

龙子川

一、理论作业

均值滤波:

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

图像边缘像素不变, 对内部像素进行操作

设 $f'(x,y)$ 表示滤波后灰度值

$$f'(2,2) = H * \begin{bmatrix} 1 & 2 & 1 \\ 1 & 10 & 2 \\ 5 & 2 & 6 \end{bmatrix} = \frac{10}{3} \approx 3$$

同理可得 均值滤波后结果 B' =

$$B' = \begin{bmatrix} 1 & 2 & 1 & 4 & 3 \\ 1 & 3 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 8 \\ 5 & 5 & 5 & 7 & 8 \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

中值滤波:

对于 $f'(2,2)$, 对 3×3 大小的图像灰度值进行排序为

1 1 1 2 2 2 5 6 10

中值为 2, $f'(2,2) = 2$

同理可得 中值滤波结果为

$$B'' = \begin{bmatrix} 1 & 2 & 1 & 4 & 3 \\ 1 & 2 & 3 & 4 & 4 \\ 5 & 5 & 5 & 6 & 8 \\ 5 & 5 & 6 & 8 & 8 \\ 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

二、编程作业

1. 灰度直方图与离散傅里叶变换频谱幅度图

(1) 原理

灰度直方图

定义：灰度直方图是对图像中各个灰度级别像素数量的统计图。

实现原理：

- a) **遍历像素：**遍历灰度图像中的每个像素，根据像素的灰度值更新直方图数组。
- b) **可视化：**可以将直方图数据可视化为柱状图，横轴表示灰度值，纵轴表示对应的像素数量。

离散傅里叶变换（DFT）幅度图

定义：离散傅里叶变换是将信号从时域转换到频域的数学工具，幅度图表示信号在不同频率成分上的强度。

实现原理：

- a) **输入数据：**首先准备好要进行傅里叶变换的信号或图像数据。
- b) **应用 DFT：**

数学上，DFT 的公式为：

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

其中 $F(u, v)$ 是频域中的复数表示， $f(x, y)$ 是时域中的信号。

c) **计算幅度谱**: 由 DFT 得到的复数结果的幅度可以通过以下公式计算:

$$|F(u, v)| = \sqrt{\text{Re}(F(u, v))^2 + \text{Im}(F(u, v))^2}$$

Re 和 Im 分别表示复数的实部和虚部。

d) **中心化**: 使用傅里叶变换的中心化函数, 将低频成分移动到频谱中心。

(2)代码

```
% 灰度图转换
function image_gray()
I=imread('test_2.jpg');
% 将RGB图像转换为灰度图
I_1=rgb2gray(I);
imshow(I_1);
imwrite(I_1, 'gray_img.jpg')
end

% 输出灰度直方图
function histogram()
% 输入图像
I=imread('gray_img.jpg');
[m,n]=size('gray_img'); % 计算图像的长宽
p=zeros(1,256); %创建数组存储像素个数
% 统计每个像素出现的概率, 得到概率直方图
% 用length函数计算出相同像素的个数
for i=0:255
    p(i+1)=length(find(I==i));
end
bar([0:255],p);
end

% 计算二维离散傅里叶变换
% 函数实现
function imageDFT()
    I=imread('gray_image.jpg');
    I=im2double(I);
    [x,y] = size(I);
    Ax = ones(x,y);
    A = ones(x,y);
```

```

com = 0+1i;
% 对每一列进行DFT
for k =1:x
    for m=1:y
        sn =0;
        for n =1:x
            sn =sn + I(n,m)*exp(-com*2*pi*k*n/x);
        end
        Ax(k,m) = sn;
    end
end
% 对每一行进行DFT
for p =1:y
    for k =1:x
        sn =0;
        for m=1:y
            sn = sn+Ax(k,m)*exp(-com*2*pi*p*m/y);
        end
        A(k,p) = sn;
    end
end
F=my_fftshift(A);
F= abs(F);
F=log(F+1);
imshow(F, []);
end

function y = my_fftshift(x)

sz = size(x);
% 对于二维数组
if length(sz) == 2
    % 对行列都进行 fftshift
    y = fftshift2d(x, sz(1), sz(2));
% 对于一维数组
else
    N = length(x);
    if mod(N, 2) == 0
        % 将前半部分和后半部分进行置换
        y = [x(N/2 + 1:end); x(1:N/2)];
    else
        % N为奇数的情况
        y = [x(ceil(N/2) + 1:end); x(1:floor(N/2))];
        x(ceil(N/2));
    end
end

