Notes of Digital Image Processing

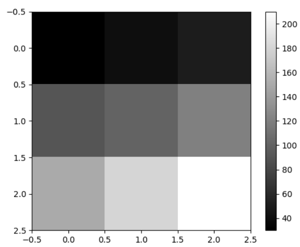
Zebediah

2024 年 6 月 28 日

前言

这是一份数字图像处理 (DIP) 的笔记, 主要参考 Rafael C. Gonzalez, Richard E. Woods 编著的 *Digital Image Processing, Fourth Edition.* 这份笔记并不会像书中写得十分详细, 而更侧重于对一些有用的知识点进行汇总, 因此笔记中会省略很多描述性细节. 在笔记中, 我们将默认已经学习过本科数学专业基础课程, 因此不再一一介绍预备知识. 如若遇到笔记中没有学习过的专业知识, 读者可自行查阅.

这里, 我将对 DIP 做一些简单的描述. DIP 是使用计算机对数字图像进行处理的技术和方法, 图像可以转换成一个矩阵, 同样矩阵也可以转换成一个矩阵, 如图 1 所示, 图像有九块像素, 每个像素值对应着矩阵的每个元素 (这里我们取的像素值范围是 0-255).



$$\xleftrightarrow[\text{To Image}]{\text{To Matrix}} \begin{pmatrix} 30 & 40 & 50 \\ 90 & 100 & 120 \\ 150 & 180 & 210 \end{pmatrix}$$

图 1: 图像和矩阵可以相互转换

因此, 对图像的处理问题, 可以转换成处理图像所对应的矩阵, 从而可以进一步利用线性代数、计算机等工具对其进行不同的分析、变换和处理. 当然, 我们也很容易联想到: 一个视频可以看作成一个三维的张量, 它的帧数构成张量的一个维度, 而每一帧对应的图像构成张量的其余两个维度. 通过推广至高维张量, 可以更好地描述图像的各种特征, 如我们刚才所讲的视频, 又比如一个彩色图像 (多出了颜色通道的维度) 等.

总的来说, 笔者还是围绕参考书的框架撰写此笔记, 此书一共 700 多页, 所以在此笔记中会省去大量篇幅, 难免会有一些逻辑不清之处. 同时, 笔者也是在学习过程中, 如有不妥之处, 敬请批评指正!

Zebediah

2024 年 6 月 28 日

目录

Chapter 1

数字图像基础

1.1 图像取样和量化

1.1.1 取样的量化的基本概念

对于一幅连续图像 f, 我们需要把它转换为数字形式. 一幅图像的 x 坐标和 y 坐标是连续的, 其幅度也是连续的. 要将函数数字化, 就要对该函数的坐标和幅度 (即灰度值) 进行取样. 对坐标值进行数字化称为取样 (或采样), 对幅度值进行数字化称为量化.

1.1.2 数字图像表示

我们把该连续图像取样为一幅数字图像 $f(x,y)$, 该图像包含 M 行和 N 列, 其中 (x,y) 是离散坐标. 有三种基本方法表示 $f(x,y)$:

(a) $f(x,y)$ 是 x 和 y 的函数, 对每个 (x,y), 将函数值 $f(x,y)$ 作为第三个维度的值, 绘制出的图像是形式为 $(x,y,f(x,y))$ 的三维图像;

(b) 第二种表示的是 $f(x,y)$ 出现在计算机显示器或照片上的情况, 每个点的灰度与该点处的 f 值成比例. 如果这些灰度值被归一化到区间 $[0,1]$, 那么图像中每个点的灰度值都是 0, 0.5 或 1, 即黑色、灰色或白色;

(c) 第三种表示是由数值 $f(x,y)$ 组成的一个阵列 (矩阵):

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix},$$

该式的右边是以实数矩阵表示的数字图像, 矩阵中的每个元素称为图像单元、图像元素或像素. 使用

传统的矩阵表示法来表示数字图像及其像素更为方便:

$$
\boldsymbol{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix},
$$

其中 $a_{ij} = f(i, j)$.

图 1.1 给出了图像三种表示方法的示例.

