

实验一 数字图像的基本操作和灰度变换

一、实验目的

1. 了解数字图像的基本数据结构
2. 熟悉 Matlab 中数字图像处理的基本函数和基本使用方法
3. 掌握图像灰度变换的基本理论和实现方法
4. 掌握直方图均衡化增强的基本理论和实现方法

二、实验原理与方法

1. 图像灰度的线性变换

灰度的线性变换可以突出图像中的重要信息。通常情况下，处理前后的图像灰度级是相同的，即处理前后的图像灰度级都为 $[0, 255]$ 。那么，从原理上讲，我们就只能通过抑制非重要信息的对比度来腾出空间给重要信息进行对比度展宽。

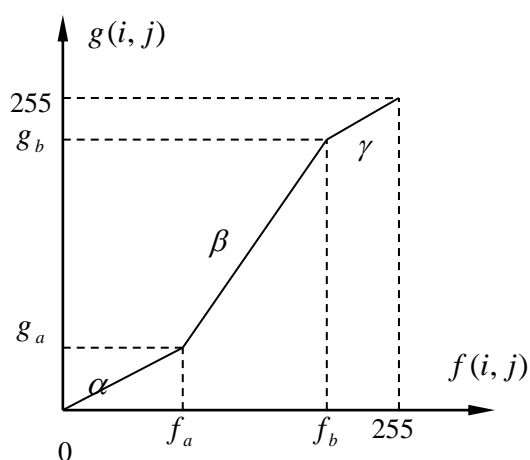


图 1.1 对比度线性变换关系

设原图像的灰度为 $f(i, j)$ ，处理后的图像的灰度为 $g(i, j)$ ，对比度线性展宽的原理示意图如图 1.1 所示。假设原图像中我们关心的景物的灰度分布在 $[f_a, f_b]$ 区间内，处理后的图像中，我们关心的景物的灰度分布在 $[g_a, g_b]$ 区间内。在这里 $\Delta g = (g_b - g_a) > \Delta f = (f_b - f_a)$ ，也就是说我们所关心的景物的灰度级得到了展宽。根据图中所示的映射关系中分段直线的斜率我们可以得出线性对比度展

宽的计算公式：

$$g(i, j) = \begin{cases} \alpha f(i, j), & 0 \leq f(i, j) < f_a \\ \beta(f(i, j) - f_a) + g_a, & f_a \leq f(i, j) < f_b \\ \gamma(f(i, j) - f_b) + g_b, & f_b \leq f(i, j) < 255 \end{cases} \quad (1-1)$$

$(i = 1, 2, 3 \dots, m; j = 1, 2, 3 \dots, n)$

其中， $\alpha = \frac{g_a}{f_a}$ ， $\beta = \frac{g_b - g_a}{f_b - f_a}$ ， $\gamma = \frac{255 - g_b}{255 - f_b}$ ，图像的大小为 $m \times n$ 。

2. 直方图均衡化

灰度直方图反映了数字图像中每一灰度级与其出现频率间的关系，它能描述图像的概貌。通过修改直方图的方法增强图像是一种实用而有效的处理技术。直方图均衡化是将原始图像通过某种变换，得到一幅灰度直方图为均匀分布的新图像的方法。

离散图像均衡化处理可通过变换函数：

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n}$$

来实现

三、实验内容与步骤

1. 熟悉 MATLAB 语言中数字图像处理函数的使用。

2. 图像灰度线性变换的实现

1) 读入一幅灰度图像 test1，显示其灰度直方图

2) 根据图像灰度直方图，选择所关心的图像景物的灰度分布范围 $[f_a, f_b]$ ，以及拟变换的灰度分布范围 $[g_a, g_b]$

3) 实现对图像的灰度线性变换

4) 调整 α ， β ， γ 的值，观察对处理结果的影响。

3. 图像的均衡化处理

1) 读入一幅灰度图像 test2，求出其直方图

2) 利用 Matlab 函数实现图像的均衡化处理

3) 同屏显示处理前后的图像和灰度直方图，说明处理前后直方图的变化以及对应的灰度变化

四、思考问题

1. 在映射关系中，分段直线的斜率的大小对图像处理结果有哪些影响？
2. 在进行对比度扩展时，如果确定和选取所关心的景物？
3. 直方图均衡化适用于什么形式的灰度分布情形？