```

```

        end
    end
end

function y = fftshift2d(x, M, N)
    % 二维数组进行行和列的 fftshift

    y = x;
    % 行处理
    if mod(M, 2) == 0
        % 偶数行
        y = [y(M/2 + 1:end, :); y(1:M/2, :)];
    else
        % 奇数行
        y = [y(ceil(M/2) + 1:end, :); y(1:floor(M/2), :);
y(ceil(M/2), :)];
    end

    % 列处理
    if mod(N, 2) == 0
        % 偶数列
        y = [y(:, N/2 + 1:end), y(:, 1:N/2)];
    else
        % 奇数列
        y = [y(:, ceil(N/2) + 1:end), y(:, 1:floor(N/2)), y(:,
ceil(N/2))];
    end
end
end

```

(3) 结果分析



图 1 灰度图

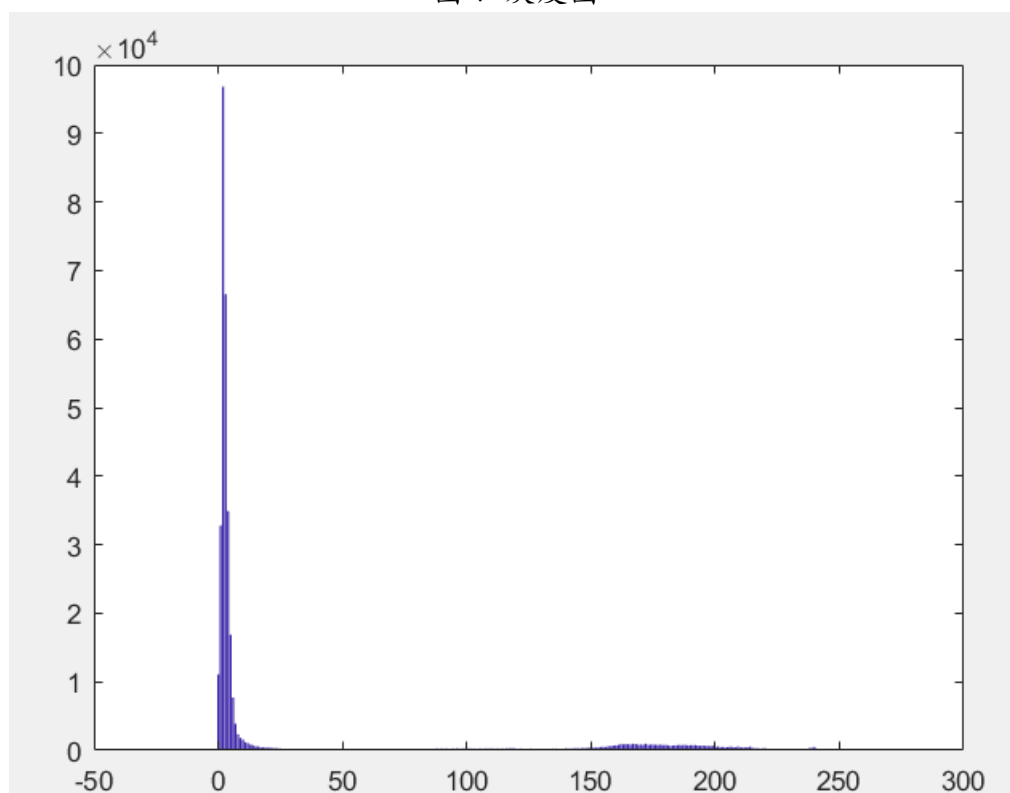


图 2 灰度直方图

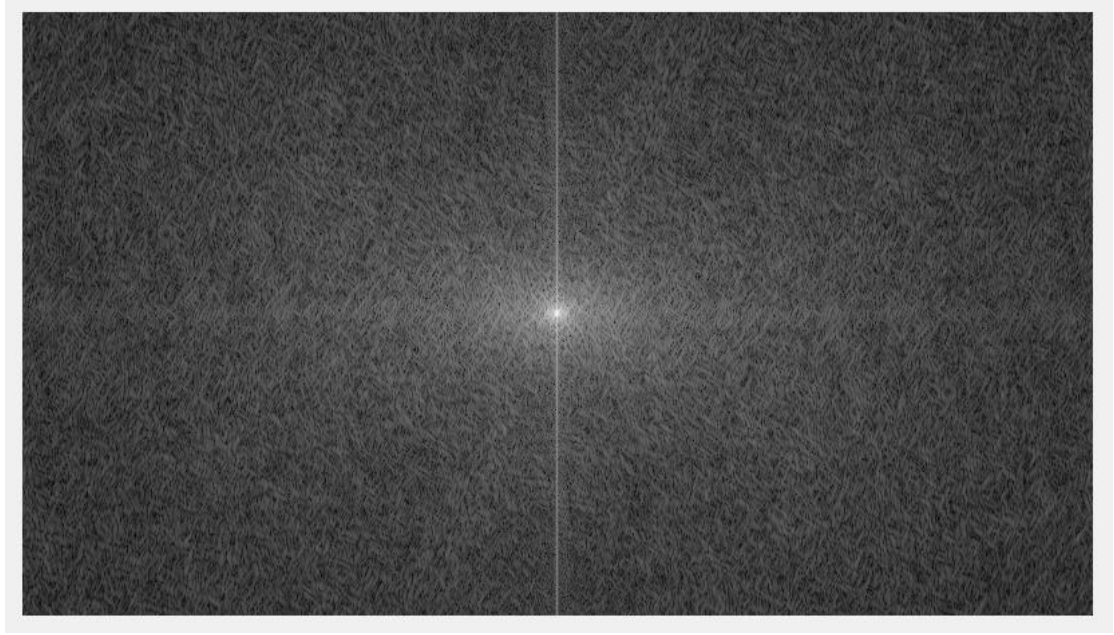


图 3 频谱图

