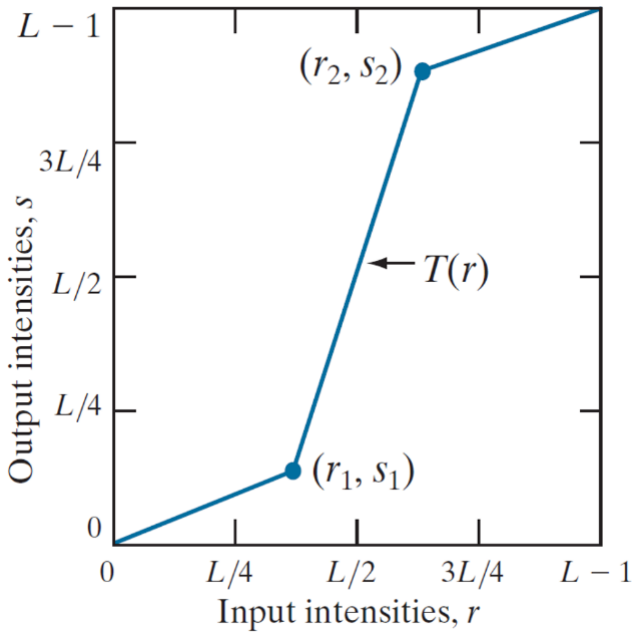
可视化作业week1

18300290007加兴华

1

Implement a piecewise linear transformation function (below figure) for image contrast stretching. The code should read in an image; for intensity of all pixels, use the function to compute new intensity values; and finally output/ save the image with new intensity values.

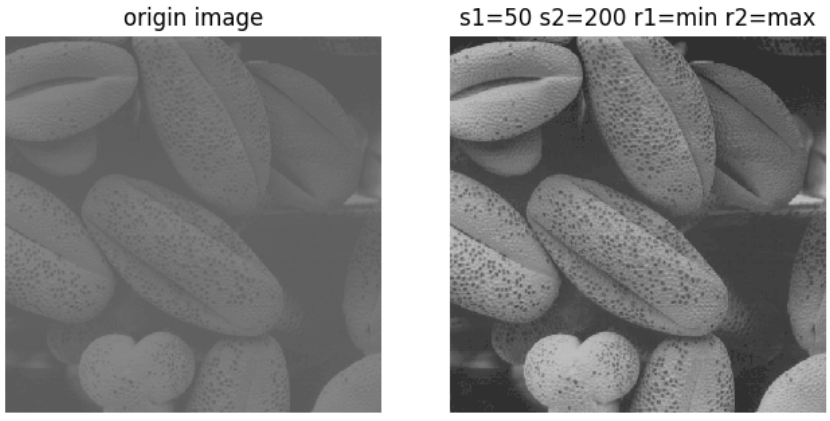


代码：（python）

```python
import matplotlib.pyplot as plt
from skimage import io, exposure,color
path='origin.tif'
img = io.imread(path)
rows,cols=img.shape[:2]

plt.subplot(121)
plt.title('origin image')
io.imshow(img)
plt.axis('off')
```

```
11
12  r2=img.max()-1
13  r1=img.min()+1 #防分母=0
14  s1=50
15  s2=200
16  i2=img
17
18  for i in range(rows):
19      for j in range(cols):
20          if (img[i,j]<=r1):
21               i2[i,j]= img[i,j]*s1/r1
22          elif  (img[i,j]>=r2):
23               i2[i,j]= (img[i,j]-r2)*(255-s2)/(255-r2)+s2
24          else :
25              i2[i,j]= (img[i,j]-r1)*(s2-s1)/(r2-r1)+s1
26
27  plt.subplot(122)
28  plt.title('s1=50 s2=200 r1=min r2=max')
29  io.imshow(i2)
30  plt.axis('off')
31
32  plt.show()
```

output：



分析：

分段线性处理灰度后，使得原图的对比度更加的明显，物体轮廓与表面细节更加清晰了，但是这种处理也有局限性，我曾选用其他图片作为素材，发现它只能处理低对比度造成的模糊，对于其他成因的模糊处理效果不佳。

2

(1) implement n-dimensional joint histogram and test the code on two-dimensional data; plot the results.

