

数字图像分析第三次实验报告

院 系： 信息科学技术学院

学 号： SA19023005

姓 名： 景军元

授课教师： 李厚强、周文罡

课程助教： 欧阳剑波、刘一衡、林丰

实验名称： 基于水平集的图像分割

一、基本原理

1、曲线演化

曲线演化方程： $\frac{\partial C}{\partial t} = FN$ ，其中 F 为速度函数， N 为曲线 C 的法向量。

2、水平集演化

水平集演化方程： $\frac{\partial \phi}{\partial t} = F|\nabla \phi|$ 。

二、实验内容

1、问题重述

请基于如下演化方程，用 MATLAB 代码实现图像分割：

$$\frac{\partial \phi}{\partial t} = \mu \left[\Delta \phi - \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] + \lambda \delta(\phi) \operatorname{div} \left(g \frac{\nabla \phi}{|\nabla \phi|} \right) + \nu g \delta(\phi)$$

其中：

$$g = \frac{1}{1 + |\nabla G_{\sigma} * I|^2},$$

$$\delta_{\varepsilon}(x) = \begin{cases} 0, & |x| > \varepsilon \\ \frac{1}{2\varepsilon} \left[1 + \cos \left(\frac{\pi x}{\varepsilon} \right) \right], & |x| \leq \varepsilon. \end{cases}$$

附件中包含另外一个水平集模型(如下演化方程)的 MATLAB 代码供参考(主程序文件为 Test_demo.m)，可在其基础上修改，实现上述水平集分割算法。

$$\frac{\partial \phi}{\partial t} = \delta_{\varepsilon}(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2 \right]$$

2、函数实现

(1) Test_demo.m

①源代码

```
clear;
```

```
close all;
```

```
clc;
```

```
Img=imread('three.bmp'); % example that CV model works well
```

```
% Img=imread('vessel.bmp'); % example that CV model does NOT work well
```

```
% Img=imread('twoCells.bmp'); % example that CV model does NOT work well
```

```
U=Img(:,:,1);
```

```
I=double(U);
```

```
% get the size
```

```
[nrow,ncol]=size(U);
```

```

c0=3;
initialLSF=c0*ones(size(U));
initialLSF(5:nrow-5, 5:ncol-5)=-c0;
phi_0=initialLSF;
figure; mesh(phi_0); title('Signed Distance Function')

delta_t = 5; %time step
mu = 0.2/delta_t; %distRictTerm coefficient
nu = 1.5; % areaTerm coefficient
lambda=5; % legthTerm coefficient

epsilon=1.5; %used in Dirac function
sigma=1.5; %gaussian lowpass filter standard deviation
g=edge_detector(I,sigma);%公式中的 g, 边缘检测子
[gx,gy]=gradient(g);%g 的梯度会在每次循环中用到

% iteration should begin from here
phi=phi_0;
figure(2);
subplot(1,2,1); mesh(phi);
subplot(1,2,2); imagesc(uint8(I));colormap(gray)
hold on;
plotLevelSet(phi,0,'r');

numIter = 1;
for k=1:200
    phi = evolution_cv(I, phi,g,gx, gy, mu, nu, lambda, delta_t, epsilon, numIter);    %
    update level set function
    if mod(k,5)==0
        pause(.05);
        figure(2); clc; axis equal;
        title(sprintf('Iteration times: %d', k));
        subplot(1,2,1); mesh(phi);
        subplot(1,2,2); imagesc(uint8(I));colormap(gray)
        hold on; plotLevelSet(phi,0,'r');
    end
end

```

②函数说明

相比所提供的示例，更改了用于初始化水平集的符号距离函数，保证初始水平集能够覆盖待分割的目标。更改了演化函数 `evolution_cv()`，设置了针对新演化函数的参数。