对灰度图像的分析显示，该照片整体亮度偏暗。灰度直方图（图 2）表明灰度值主要集中在 25 以下。此外，频谱图（图 3）显示图像的频率大部分集中在低频区域，同时也存在高频成分，这可能与整体较暗的灰度有关。为改善图像的灰度分布，我们采用直方图均衡化和同态滤波两种方法，以实现更均匀的灰度分布。

2. 直方图均衡化和同态滤波

(1) 原理

直方图均衡化

直方图均衡化是一种增强图像对比度的技术，旨在改善图像的灰度分布，使其更加均匀。其基本原理如下：

- 计算直方图：**首先计算图像每个灰度级的出现频率，生成图像的灰度直方图。
- 计算累积分布函数（CDF）：**通过将直方图中的频率累加，得出累积分布函数。
- 映射：**利用 CDF，将原始图像的灰度值映射到新的灰度值上，使得新的灰度级分布更加均匀。这通常通过将 CDF 归一化并乘以最大灰度值来实现。
- 生成新图像：**通过映射后的灰度值生成均衡化后的图像。

同态滤波

同态滤波是一种处理图像光照变化和对比度的方法，其基本原理如下：

- a) **转换到频域**：首先通过傅里叶变换将图像从空间域转换到频域。在频域中，图像可以分解为低频（代表光照信息）和高频（代表细节信息）成分。
- b) **对数变换**：对频域图像进行对数变换，以将图像的亮度和反差分离。
- c) **调制**：在频域中，设计一个滤波器来增强高频成分，同时抑制低频成分。
- d) **反变换**：将处理后的频域图像通过逆傅里叶变换转换回空间域，得到增强后的图像。

