

```

function out=Filter_Oil(img)

img=uint16(img);

radius=5;

intensity_level=100;


%将输入的图片数组转为 uint16 型，避免计算时超出范围

image_size=size(img);

height=image_size(1); %图片的高度

width=image_size(2); %图片的宽度

out=zeros(height,width,3);


%用一个同样大小的数组记录输出图像的数据，初始化为零

for x=1:height

    for y=1:width

        intensity_counter=zeros(intensity_level,1);

        %强度计数器，记录邻域每个强度出现次数

        sum_r=uint16(zeros(intensity_level,1));

        %r 分量累加器，记录每个强度 r 分量的和

        sum_g=uint16(zeros(intensity_level,1)); %g 分量累加器

        sum_b=uint16(zeros(intensity_level,1)); %b 分量累加器

        for i=(x-radius):(x+radius)

            for j=(y-radius):(y+radius) %邻域的遍历

                if i>0&& i<=height&& j>0&& j<=width

                    intensity=(img(i,j,1)+img(i,j,2)+img(i,j,3))/3*intensity_level/255;

                    %计算每一像素的强度

                    if intensity==0

                        intensity=1;

                    end %强度不能为零，避免数组下标出现零

                    intensity_counter(intensity)=intensity_counter(intensity)+1;

```

```

        %统计强度出现的次数

        sum_r(intensity)=sum_r(intensity)+img(i,j,1);

        %同一强度的 r 分量求和

        sum_g(intensity)=sum_g(intensity)+img(i,j,2);

        sum_b(intensity)=sum_b(intensity)+img(i,j,3);

    end

end

end

intensity_counter_max=max(intensity_counter);

%找出同一强度出现最多的次数

for i=1:intensity_level

    if intensity_counter(i)==intensity_counter_max

        index=i;

    end

end

%若出现最多的次数有重复，则取强度大的像素作为输出

out(x,y,1)=sum_r(index)/intensity_counter(index);

%出现最多次的强度值对应像素的 r 分量取平均值，作为当前像素输出的 r 分量

out(x,y,2)=sum_g(index)/intensity_counter(index);

%输出的 g 分量

out(x,y,3)=sum_b(index)/intensity_counter(index);

%输出的 b 分量

end

end

out=uint8(out);%将 uint16 图像转化为 matlab 默认的 uint8 型

end

```

Published with MATLAB® R2017b