（2）evolution_cv.m

①源代码

```
function phi = evolution_cv(I, phi0, g, gx, gy, mu, nu, lambda, delta_t, epsilon, numIter)
```

```

l = BoundMirrorExpand(l); % 镜像边缘延拓
phi = BoundMirrorExpand(phi0);

%计算 div(g*Nabla(phi))需要下列项
%div(uV)=Vxgrad(u) + u.*div(V)
[phix,phiy]=gradient(phi);
temp=sqrt(phix.^2 + phiy.^2 + 1e-10);%%缺少 1e-10 项会导致曲线消失
phix=phix./temp;
phiy=phiy./temp;

for k = 1 : numIter
    phi = BoundMirrorEnsure(phi);
    delta_h = Delta(phi,epsilon);
    Curv = curvature(phi);

    % updating the phi function

    distRictTerm = mu * (4 * del2(phi) - Curv);
    lengthTerm = lambda * delta_h .* ( phix.*gx + phiy.*gy + g.*Curv);
    areaTerm = nu * g .* delta_h;
    new_term = distRictTerm + lengthTerm + areaTerm;
    phi = phi + delta_t * new_term;
end
phi = BoundMirrorShrink(phi); % 去掉延拓的边缘

```

②函数说明

新的演化函数，根据所给定的公式，迭代演化水平集函数。

（3）edge_detector.m

①源代码

```

function g=edge_detector(l,sigma)
%a general edge-detector g=1/(1+laplas(G_delta(x,y))*l)
l = BoundMirrorExpand(l); % 镜像边缘延拓
G=fspecial('gaussian',15,sigma);
X=conv2(l,G,'same');
[lx,ly]=gradient(X);
g=1./(1+lx.^2+ly.^2);