(2) implement computation of local histograms of an image using the efficient update of local histogram method introduced in local histogram processing.

Note that because only one row or column of the neighborhood changes in a one-pixel translation of the neighborhood, updating the histogram obtained in the previous location with the new data introduced at each motion step is possible and efficient in computation.

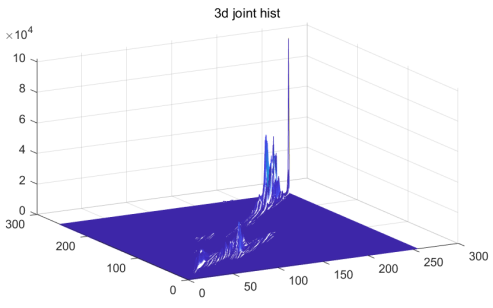
（1）

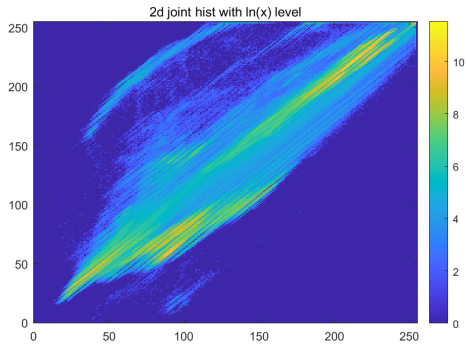
代码：（matlab）

```
clear all;clc;close all;
img=imread("smy.jpg");
subplot(2,2,1);
imshow(img);
title('origin image')

[m,n,~]=size(img);
count=zeros(256,256);
for i=1:m
    for j=1:n

 count(img(i,j,1)+1,img(i,j,2)+1)=1+count(img(i,j,1)+1,img(i,j,2)+1);      % 忽略第3通道
    end
end
count1=zeros(256,256);
for i=1:256
    for j=1:256
        count1(i,j)=reallog(count(i,j));
```

```
    end
end

subplot(1,2,2);
x=0:255;
y=0:255;
imagesc(x,y,count1);
axis xy;
colorbar();
title('2d joint hist with ln(x) level');
set(gca,'Position',[0.6 0.38 0.3 0.4])

subplot(2,2,3);
mesh(x,y,count)
title('3d joint hist');
```

output：





（2）

代码：（matlab）没看懂先做3再做这里，故直接截取自定义函数的片段，因不完全所以没法用素材验证（连续打印直方图的话画不完），但能实现第三题足够说明其有效性了

```
function localhist(B,k) %k为滑窗边长，B为图矩阵
[m,n]=size(B);
q=(k-1)/2; %中心到边界距离
```

```
c=zeros(256,1); %灰度统计列表
b=zeros(m+2*q,n+2*q); %扩张矩阵
% 下面五部将扩张边填充成原矩阵镜像
b(1+q:m+q,1+q:n+q)=B;
b(1:q,1+q:n+q)=flipud(B(2:q+1,1:n));
b(m+q+1:m+2*q,1+q:n+q)=flipud(B(m-q:m-1,1:n));
b(1+q:m+q,1:q)=fliplr(B(1:m,2:q+1));
b(1+q:m+q,1+n+q:n+2*q)=fliplr(B(1:m,n-q:n-1));
%初始化
for i=1:1+2*q
    for j=1:1+2*q
        c(b(i,j)+1)=c(b(i,j)+1)+1;
    end
end
flag=1;  %从左到右遍历或反之的标志
j=1+q;
for i=1+q:m+q
    sym=(flag-1)/2;  %配合flag使得下述代码可以统括两种遍历情况
    for j=(-1*sym*(2*q+1+n)+flag*(1+q)):flag:(-1*sym*
(2*q+1+n)+flag*(n+q))
        for x=i-q:i+q
            c(b(x,j-q)+1)= c(b(x,j-q)+1)-flag;
            c(b(x,j+q)+1)= c(b(x,j+q)+1)+flag;
        end
    end
    flag=-1*flag;
    %一行结束后下降
    if(i~=m+q)
        for x=j-q:j+q
            c(b(i-q,x)+1)= c(b(i-q,x)+1)-1;
            c(b(i+q,x)+1)= c(b(i+q,x)+1)+1;
        end
    end
end
A=uint8(A*255);
end
```

