

```

% 读取图像并转换为灰度图

img = imread('低照度图像.jpg');

%直接使用 matlab 自带 rgb2gray 函数

gray_img = rgb2gray(img);

figure();

%显示灰度图

imshow(gray_img);


% 计算灰度直方图

num_bins = 256; % 灰度级数

histogram = zeros(1, num_bins); % 初始化直方图


% 计算每个灰度级的像素数量

for i = 1:numel(gray_img)

    pixel_value = gray_img(i);

    histogram(pixel_value + 1) = histogram(pixel_value + 1) + 1;

    % +1 因为 MATLAB 中灰度范围为 0 到 255，而矩阵范围为 1 到 256

end


% 显示灰度直方图

figure;

bar(0:num_bins-1, histogram, 'BarWidth', 1, 'FaceColor', [0.5 0.5 0.5]);

xlim([0 num_bins - 1]);

title('灰度直方图');

xlabel('灰度级');

ylabel('像素数量');


F=fft2(gray_img);

F=fftshift(F);

```

```
F=abs(F);

F=log(1+F);

% 显示结果

figure;

imshow(F, []);

% 计算累计分布函数 (CDF)

cdf = cumsum(histogram);

% 归一化

cdf_normalized = cdf / numel(gray_img);

% 使用 CDF 映射到新的像素值

img_eq = uint8(255 * cdf_normalized(double(gray_img) + 1)); % +1 是因为 MATLAB 的索引从 1 开始

% 进行同态滤波

img_homo = homomorphic_filter(gray_img);

% 显示结果

figure;

subplot(1, 3, 1);

imshow(gray_img);

title('原图');

subplot(1, 3, 2);

imshow(img_eq);

title('直方图均衡化');

subplot(1, 3, 3);

imshow(img_homo);
```

```
title('同态滤波');

% 同态滤波

function J = homomorphic_filter(I)

    % 转换为浮点数

    I = double(I) + 1; % +1 防止对数运算中的零

    % 对成像模型取对数

    I_log = log(I);

    % 进行 FFT 变换

    H = fspecial('gaussian', 15, 5); % 高斯滤波器

    H = H - mean(H(:)); % 中心化滤波器

    F_H = fft2(H, size(I, 1), size(I, 2)); % 滤波器在频域表示

    % FFT

    F = fft2(I_log);

    % 频域乘法

    G = F .* F_H;

    % 反变换

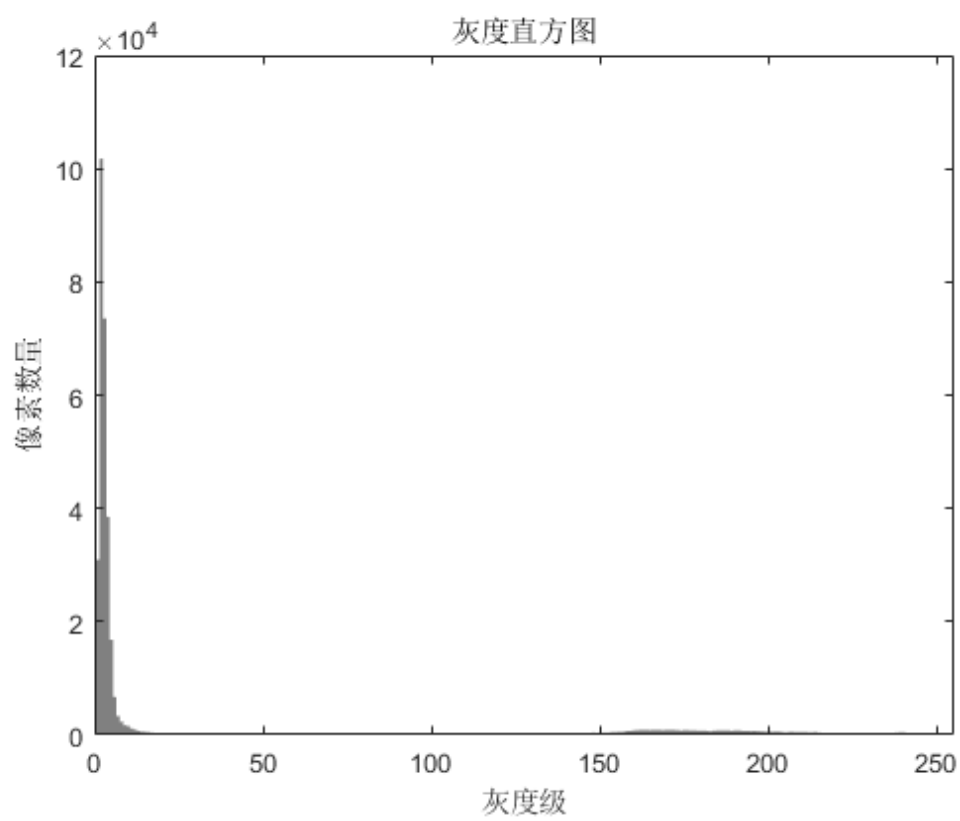
    g = ifft2(G);

    % 取指数

    J = exp(real(g)) - 1; % 反对数

    J = uint8(J / max(J(:)) * 255); % 归一化到 [0, 255]

end
```





原图



直方图均衡化



同态滤波