```

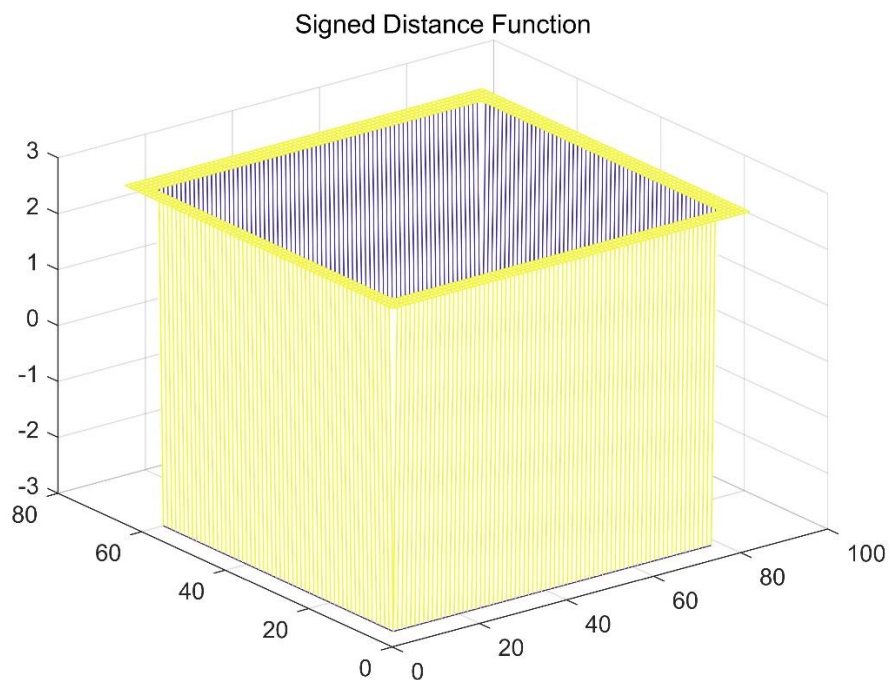
②函数说明

边缘检测子，即公式中的函数 g 。

三、实验结果及分析

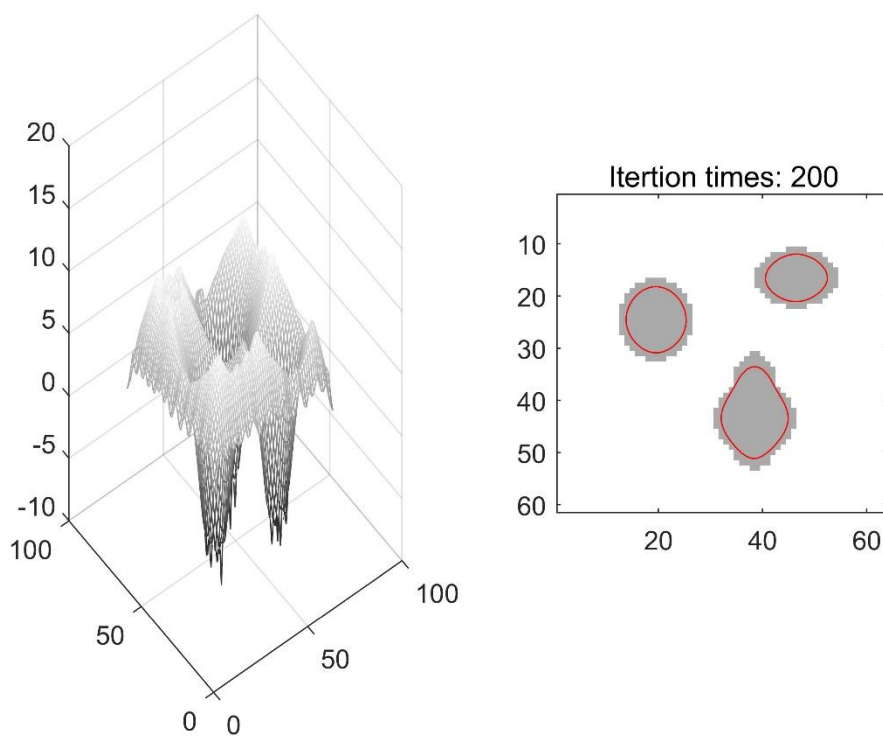
1、实验结果

如图为所使用的符号距离函数，使用该符号距离函数初始化水平集是为了使得初始零水平集能够完全覆盖待检测对象。



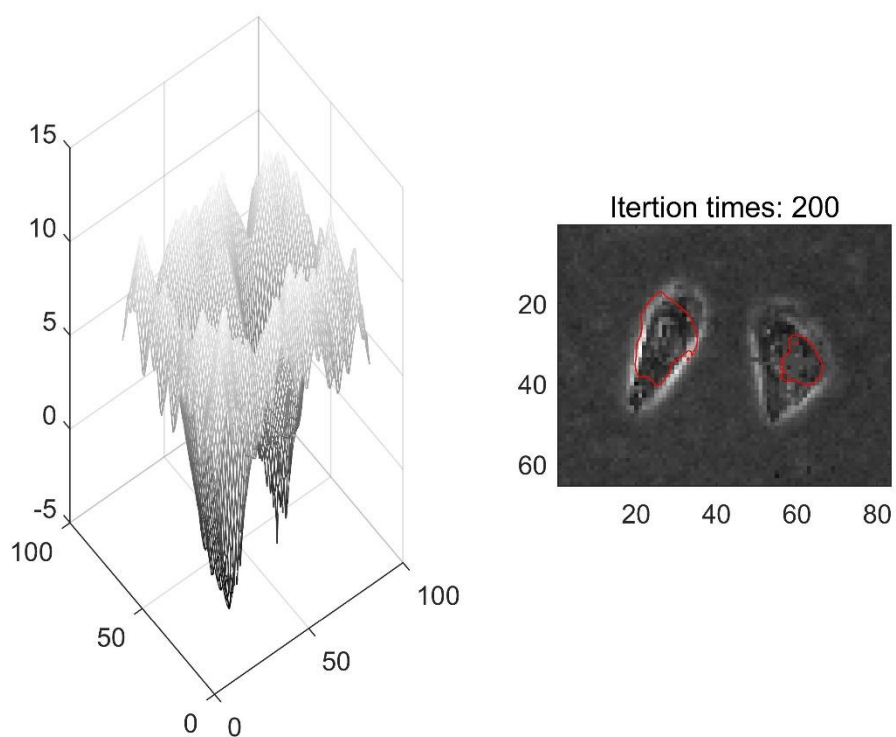
符号距离函数

如图所示为 **three.bmp** 在迭代 200 次以后的结果，此时结果已经收敛。可以看到，相比示例所提供的演化方程，新的演化方程对于边缘的光滑度要求更高，因此检测结果没有将锯齿状边缘包含在内。