(2)代码

```
%直方图均衡化
I=imread('gray_img.jpg');
hstepimg(I)
%函数实现
function his=hstepimg(I)
% 输入图像
[m,n]=size(I); % 计算图像尺寸
p=zeros(1,256); %创建数组存储像素概率
% 统计每个像素出现的概率，得到概率直方图
% 用length函数计算相同像素的个数
for i=0:255
    p(i+1)=length(find(I==i))/(m*n);
end
% 求累计概率
s=zeros(1,256);
for i=1:256
    for j=1:i
        s(i)=p(j)+s(i);
    end
end
%每个像素映射
a=round(s*255); %四舍五入
his=I;
for i=0:255
    his(I==i)=a(i+1);
end
%均衡化后的直方图
k=zeros(1,256);
```



```

for i=0:255
    k(i+1)=length(find(his==i));
end
%输出图像进行对比
figure;
subplot(121),imshow(I);title('原图')
subplot(122),imshow(his);title('均衡化')
figure;
bar([0:255],k);%直方图均衡化后的灰度直方图
end

```

```

%同态滤波
I=imread('gray_img.jpg');
homomorphic(I)
%函数实现
function imgtemp=homomorphic(image)
%参数声明
rH = 0.15;
rL = 0.1;
c = 0.15;%介于rH和rL之间
D0 = 0.2;
[M, N] = size(image);
%取对数
img_log = log(double(image) + 1);
%平移到中心
img_py = zeros(M, N);
for i = 1:M
    for j= 1:N
        if mod(i+j, 2) == 0
            img_py(i,j) = img_log(i, j);
        else
            img_py(i,j) = -1 * img_log(i, j);
        end
    end
end
end
% 对平移后的图像进行傅里叶变换
img_py_fft = fft2(img_py);
%同态滤波函数
%选用高斯滤波
H = zeros(M, N);
r = rH - rL;
D = D0^2; %设置参数
m_mid=floor(M/2);%中心点坐标
n_mid=floor(N/2);

```

```

for i = 1:M
    for j = 1:N
        dis = ((i-m_mid)^2+(j-n_mid)^2); %到中心点的距离
        H(i, j) = r * (1-exp((-c)*(dis/D))) + rL; %高斯同态滤波函数
    end
end
imgtemp = img_py_fft.*H; %滤波
%反变换
imgtemp = abs(real(ifft2(imgtemp)));
imgtemp = exp(imgtemp) - 1;
%归一化
max_num = max(imgtemp(:)); min_num = min(imgtemp(:));
range = max_num - min_num;
img_after = zeros(M,N, 'uint8');
for i = 1 : M
    for j = 1 : N
        img_after(i,j) = uint8(255 * (imgtemp(i, j)-min_num) /
            range);
    end
end
%同态滤波后的直方图
k=zeros(1,256);
for i=0:255
    k(i+1)=length(find(img_after==i));
end
figure;
subplot(1,2,1), imshow(image), title('原图像');
subplot(1,2,2), imshow(img_after), title('变换后');
figure;
bar([0:255],k); %同态滤波后的灰度值方图
end

