记图叠加到原始图像

%
I2 = I;
I2(fgm) = 将255;%中的前景区域（像素值为）标记到原图上（置白色）fgm1
figure
imshow(I2), title('Regional maxima superimposed on original
    image (I2)')注意到一些大部分重合或被阴影遮挡的物体没有被标记出来。这意味着
    这些物体最终可能不会被正确的分割出来。


%并且，有些物体中前景标记正确的到达了物体的边缘。这意味着你应该清除掉标记斑块的边
    缘，向内收缩一点。
%可以通过先闭操作，再腐蚀做到这点。
%
se2 = strel(ones(3,3));
fgm2 = imclose(fgm, se2);
fgm3 = imerode(fgm2, se2);这个操作会导致遗留下一些离群的孤立点，这些是需要
    被移除的。

%你可以通过
%做到这点，函数将移除那些包含像素点个数少于指定值的区域bwareaopen
fgm4 = bwareaopen(fgm3, 1);
I3 = I;
```

```
I3(fgm4) = 255;
figure
imshow(I3)
title('Modified regional maxima superimposed on original image (
    fgm4)')


%% 4 计算背景标注背景标记

%
sebw = strel('disk', 1);
Iebw = imerode(I, sebw)腐蚀, 重建;%
Iobrbw = imreconstruct(Iebw, I);
Iobrdbw = imdilate(Iobrbw, sebw)膨胀;%
Iobrcbrbw = imreconstruct(imcomplement(Iobrdbw), imcomplement(
    Iobrbw))重
    建;%
Iobrcbrbw = imcomplement(Iobrcbrbw)重建结果取补;%原算法的前提假设是: 图
    像中相对亮的是物体, 相对暗的是背景



%二值化
%
% bw = imbinarize(Iobrcbr);

% bw = imregionalmin(Iobrcbr)计算图像中大量局部最小区域的位置;%
% bw = imextendedmin(Iobrcbr,15); 计算图像中比周围点更深的点的集合（通过
    某个高度阈值）%


se = strel('disk', 2);



bw1 = im2bw(Iobrcbrbw, graythresh(Iobrcbrbw)); 这种二值化方法也可以,
```

```
    大津法%otsu
bw2 = medfilt2(bw1);
bw = imopen(bw2, se);
% bw3 = imerode(bw2, se);
% bw3 = imdilate(bw3, se);

% bw4 = imreconstruct(bw3,bw2);
bw = imfill(bw,'holes'); 填洞%

figure
imshow(bw), title('Thresholded opening-closing by reconstruction
    (bw)')
imwrite(bw, 'bw.png');通过对

%的距离变换进行分水岭变换，寻找分水岭脊线（bwDL）==0
D = bwdist(bw)距离变换;%
DL = watershed(D);
bgm = DL == 区域分界线0;%
figure
imshow(bgm), title('Watershed ridge lines (bgm)')
imwrite(bgm, 'edgebgm.png');

%% 5 计算修改后的分割函数的分水岭变换

gradmag2 = imimposemin(gradmag, bgm | fgm4);

Lrow = watershed(gradmag2); 调用分水岭函数%

%% 分类kmeans

areastr = regionprops(Lrow, 'Area');
areacell = struct2cell(areastr);
areamat = cell2mat(areacell);
[m n] = size(areamat);
```

```
[idx, C] = kmeans(areamat(1, 2:n)',2);
[a b] = sort(idx, 'descend');
[m1 n1] = size(Lrow);
zero1 = zeros(size(Lrow));
zero2 = zero1;
numbig = sum(a == size(C, 1));
for i = 1:numbig
    c(i) = b(i) + 1;
    zero1(Lrow == c(i)) = 2;
end
zero2(Lrow == 1) = 1;
L = zero1 + zero2;


%% 6 可视化结果
I4 = I;
I4(imdilate(L == 0, ones(3, 3)) | bgm | fgm4) = 255;
figure
imshow(I4)
title('Markers and object boundaries superimposed on original
   image (I4)')

Lrgb = label2rgb(L, 'jet', 'w', 'shuffle');
figure
imshow(Lrgb)
title('Colored watershed label matrix (Lrgb)')
imwrite(Lrgb, 'Lrgb_big.png');

figure
imshow(I)
hold on
himage = imshow(Lrgb);
himage.AlphaData = 0.3;
title('Lrgb superimposed transparently on original image')
```