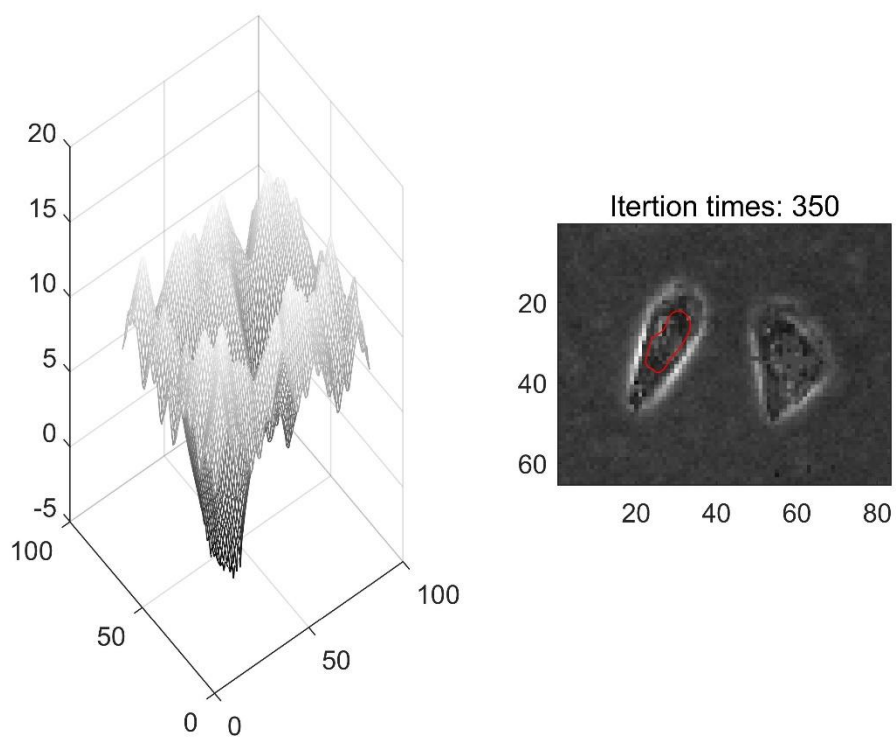


three.bmp 分割结果

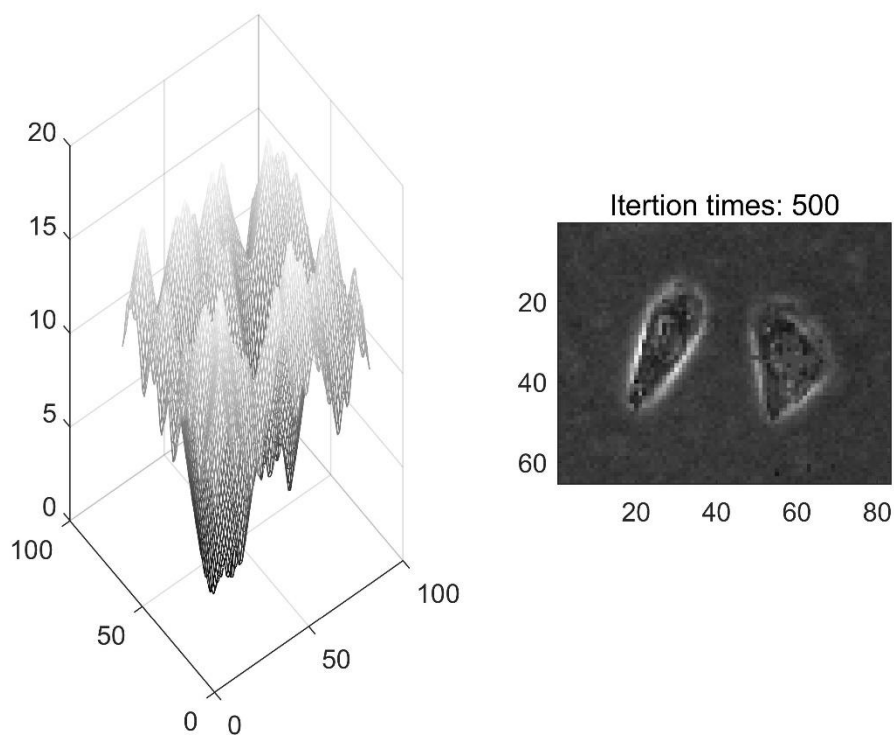
针对 twocells.bmp，迭代 200 次结果如下：



迭代 350 次结果如下：

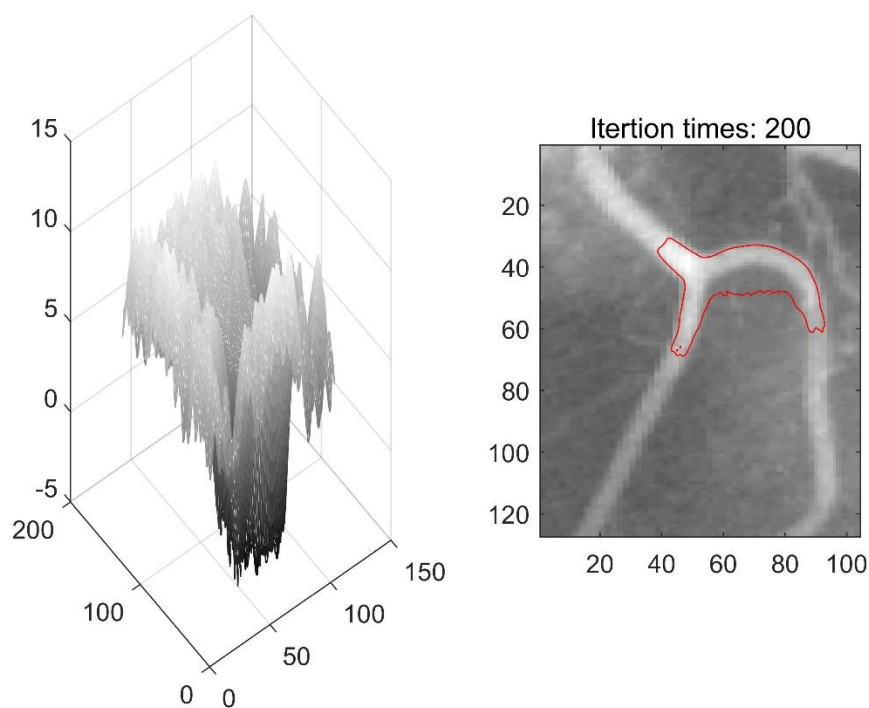


迭代 500 次结果如下：

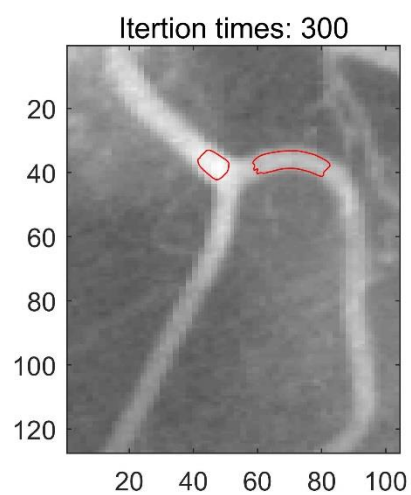
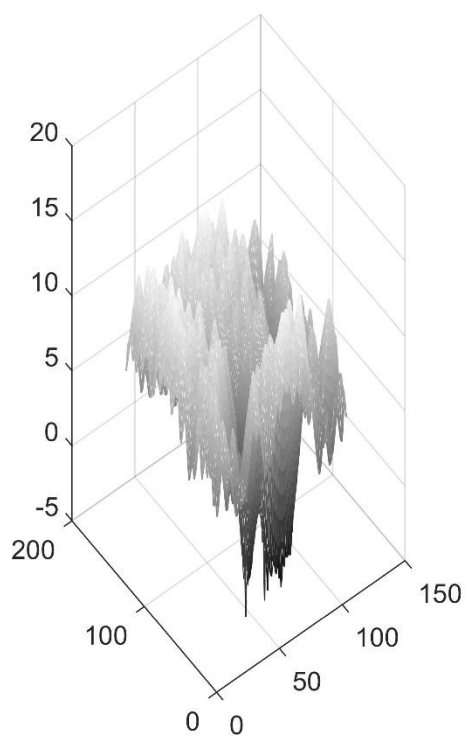


