

```

function [L,N, C] = MyASLIC(I, I_gray, K, M,display)

[m,n] = size(I_gray);

% 计算 S
S = round(sqrt(m*n/K));

% 初始化 Label 和 Distance
Label = -1 * ones(m,n);

Distance = Inf * ones(m,n);

[H,w] = meshgrid(round(0.5 * S):S:m,round(0.5 * S):S:n);

H = reshape(H, 1, []);

w = reshape(w, 1, []);

N = size(H);

N = N(2);

% 初始化 Center
C = zeros(N,5);

for i = 1:N

    % 找 3*3 邻域梯度最小的设为初始中心

    if H(i)+1 > m

        img = I(H(i)-2:H(i),W(i)-1:W(i)+1,1);

        [Fx, Fy] = gradient(img);

        F = sqrt(Fx.^2 + Fy.^2);

        [h,w] = ind2sub([3,3],find(F == min(min(F))));

        C(i,1) = I(H(i)-3+h(1),W(i)-2+w(1),1);

        C(i,2) = I(H(i)-3+h(1),W(i)-2+w(1),2);

        C(i,3) = I(H(i)-3+h(1),W(i)-2+w(1),3);

        C(i,4) = H(i)-2+h(1);

```

```
C(i,5) = W(i)-2+w(1);
```

```
else
```

```
if W(i)+1 > n
```

```
img = I(H(i)-1:H(i)+1,W(i)-2:W(i),1);
```

```
[Fx, Fy] = gradient(img);
```

```
F = sqrt(Fx.^2 + Fy.^2);
```

```
[h,w] = ind2sub([3,3],find(F == min(min(F))));
```

```
C(i,1) = I(H(i)-2+h(1),W(i)-3+w(1),1);
```

```
C(i,2) = I(H(i)-2+h(1),W(i)-3+w(1),2);
```

```
C(i,3) = I(H(i)-2+h(1),W(i)-3+w(1),3);
```

```
C(i,4) = H(i)-2+h(1);
```

```
C(i,5) = W(i)-2+w(1);
```

```
else
```

```
img = I(H(i)-1:H(i)+1,W(i)-1:W(i)+1,1);
```

```
[Fx, Fy] = gradient(img);
```

```
F = sqrt(Fx.^2 + Fy.^2);
```

```
[h,w] = ind2sub([3,3],find(F == min(min(F))));
```

```
C(i,1) = I(H(i)-2+h(1),W(i)-2+w(1),1);
```

```
C(i,2) = I(H(i)-2+h(1),W(i)-2+w(1),2);
```

```
C(i,3) = I(H(i)-2+h(1),W(i)-2+w(1),3);
```

```
C(i,4) = H(i)-2+h(1);
```

```
C(i,5) = W(i)-2+w(1);
```

```
end
```

```
end
```

```
end
```

```
% 根据论文上的说法，循环最多 10 次即可收敛
```

```
for i=1:10
```

```
    mc = 0;
```

```
    ms= 0;
```

```
    mcs = zeros(N,2);
```

```
    for j= 1:N
```

```
        range = [max(C(j,4)-2*S,1), min(C(j,4)+2*S,m), max(C(j,5)-2*S,1), min(C(j,5)+2*S,n)];
```

```
        for h = range(1):range(2)
```

```
            for w = range(3):range(4)
```

```
                dc = sqrt((I(h,w,1)-C(j,1))^2 + (I(h,w,2)-C(j,2))^2 + (I(h,w,3)-C(j,3))^2);
```

```
                ds = sqrt((h-C(j,4))^2 + (w-C(j,5))^2);
```

```
                if i == 1
```

```
                    d = sqrt(dc^2 + ((ds/S)^2)*(M^2));
```

```
                else
```

```
                    d = sqrt((dc/mcs(j,1))^2 + ((ds/mcs(j,2))^2));
```

```
                end
```

```
                if d < Distance(h,w)
```

```
                    if dc > mc
```

```
                        mc = dc;
```

```
                    end
```

```
                    if ds > ms
```

```
                        ms = ds;
```

```
                    end
```

```
                    Distance(h,w) = d;
```

```
                    Label(h,w) = j;
```

```
end
```

```
end
```

```
end
```

```
mcs(j,:) = [mc,ms];
```

```
ms = 0;
```

```
mc = 0;
```

```
end
```

```
% 更新中心
```

```
for j = 1:N
```

```
index = find(Label == j);
```

```
[h,w] = ind2sub([m,n], index);
```

```
Sizeh = size(h);
```

```
sum = zeros(1,5);
```

```
for k = 1:Sizeh(1)
```

```
sum = sum + [I(h(k),w(k),1), I(h(k),w(k),2), I(h(k),w(k),3), h(k), w(k)];
```

```
end
```

```
sum = sum / Sizeh(1);
```

```
sum(4) = round(sum(4));
```

```
sum(5) = round(sum(5));
```

```
C(j,:) = sum;
```

```
end
```

```
% 显示每一次的过程
```

```
if display == 1
```

```
BW = boundarymask(Label);
```

```
imshow(imoverlay(lab2rgb(I),BW,'cyan'),'InitialMagnification',67);
```

```
end
```

```
end
```

```
L = Label;
```

Published with MATLAB® R2018b