Table of Contents

```
function [label]=SLIC_Proc(I,ks,m,addGaborFeat,visualization)
```

SLIC Processing

input: I-image,ks-number of superpixel, m-param for distance calc output: superpixel label

default param setting

```
if nargin<4
    addGaborFeat = false;
end
if nargin < 5
   visualization = false;
end
```

initialization

```
gabormag=0;
if addGaborFeat
    gray = rgb2gray(im2single(I));
    wavelength = 2.^(0:3) * 2;
    orientation = 0:45:135;
    g = gabor(wavelength,orientation);

    gabormag = imgaborfilt(gray,g);
    for i = 1:length(g)
        sigma = 0.5*g(i).Wavelength;
        gabormag(:,:,i) = imgaussfilt(gabormag(:,:,i),3*sigma);
    end
end

I = double(I);
image_height = size(I,1);
image_width = size(I,2);
pixel_num = image_height*image_width;
super_pixel_unit = floor(sqrt(pixel_num/ks));
```

```
distance = ones([image_height,image_width])*inf;% distance assume to
 be infinity
label = zeros([image_height,image_width]);


clusters = zeros([image_height,image_width]);
cls_num = 1;
for h = ceil(super_pixel_unit/2) :super_pixel_unit:image_height
    for w = ceil(super_pixel_unit/2):super_pixel_unit:image_width
        clusters(h,w) = cls_num;
        cls_num = cls_num + 1;
    end
end
```

```
Not enough input arguments.

Error in SLIC_Proc (line 28)
I = double(I);
```

refine the cluster center

```
clusinuse = refine_cluster(clusters,I);
%     figure(1),imshow(clusinuse,[]);
handle = waitbar(0,'Waiting For SLIC...');
```

main assignment

```
for i = 1:10
    [clusinuse,distance,label] =
 assignment(clusinuse,I,super_pixel_unit,m,distance,label,addGaborFeat,gabormag);
    if visualization
        figure(1),imshow(clusinuse,[]);
        figure(2),imshow(label,[]);
    else
        waitbar(i/10,handle);
    end
end
delete(handle);

end
```

get the gradient in 3x3 region

```
function gradient=get_gradient(I,local_h,local_w)
if local_w+1 > size(I,2)
    local_w = size(I,2)-1;
end
if local_h+1 > size(I,1)
    local_h = size(I,1)-1;
end
```

```
gradient=I(local_h+1,local_w+1,1)-I(local_h,local_w,1)+...
    I(local_h+1,local_w+1,2)-I(local_h,local_w,2)+...
    I(local_h+1,local_w+1,3)-I(local_h,local_w,3);

end
```

refined the init cluster center based on gradient

```
function refined_cluster=refine_cluster(clusters,I)
refined_cluster = clusters;
idx = find(clusters ~= 0);
[h,w]=ind2sub(size(clusters),idx);
for i = 1:size(idx)
    current_gradient = get_gradient(I,h(i),w(i));
    for dh = -1:1
        for dw = -1:1
            tmph = h(i)+dh;
            tmpw = w(i)+dw;
            new_gradient = get_gradient(I,tmph,tmpw);
            if new_gradient < current_gradient
                tmp = refined_cluster(h(i),w(i));
                refined_cluster(h(i),w(i)) = 0;
                refined_cluster(tmph,tmpw) = tmp;
                current_gradient = new_gradient;
            end
        end
    end
end
end
```

main assignment: calc distance and do clustering

```
function [newcls,newdistance,newlabel] =
 assignment(clusters,I,super_pixel_unit,m,distance,label,addgabor,gabormag)
newcls = clusters;
newdistance = distance;
newlabel=label;

idx = find(clusters ~= 0);
[h,w]=ind2sub(size(clusters),idx);
for i = 1:size(idx)
    for dh = -super_pixel_unit:super_pixel_unit
        spoth = h(i)+dh;
        if spoth < 1 | spoth > size(I,1)
            continue;
        end
        for dw = -super_pixel_unit:super_pixel_unit
            spotw = w(i)+dw;
            if spotw < 1 | spotw > size(I,2)
                continue;
            end
            l = I(spoth,spotw,1);
            a = I(spoth,spotw,2);
            b = I(spoth,spotw,3);

            dc = sqrt((l-I(h(i),w(i),1))^2+(a-I(h(i),w(i),2))^2+(b-
I(h(i),w(i),3))^2);
            ds = sqrt((dh)^2+(dw)^2);
            D = sqrt((dc)^2+(ds/super_pixel_unit)^2*m^2);
            gaborD = 0;

            if addgabor
                for q=1:size(gabormag,3)
                    cur = gabormag(spoth,spotw,q);
                    past = gabormag(h(i),w(i),q);
                    gaborD = gaborD+(cur-past)^2;
                end
                gaborD = sqrt(gaborD);
            end

            D = sqrt(D^2+gaborD^2);

            if D < newdistance(spoth,spotw)
                newdistance(spoth,spotw) = D;
                newlabel(spoth,spotw) = clusters(h(i),w(i));
            end
        end
    end
end

% update clusters
for i = 1:size(idx)
    minions = find(newlabel==clusters(idx(i)));
    [cls_h,cls_w] = ind2sub(size(newlabel),minions);
    sum_h = sum(cls_h(:));
    sum_w = sum(cls_w(:));
    number = size(minions,1);
    %     for p = 1:size(minions)
    %         sum_h = sum_h + cls_h(p);
    %         sum_w = sum_w + cls_w(p);
    %         number = number + 1;
    %     end
    newcls_h = ceil(sum_h/number);
    newcls_w = ceil(sum_w/number);

    newcls(idx(i)) = 0;
    newcls(newcls_h,newcls_w) = clusters(idx(i));


end

end
```

Published with MATLAB® R2019a