可以看出，在迭代了 500 次以后，零水平集已经消失，无法分割目标。这是由于在这张图片中，前景和背景的区别不够明显，该算法难以区分。

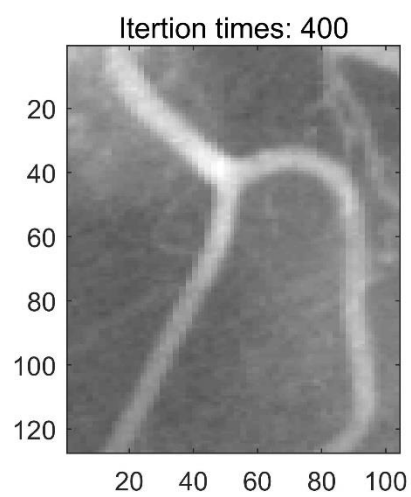
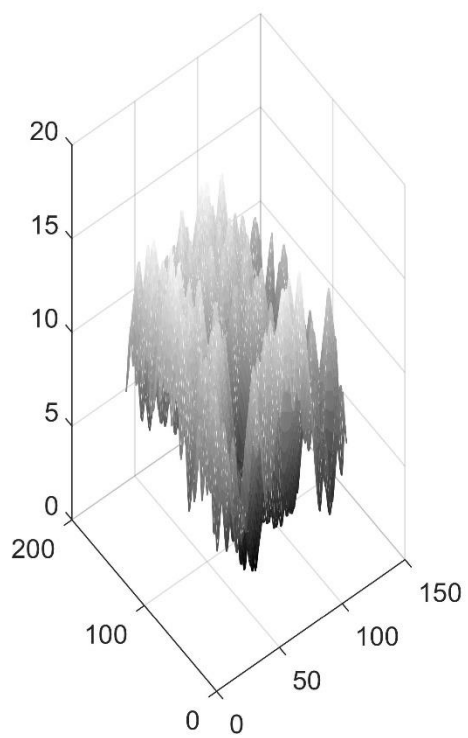
针对 `vessel.bmp`，在迭代 200 次以后，结果如下：



迭代 300 次后:



迭代 400 次后:



同 `twocells.bmp` 相同，在迭代若干次后零水平集同样消失，无法分割目标。但是这张图和 `twocells.bmp` 略有不同，这张图前景背景区分还算清晰，但是由于前景在整张图中不封闭，没有闭合的边缘，因此初始水平集无法完全包含该目标，因此无法检测。

2、简单分析

新演化方程较原方程对于平滑性要求高，对于锯齿状的边缘会忽略。针对不同的图像，参数设置的不同会导致不同的分割结果，甚至对于一些图像无法分割。有两方面的原因导致无法分割，一方面，初始水平集函数要覆盖被分割前景，如果不具有闭合边缘则不能分割；另一方面，前景和背景的区别要足够明显，否则在演化过程中会忽略不够清晰的边缘。

四、实验收获

- 1、了解了水平集方法进行图像分割的步骤。
- 2、加深了对于演化方程、水平集等概念的理解。