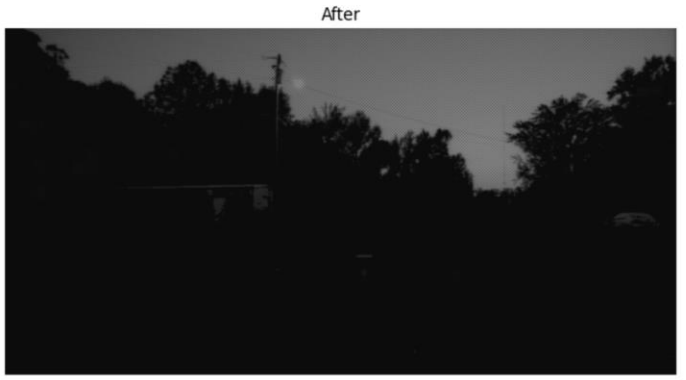
Homework 3

提高 2101 班杨筠松 U202115980

```
function image_out = homomorphic_filter(image_in, high, low, c, d0)
    I = double(image_in);
    [M, N] = size(I);
    a = M / 2;
    b = N / 2;
    log_img = log1p(I);
    log_fft = fftshift(fft2(log_img));
    D = zeros(M, N);
    H = zeros(M, N);

    for i = 1:M
        for j = 1:N
            D(i, j) = sqrt((i - a)^2 + (j - b)^2);
            H(i, j) = (high - low) * (1 - exp(-c * (D(i, j)^2 / d0^2))) + low;
        end
    end

    H = ifftshift(H);
    filtered_fft = H .* log_fft;

    log_ifft = ifft2(filtered_fft);
    image_out = real(expm1(log_ifft));

end
image_in = imread('data/image1.png');
if isempty(image_in)
    disp('Image not found');
else
    % 转为灰度图，如果需要
    if size(image_in, 3) == 3
        image_in = rgb2gray(image_in);
    end

    % 应用同态滤波
    image_out = homomorphic_filter(image_in, 2.0, 0, 1.0, max(size(image_in)));

    % 显示结果
    figure;
    subplot(1, 2, 1);
    imshow(image_in);
```

```
    title('Original');

    subplot(1, 2, 2);
    imshow(image_out, []);
    title('After');
end
```



```
image_path = 'data/image2.png';
img = imread(image_path);
if isempty(img)
    disp('Image not found!');
else

    if size(img, 3) == 3
        img = rgb2gray(img);
    end


    global_hist_eq = histeq(img);

    % CLAHE
    clahe_eq = adapthisteq(img, 'ClipLimit', 0.02, 'NumTiles', [8 8]);
        figure;
    subplot(1, 3, 1);
    imshow(img);
```

```
    title('Original Image');

    subplot(1, 3, 2);
    imshow(global_hist_eq);
    title('Global Histogram Equalization');

    subplot(1, 3, 3);
    imshow(clahe_eq);
    title('CLAHE');
end
```


```
original = imread("cameraman.tif");
figure;
distorted = imread("C:\Users\DELL\Desktop\inclass\DIPlab_source\1.jpg");
subplot(121);imshow(original,[]),title('Original');
subplot(122);imshow(distorted,[]),title('distorted');

[movingPoints,fixedPoints] = cpselect(distorted,original,"Wait",true);

tform = fitgeotform2d(movingPoints,fixedPoints,'polynomial',3);

Roriginal = imref2d(size(original));

recovered = imwarp(distorted,tform,OutputView=Roriginal);

figure;
montage({original,recovered})
```

```
img = imread('redeye.jpg');

% rgb to hsi model, implementation in rgb2hsi
hsi_img = rgb2hsi(img);

% get hue and saturation of img.
hue_channel = hsi_img(:,:,1);
saturation_channel = hsi_img(:,:,2);

% marking red eye region
red_eye_mask = (hue_channel > 0.9) & (hue_channel < 1.0)...
                        & (saturation_channel > 0.4);

% turn 0 for red eye region
saturation_channel(red_eye_mask) = 0;
hsi_img(:,:,2) = saturation_channel;

% from hsi return to rgb
output_img = hsi2rgb(hsi_img);

figure;
subplot(1,2,1); imshow(img); title('original');

subplot(1,2,2); imshow(output_img); title('alleviating red eye effect');
```