

# MatLab图像处理

## 图像表示

灰度是指黑白图像中点的颜色深度，范围一般从0到255，白色为255，黑色为0，故黑白图片也称灰度图像。

```
% 图像读取
clear all % 清除所有变量与窗口
im = imread('image.jpg');

% 图像显示
imshow(im);
imshow(im(:, :, 1)) % 显示红色分量
imshow(im(:, :, 2)) % 显示绿色分量
imshow(im(:, :, 3)) % 显示蓝色分量
for k = 1: 3          % 多个图像同时显示
    subplot(1, 3, k);
    imshow(im(:, :, k));
end

[m, n, p] = size(im);
im1 = zeros(m, n, p);
im1(:, :, 1) = im(:, :, 1);
im1 = uint8(im1);    % zeros生成的矩阵数据类型不是uint8
imshow(im1);
```

## 图像的拼接

```
% 图像拼接
im = [im1; im2]; % 上下拼接
im = [im1, im2]; % 左右拼接

% 图像复制
im3 = [im, im(:, end: -1: 1, :), im]
```

## 信息隐藏

前置知识：

1. 灰度值 $r$ 的 $n$ 阶矩定义为： $\mu_n = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i)$ ，其中 $r_i$ 为灰度值 $i$ 的像素数量， $p(r_i)$ 表示 $i$ 出现的概率
2. 灰度值的平均值 $m$ ： $m = \sum_{i=0}^{L-1} r_i p(r_i)$
3. 方差（灰度对比度测度）： $\sigma^2 = \mu_2 = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$
4. 领域 $S_{\{xy\}}$ 的像素均值（平均亮度）： $m_{\{S_{\{xy\}}\}} = \sum_{i=0}^{L-1} r_i p(S_{\{xy\}}(r_i))$ ， $(x, y)$ 为图片中心的坐标

5. 灰度变换公式：其中 $k_0$ 一般取0,  $k_1 \in [0.1, 0.25]$ ,  $k_2 = 0$ ,  $k_3 = 0.1$ ,  $C = 22.8$ , 情况一表示局部亮度较低或局部标准差小于全局标准差（增大低灰度值和低对比度像素的灰度值）

$$g(x, y) = \begin{cases} Cf(x, y) & \text{if } k_0 m_G \leq m_{\{S_{xy}\}} \leq k_1 m_G \text{ and } k_2 \sigma_G \leq \sigma_{\{S_{xy}\}} \leq k_3 \sigma_G \\ f(x, y) & \text{otherwise} \end{cases}$$

```
% 隐藏信息的显示
clear all;
f = imread('1_1.jpg');
subplot(1 2 3), imshow(f), title('original image')
f = double(f);

% 计算全局数据和局部数据
gm = mean(f(:)); % 计算全局的平均灰度
gsigma = std(f(:)); % 计算全局的灰度标准差
[M, N] = size(f);
lm = zeros(M, N);
lsigma = zeros(M, N); % 提前准备好标准差的矩阵
for x = 2: M - 1
    for y = 2: N - 1
        Sxy = f(x - 1: x + 1, y - 1: y + 1);
        lm(x, y) = mean(Sxy(:)); % 计算中心为(x, y)的3x3矩阵的局部平均灰度
        lsigma(x, y) = std(Sxy(:));
    end
end

% 照片处理
g = f; % g为处理后的照片
for x = 2: M - 1
    for y = 2: N - 1
        if lm(x, y) > 0 && lm(x, y) < 0.1 * gm && lsigma(x, y) > 0 && lsigma(x, y) < 0.1 * gsigma
            g(x, y) = 22.8 * f(x, y) % 对满足条件的像素点进行平均灰度放大
        end
    end
end
subplot(1, 2, 2), imshow(g), title('result image');
```

## 领域滤波

领域滤波就是将数字信号中的某些特征频率保留或加强，同时滤除或减弱另外一部分的频率成分。

离散傅里叶变换（DFT, Discrete Fourier Transform）、快速傅里叶变换（FFT, Fast Fourier Transform）。实现FFT的函数为FFT，对二维数字图像则需要用函数FFT2。

```
clc % 清空显示屏
clear % 清空数据
im = imread('1_1.jpg');
im1 = rgb2gray(im); % 转化为灰度图像
im2 = imresize(im1, [512, 512]); % 傅里叶变化能高效进行2的n次方大小图片
```

```
imshow(im2), title('original image');

f = double(im2); % 便于计算
F = fft2(f);      % 变换 ( 负数矩阵 )
S = abs(F);       % 取绝对值
imshow(S, []);    % '[]'表示将数据归一化
title('Frequency Spectrum'); % 黑色频谱图
imshow(abs(F(1: end / 16, 1: end / 16)), []); % 左上角出现白点
Fc = fftshift(F); % 实现低频分量和直流分量居中
S = abs(Fc);
imshow(S, []);
title('Centered Frequency Spectrum');

S2 = log(1 + S);
imshow(S2);      % 取对数以扩大范围
[M, N] = size(f);
sig = 10;
H = lpfilter('gaussian', M, N, sig); % 低通高斯滤波器，卷积
imshow(1 - H, []); % '1 - H'显示为高通滤波器，四个角为白色
G = H .* F;
g = real(ifft2(G)); % 低频为边界，高频为细节，不同滤波器只能让该频率的信号通过，因此用
低通滤波器处理后的图像变模糊
imshow(g, []);
```