五、附灰度线性变换部分实现代码

设计两重循环（即：for (i=1:m) for(j=1:n)）按照公式（1-1），给出新图像的每个像素点的灰度值；

（因为在 MATLAB 中，图像的像素值为 uint8 型数据描述的，而这种类型不允许进行算术运算，所以要对 F 进行数据类型转换，即：f=double(f)；然后，在进行循环处理。）

设输入图像 $f(i, j)$ ，输出图像仍放入 $f(i, j)$ 中，图像大小为 $m \times n$

```
for i=1:m
    for j=1:n
        if f(i, j)<fa
            f(i, j)=a* f(i, j);
        elseif f(i, j)<fb
             $f(i, j) = \beta * (f(i, j) - f_a) + g_a;$ 
        else
             $f(i, j) = \gamma * (f(i, j) - f_b) + g_b$ 
        end
    end
end
```

六、实验报告要求

1. 简述实验目的和实验原理。
2. 编写 MATLAB 程序，实现相应功能。
3. 分析实验结果。
4. 回答思考问题。

实验二 图像的傅里叶变换及其性质

一、实验目的

1. 熟悉图像空间域和频率域的关系，掌握快速傅里叶变换
2. 掌握离散傅里叶变换的性质和应用

二、实验原理与方法

图像既能在空间域处理，也能在频率域处理。把图像信息从空域变换到频域，可以更好地分析、加工和处理

二维离散傅立叶正变换的表达式为

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$
$$u = 0, 1, 2, \dots, M-1; \quad v = 0, 1, 2, \dots, N-1$$

逆变换为：

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$
$$x = 0, 1, 2, \dots, M-1 \quad y = 0, 1, 2, \dots, N-1$$

二维离散傅立叶变换具有若干性质，如：线性性、平移性、可分离性、周期性、共轭对称性、旋转不变性等。

三、实验内容与步骤

1. 产生一幅如图所示亮块图像 $f(x, y)$ (500×500 大小、暗处=0，亮处=255)：



- (1) 对其进行二维傅里叶变换，同屏显示原图 f 和其傅里叶变换幅度谱图；
- (2) 若令 $f_1(x, y) = (-1)^{x+y} f(x, y)$ ，重复以上过程，比较二者幅度谱的异同，简述理由；
- (3) 若将 $f_1(x, y)$ 顺时针旋转 45 度得到 $f_2(x, y)$ ，试求出并显示 $f_2(x, y)$ 的傅里叶幅度谱，并与 $f_1(x, y)$ 的幅度谱进行比较和说明。

2. 读入一幅数字图像 test6

- (1) 对其进行二维傅里叶变换，显示其傅里叶变换的幅度谱和相位谱。
- (2) 令其幅度谱为常数 A ，仅利用相位谱反傅里叶变换重建原图像，说明重建图像的特点

(3) 令相位谱为 0，仅利用幅度谱进行反傅里叶变换重建原图像，分析说明重建图像的特点。

(4) 使用题 1 的幅度谱与 test6 的相位谱进行反傅里叶变换，分析说明重建图像的特点。

四、思考问题

1. 在傅里叶反变换中，幅度谱和相位谱对重建图像的影响分别体现在什么方面？

五、实验报告要求

1. 简述实验目的和实验原理。
2. 编写 MATLAB 程序，实现相应功能。
3. 分析实验结果。
4. 回答思考问题。

实验三 图像的空间域和频率域增强

一、实验目的

1. 熟悉图像空间域增强方法，掌握增强模板使用方法
2. 掌握均值滤波器、中值滤波器的理论基础和实现方法
3. 掌握频率域滤波的基本理论和实现方法

二、实验原理与方法

图像增强是数字图像处理的基本内容之一，其目的是根据应用需要突出图像中的某些“有用”信息，削弱或去除不需要的信息，以改善图像的视觉效果，或突出图像的特征，便于计算机处理。图像增强可以在空间域进行，也可以在频率域中进行。