3

Implement the algorithm of local histogram equalization:

(1) first implement histogram equalization algorithm, and then

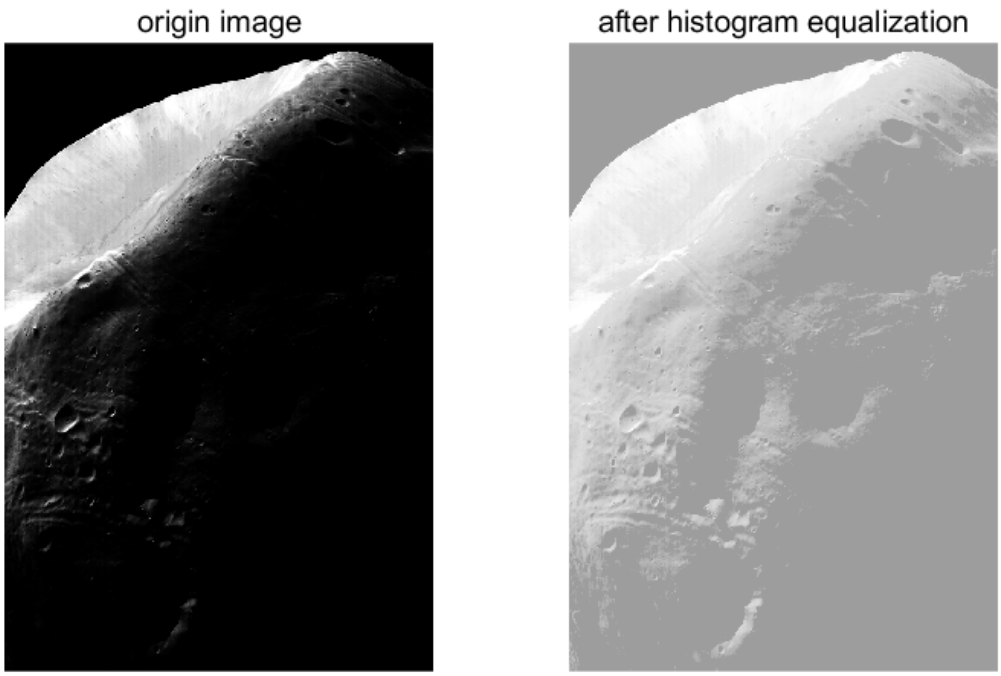
(2) implement the local histogram equalization using efficient computation of local histogram. Please test your code on images and show the results in your report.

(1)

代码：（matlab）

```
clear all;clc;close all;
img=imread("moon.tif");
subplot(1,2,1);
imshow(img);
title('origin image')

[m,n]=size(img);
count=zeros(256,1);
for i=1:m
    for j=1:n
        count(img(i,j)+1)=count(img(i,j)+1)+1;
    end
end
count=count./(m*n);   % 归一化
for i=2:256
    count(i)=count(i-1)+count(i);
end
count=uint8(count*255);
for i=1:m
    for j=1:n
        img(i,j)=count(img(i,j)+1);
    end
end
subplot(1,2,2);
imshow(img)
title('after histogram equalization')
```

output：



分析：

直方图均衡化使得图片中对比度低的部分（素材中为右下黑色区域）得到改善，但相对的图中对比度高的地方也受到了削弱，出现一定程度的失真。

(2)

