

# 第一次DIA作业：水平集分割

---

实验名称：

水平集图像分割

实验目标：

使用matlab软件实现以下水平集演化方程以达到分割图像的效果。

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

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

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

实验步骤：

1. 获取图像
2. 初始化零水平集和参数
3. 根据演化方程不断更新水平集函数
4. 直到零水平集不再变化，终止迭代

实验内容：

## 1.test\_demo.m

test\_demo为主要运行文件，控制水平集不停更新迭代直到达到收敛后停止。

其具体代码如下：

```
%清空缓存
```

```
clear;
```

```
close all;
```

```
clc;
```

```
%读取图像数据：有四张图可随意更换图像查看结果
```

```
%Img=imread('test_images/three.bmp');
```

```
%Img=imread('test_images/circle.jpg');
```

```
% Img=imread('test_images/vessel.bmp');
```

```
Img=imread('test_images/twoCells.bmp');
```

```
if size(Img, 3) > 1
```

```
%判断图像数据是否为rgb图像，若
```

```
是则转为灰度图像
```

```
    U = rgb2gray(Img);
```

```
else
```

```
    U = Img;
```

```
end
```

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

```
%初始化零水平集函数并绘制其三维空间曲面
```

```
ic=nrow/2;
```

```
jc=ncol/2;
```

```
r=27; %初始水平集函数的半径
```

```
phi_0 = sdf2circle(nrow,ncol,ic,jc,r);
```

```
% construct the signed distance function
```

```
figure; mesh(phi_0); title('Signed Distance Function')
```

```

%设置超参数
delta_t = 5;           %更新步长
lambda_1=3;
nu=5;
h = 1;
epsilon = 0.4;
mu = 0.04; %0.01*255*255;
numIter = 3;           %迭代次数

```

```

g=GaosiGrad2(I,0.5);    %对原图作高斯平滑后求初始梯度的倒数g

```

```

for k=1:250,
    phi = evolution_cv(I, phi, mu, nu, lambda_1, delta_t, epsilon,
numIter,g); % evolution_cv是演化方程
    if mod(k,2)==0
        pause(.5);
        figure(2); clc; axis equal;
        title(sprintf('Iteration times: %d', k));
        subplot(1,2,1); mesh(phi); %绘制零水平集的3D图
        subplot(1,2,2);
        imagesc(uint8(I)); %绘制被分割的图像
        colormap(gray)
        hold on; plotLevelSet(phi,0,'r');

        if k == 2
            seg_region_old = (phi < 0);
        else
            seg_region_new = (phi < 0);
            dif_pixNum = sum(sum(abs(seg_region_old -
seg_region_new)));
            if dif_pixNum < 1 % 零水平集包围的区域不再变化，则终止迭代
                fprintf('Level set evolution is converged.\n');
                break;
            else
                seg_region_old = seg_region_new;
            end
        end
    end
end
end

```

```
end;
```

## 2. evolution\_cv.m

水平集演化方程文件：实现图1中的演化方程

```
function phi = evolution_cv(I, phi0, mu, nu, lambda_1, delta_t,
epsilon, numIter,g);
```

其中，I是输入图像，phi0是初始零水平集函数，mu,nu,lambda\_1,delta\_t,epsilon,numIter都是设定好的超参数，g是由高斯平滑梯度函数计算出来的值。

```
I = BoundMirrorExpand(I); % 镜像边缘延拓
phi = BoundMirrorExpand(phi0);
g = BoundMirrorExpand(g);

for k = 1 : numIter
    phi = BoundMirrorEnsure(phi); %镜像边缘确保
    g = BoundMirrorEnsure(g);

    delta_h = Delta(phi,epsilon); %计算冲激函数
    Curv = curvature(phi); %曲率公式
    delta_p = del2(phi); %拉普拉斯算子：这里不可直接用
    gradient算，差距会很大

    %计算初始梯度项和零水平集函数的梯度
    [gx,gy]=gradient(g);
    [phix,phiy]=gradient(phi);
    norm=sqrt(phix.^2 + phiy.^2 + 1e-10);
    phixn=phix./norm;phiyn=phiy./norm;

    % 根据演化方程更新水平集函数，更新步长为delta_t
    phi=phi+delta_t*(mu*(4*delta_p-Curv)+lambda_1*delta_h.*
(gx.*phixn+gy.*phiyn+g.*Curv)+nu*g.*delta_h);
end
phi = BoundMirrorShrink(phi); % 去掉延拓的边缘
```

### 3.Delta.m

实现图一中的冲激函数项。

```
fphi=(0.5/epsilon)*(1+cos(pi*phi/epsilon));  
Delta_h= fphi.*(phi<=epsilon)&(phi>=-epsilon);
```

### 4.GaosiGrad2.m

```
function g = GaosiGrad2(I,sigma)  
  
%生成二维高斯卷积核, sigma=0.5  
G=fspecial('gaussian',15,sigma);  
II=conv2(I,G,'same');    %实现高斯卷积操作  
[Ix,Iy]=gradient(II);    %求平滑后的图像梯度  
f = Ix.^2 + Iy.^2;       %求梯度模  
g = 1./(1+f);            %计算倒数  
  
end
```

实验结果：