1. 空间域增强滤波

空间域滤波主要利用空间模板进行，如 3×3 ， 5×5 模板等。

一般来说，使用大小为 $m \times n$ 的滤波器对大小为 $M \times N$ 的图像 f 进行空间滤波，可表示成：

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

其中， $m=2a+1$ ， $n=2b+1$ ， $w(s, t)$ 是滤波器系数， $f(x, y)$ 是图像像素值

均值滤波器是一种空间平滑滤波器，它是对包含噪声的图像上的每个像素点，用它邻域内像素的平均值替代原来的像素值。例如，采用一个 3×3 的模板，待处理的像素为 $f(i, j)$ ，则处理后图像对应的像素值为

$$g(i, j) = 1/9 * (f(i-1, j-1) + f(i-1, j) + f(i-1, j+1) + f(i, j-1) + f(i, j) + f(i, j+1) + f(i+1, j-1) + f(i+1, j) + f(i+1, j+1));$$

中值滤波器也是一种空间平滑滤波器，它是对以图像像素点为中心的一个滑动窗口内的诸像素灰度值排序，用中值代替窗口中心像素的原来灰度值，因此它是一种非线性的图像平滑法。

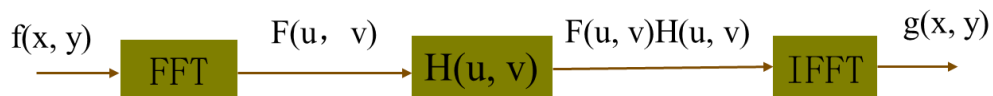
空间域锐化可采用一阶微分算子，如梯度算子、Roberts、Prewitt 和 Sobel 算子等，也可采用二阶 Laplacian 锐化算子，它们的目都是突出图像的边缘或轮廓。微分算子也可以写成模板作用的形式，如二阶 Laplacian 锐化算子可写为：

$$g(i, j) = 4 * f(i, j) - (f(i-1, j-1) + f(i, j-1) + f(i, j+1) + f(i+1, j));$$

其中， $f(i, j)$ 为待处理的像素值， $g(i, j)$ 为处理后图像对应的像素值。

2. 频率域滤波

图像频率域滤波可利用离散傅里叶变换，将图像信号从空间域变换到频率域，在频率域选择合适的滤波器 $H(u, v)$ 对图像的频谱成分 $F(u, v)$ 进行处理，然后经逆傅立叶变换得到处理图像，实现图像处理结果。频率域滤波的一般过程如下：



其中
$$g(x, y) = \text{IFFT}[F(u, v)H(u, v)]$$

由于噪声主要集中在高频部分，为去除噪声改善图像质量，图像平滑滤波采用低通滤波器 $H(u, v)$ 来抑制高频成分，通过低频成分，达到平滑图像的目的。而图像锐化是为了消除模糊，突出边缘和细节，因此采用高通滤波器让高频成分通过，使低频成分削弱，再经逆傅立叶变换得到边缘锐化的图像。

三、实验内容与步骤

1. 读入一幅数字图像 test3

2. 图像的平滑滤波处理

1) 对原图像分别加入高斯噪声、椒盐噪声。

2) 利用邻域平均法，分别采用 3×3 ， 5×5 模板对加噪声图像进行平滑处理，显示原图像、加噪图像和处理后的图像。

3) 利用中值滤波法，分别采用 3×3 ， 5×5 模板对加噪声图像进行去噪处理，显示原图像、加噪图像和处理后的图像。

4) 比较各种滤波方法和滤波模板的处理结果，并分析说明。

3. 读入一幅数字图像 test5，选择适当的滤波半径，在频域分别进行理想低通、理想高通、高斯低通、高斯高通滤波，显示原始图像及它的傅里叶幅度谱图，以及低通、高通滤波后的结果图像及它们的傅里叶幅度谱，分析说明滤波处理后图像与原始图像的区别。