```

(3) 结果分析

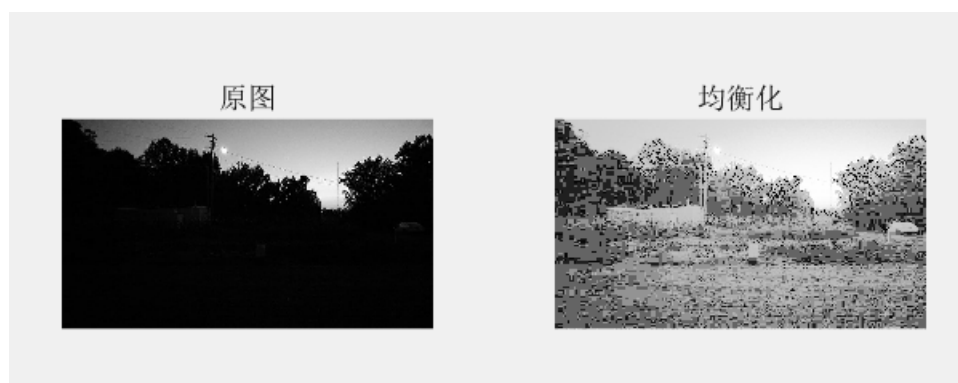


图 4 直方图均衡化前后对比图

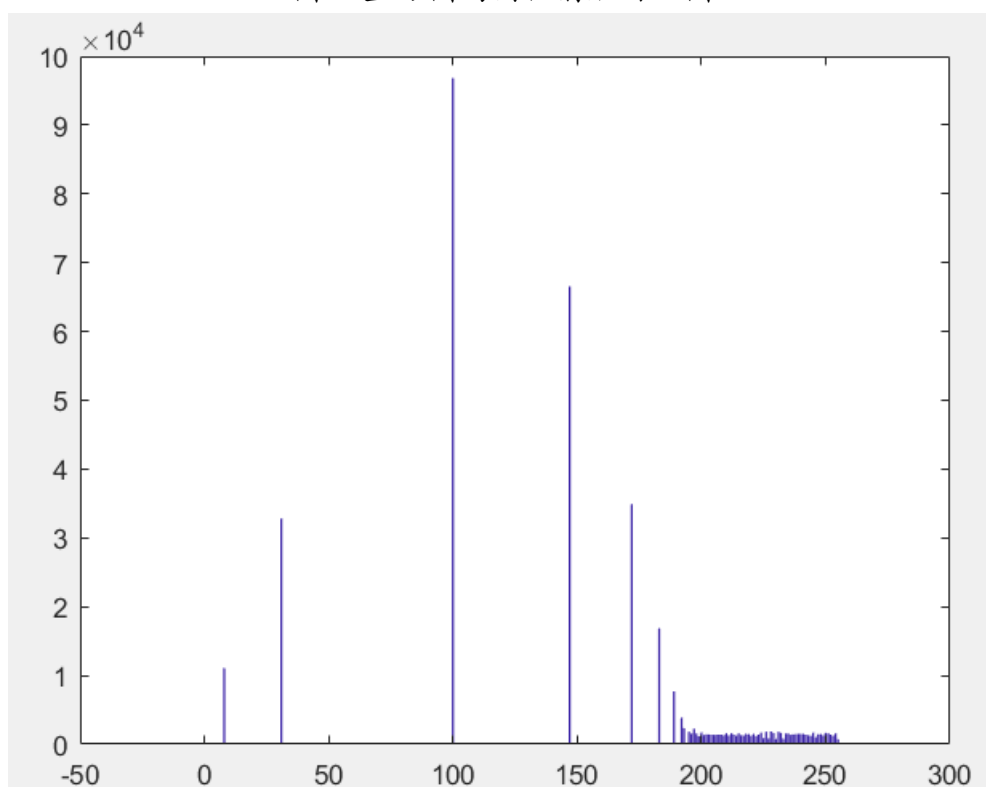


图 5 均衡化后的灰度直方图

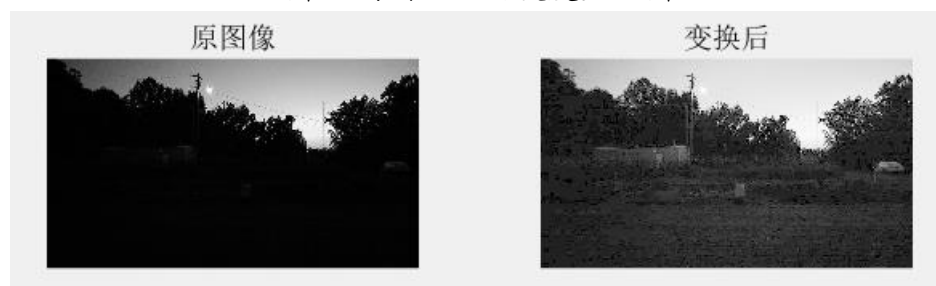


图 6 同态滤波前后对比图

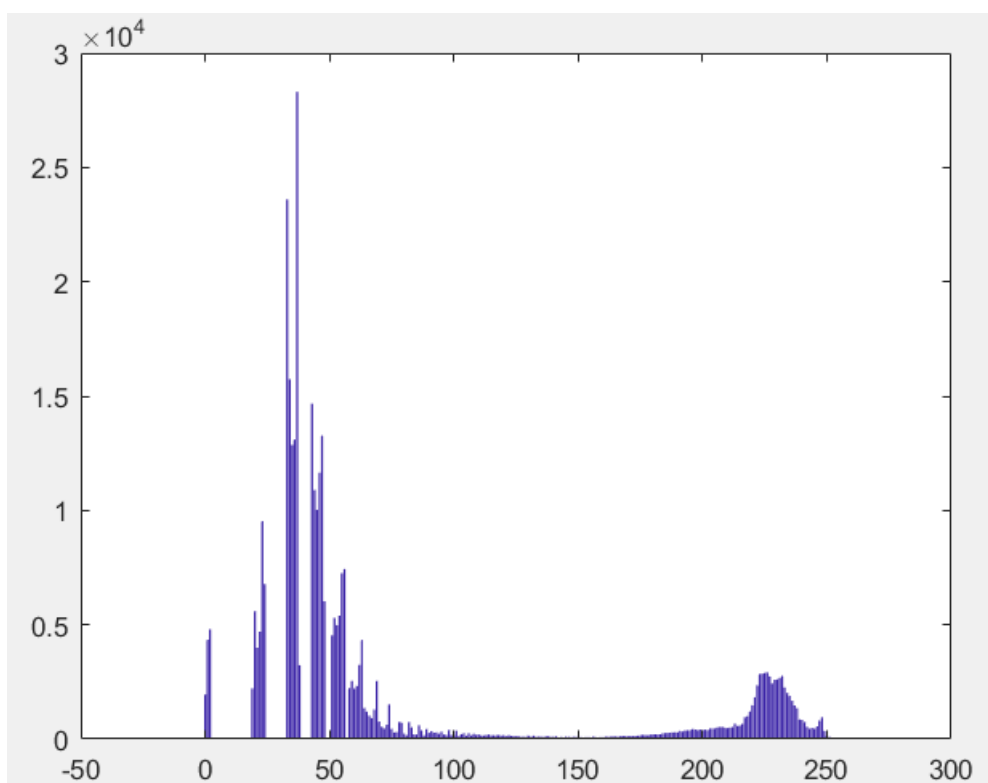


图 7 同态滤波后的灰度直方图

首先，观察直方图均衡化的结果，其灰度直方图显示，尽管某些灰度值仍然比较集中，但整体效果上已将原本密集的灰度分布调整为较为均匀的状态。改进后的图像在灰度对比度上有显著提升，物体的形状更为清晰，灰度级也普遍提高。

接下来，分析同态滤波后的灰度直方图，可以看出低灰度值依然较为集中，但相比原图有所改善。滤波后的图像整体显得较暗，但物体间的对比度明显增强，使得照片中的物体可被辨识。

在直方图均衡化与同态滤波的结果对比中，从视觉感受上看，同态滤波的效果更优。直方图均衡化后，图像出现较大不连续性，给人以突兀之感；而同态滤波虽然整体较暗，视觉效果却显得更加自然。

从原理上讲，直方图均衡化旨在均衡灰度级分布，然而最佳视觉效果的灰度分布并不总是均匀。因此，通过分析图像要素并实施直方图规定化，可以改善最终效果。而同态滤波则通过调整入射与反射部分的频域特性，降低低频成分、增强高频成分，从而提升对比度，使我们能够更清晰地观察暗区物体。由于滤波过程保留了一部分低频成分，整体图像依然保持较为平滑，视觉效果优于直方图均衡化。