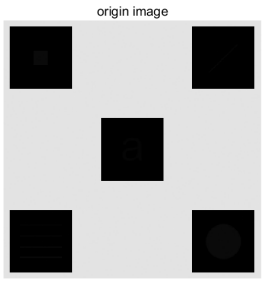
代码：（matlab）

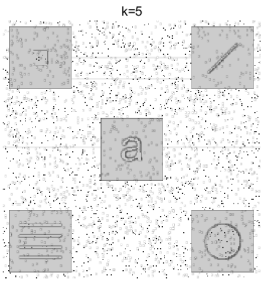
```
clear all;clc;close all;
img=imread("black.tif");
subplot(2,2,1);
imshow(img);
title('origin image')
subplot(2,2,2);
A=localhist(img,5);
imshow(A)
title('k=5');
subplot(2,2,3);
A=localhist(img,9);
imshow(A)
title('k=9');
```

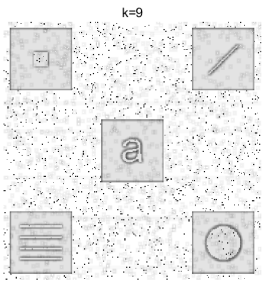
```
subplot(2,2,4);
A=localhist(img,13);
imshow(A)
title('k=13');

function A=localhist(B,k)
[m,n]=size(B);
q=(k-1)/2; %中心到边界距离
A=zeros(m,n); %目标矩阵
c=zeros(256,1); %灰度统计列表
b=zeros(m+2*q,n+2*q); %扩张矩阵
% 下面五部将扩张边填充成原矩阵镜像
b(1+q:m+q,1+q:n+q)=B;
b(1:q,1+q:n+q)=flipud(B(2:q+1,1:n));
b(m+q+1:m+2*q,1+q:n+q)=flipud(B(m-q:m-1,1:n));
b(1+q:m+q,1:q)=fliplr(B(1:m,2:q+1));
b(1+q:m+q,1+n+q:n+2*q)=fliplr(B(1:m,n-q:n-1));
%初始化
for i=1:1+2*q
    for j=1:1+2*q
        c(b(i,j)+1)=c(b(i,j)+1)+1;
    end
end
flag=1;  %从左到右遍历或反之的标志
j=1+q;
for i=1+q:m+q
    A(i-q,j-q)=sum(c(1:b(i,j)+1))/k^2;
    sym=(flag-1)/2;  %配合flag使得下述代码可以统括两种遍历情况
    for j=(-1*sym*(2*q+1+n)+flag*(1+q)):flag:(-1*sym*
(2*q+1+n)+flag*(n+q))
        for x=i-q:i+q
            c(b(x,j-q)+1)= c(b(x,j-q)+1)-flag;
            c(b(x,j+q)+1)= c(b(x,j+q)+1)+flag;
        end
        A(i-q,j-q)=sum(c(1:b(i,j)+1))/k^2;
    end
    flag=-1*flag;
    %一行结束后下降
    if(i~=m+q)
        for x=j-q:j+q
            c(b(i-q,x)+1)= c(b(i-q,x)+1)-1;
            c(b(i+q,x)+1)= c(b(i+q,x)+1)+1;
        end
```

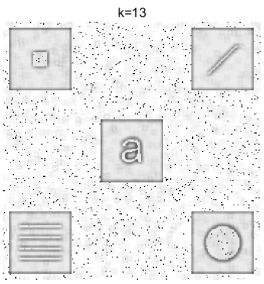
```
    end
end
A=uint8(A*255);
end
```

output：









分析：

随着k增大，局部均衡化在精度上就越低，但是如果是素材中轮廓宽度本身就很窄情况，k值极小反而不利于人眼辨识图案，更高的k值起到一种加粗的效果，此外小的k值还会带来更多噪音。