四、思考问题

1. 采用均值滤波、中值滤波，对高斯噪声和椒盐噪声的抑制哪种比较有效？
2. 模板大小的不同，所处理效果有何不同？为什么？
3. 频率域滤波在实现过程应如何处理？

五、实验报告要求

1. 简述实验目的和实验原理。
2. 编写 MATLAB 程序，实现相应功能。
3. 分析实验结果。
4. 回答思考问题。

实验四 图像的微分运算及其边缘检测

一、 实验目的

1. 学习和了解边缘的基本特征
2. 掌握使用一阶微分算子和二阶微分算子进行图像锐化的原理和方法；
3. 掌握使用一阶微分算子和二阶微分算子进行图像边缘检测的原理和方法；
4. 通过对比图像二阶微分信息，进一步掌握图像锐化和边缘检测的实质；

二、实验原理与方法

边缘是图像的重要特征，是图像分割的一个重要依据。不同区域的交界(边缘)处像素值具有不连续性或突变性，可根据微分算子检测出交界处的灰度变化，从而确定区域交界处的点、线或边，分割出不同区域。

图像处理中应用微分最常用的方法是计算梯度。对于图像 $f(x, y)$ ，在点 (x, y) 处的梯度定义为：

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

梯度是一个向量，其大小和方向为：

$$M(x,y)=\text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

$$\theta = \text{tg}^{-1}(g_y / g_x) = \text{tg}^{-1}\left(\frac{\partial f(x,y)}{\partial y} / \frac{\partial f(x,y)}{\partial x}\right)$$

离散图像处理时，常用到梯度的大小，因此习惯称梯度的大小为“梯度”。

并且一阶偏导数采用一阶差分近似表示，即：

$$g_x = f(x+1,y) - f(x,y)$$

$$g_y = f(x,y+1) - f(x,y)$$

为简化梯度的计算，可令梯度值为

$$M(x,y)=|g_x| + |g_y|, \text{ 或 } M(x,y)=\max\{|g_x|, |g_y|\}$$

除梯度算子以外，其它常用的一阶微分算子有 Roberts、Prewitt 和 Sobel 算子等，同时也可采用二阶 Laplacian 微分算子来进行边缘检测。

二阶 Laplacian 锐化算子可写为：

$$g(i, j) = (f(i-1, j-1) + f(i, j-1) + f(i, j+1) + f(i+1, j)) - 4*f(i, j) ;$$

其中, $f(i, j)$ 为待处理的像素值, $g(i, j)$ 为处理后图像对应的像素值。

三、实验内容与步骤

1. 图像的微分运算

1) 分别利用梯度算子、Roberts、Prewitt 和 Sobel 等一阶微分算子, 对数字图像 test7、test8 进行微分运算, 计算出梯度幅值, 显示处理前、后图像。

2) 利用 Laplacian 二阶微分算子对数字图像 test7、test8 进行微分运算, 计算出微分幅值, 显示处理前、后图像。

2. 图像的边缘检测

1) 选择适当的阈值分别对一阶和二阶微分算子处理后的图像进行阈值化处理, 检测出边缘。显示阈值化处理后的图像。

2) 在 test7 图像中砖墙结构产生较多的细节分量, 它们往往表现为噪声, 微分计算增强了这些噪声, 从而使图像中主要边缘的检测变得复杂。为了减少这些噪声, 可先对图像进行平滑处理, 如采用 5×5 的均值平滑滤波器, 然后再进行微分运算和阈值化。编程重复这一过程, 显示处理过程图像和结果图, 分析比较与没有平滑过程的结果。

四、思考问题

1. 各种一阶微分算子在检测图像细节时, 有什么不同特点?
2. 一阶微分算子和二阶微分算子在提取图像细节信息时, 有什么异同?
3. 边缘检测时, 阈值选择对提取边缘有何影响?

五、实验报告要求

1. 简述实验目的和实验原理。
2. 编写 MATLAB 程序, 实现相应功能。
3. 分析实验结果。
4. 回答思考问题。