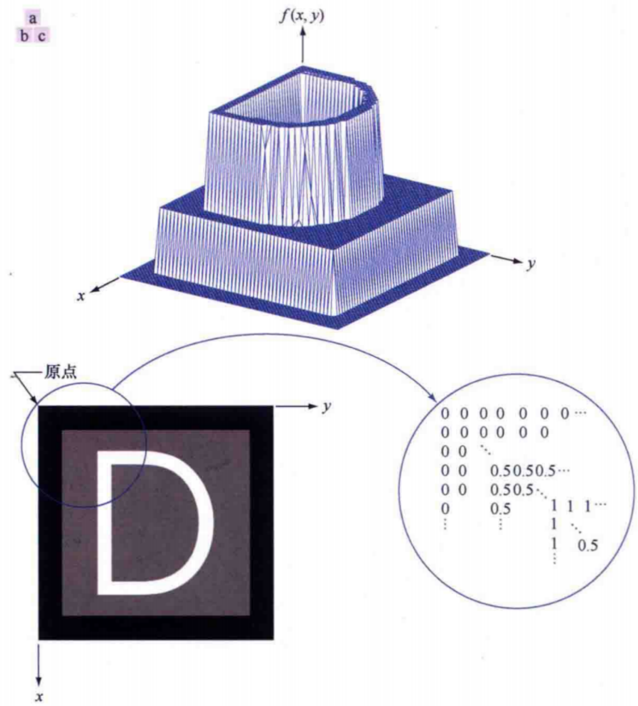


图 1.1: 图像的三种表示

图像数字化要求对 M 值、N 值和离散灰度级数 L 进行判定. 对于 M 和 N, 只要求取正整数. 而对于灰度级数 L, 它描述了图像中不同像素的灰度值分布, 通常取为 2 的整数次幂 (记为 2^k), 这些灰度值通常在 0 到 2^k 之间, 其中 0 表示纯黑, 2^k 表示纯白, 介于两者之间的值表示不同程度的灰色, 而区间 $[0, 2^k]$ 称为灰度级.

1.1.3 空间分辨率和灰度分辨率

空间分辨率是是图像中可以分辨的最小细节或像素的密度. 空间分辨率通常以每单位长度的像素数来表示, 例如像素每英寸或像素每厘米. 更高的空间分辨率意味着图像中可以显示更多的细节, 图像更清晰.

灰度分辨率是图像中每个像素可以表示的灰度级别的数量. 灰度分辨率通常用每个像素的比特数来表示, 例如, 8 位灰度分辨率意味着每个像素可以表示 256 个灰度级别 (2^8).

1.1.4 图像内插

内插通常在图像放大、缩小、旋转和几何校正等任务中使用. 内插是用已知数据来估计未知位置的值的过程, 这里我们主要介绍最近邻内插、双线性内插和双三次内插.

最近邻内插: 将原图像中最近邻的灰度赋给了每个新位置, 这会造成某些直边缘严重失真.

双线性内插: 使用 4 个最近邻的灰度来计算给定位置的灰度. 具体地, 所赋的值由如下公式得到:

$$v(x, y) = ax + by + cxy + d,$$

其中 4 个系数可用由点 (x, y) 的 4 个最近邻点写出的 4 个未知方程求出. 此方法结果会好得多, 但计算量会随之增大.

双三次内插: 使用 16 个最近邻的灰度来计算给定位置的灰度. 具体地, 所赋的值由如下公式得到:

$$v(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j,$$

其中 16 个系数可用由点 (x, y) 的 16 个最近邻点写出的 16 个未知方程求出.

1.2 像素间的一些基本关系

1.2.1 像素的相邻像素

坐标 (x, y) 处的像素 p 有 2 个水平的相邻像素和 2 个垂直的相邻像素, 它们的坐标是

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1),$$

这组像素称为 p 的 4 邻域, 用 $N_4(p)$ 表示.

p 的 4 个对角相邻像素的坐标是

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1),$$

用 $N_D(p)$ 表示. 这些相邻像素和 4 邻域合称为 p 的 8 邻域, 用 $N_8(p)$ 表示. 图示可见图 1.2. 此外, 点 p 的相邻像素的图像位置集称为 p 的邻域. 如果一个邻域包含 p, 那么称该邻域为闭邻域, 否则称该邻域为开邻域.

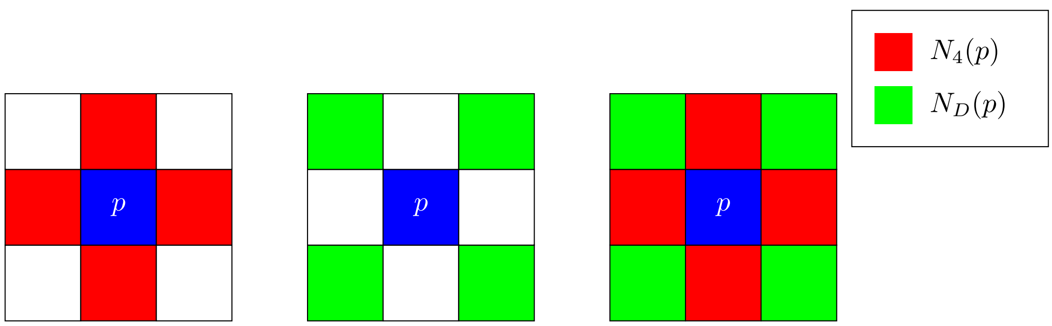


图 1.2: 像素的相邻像素. 从左到右分别是 $N_4(p)$, $N_D(p)$, $N_8(p)$

1.2.2 邻接、连通、区域和边界

这里介绍三种类型的**邻接**:

- **4 邻接:** q 在集合 $N_4(p)$ 中时, 称 p 和 q 是 4 邻接的;
- **8 邻接:** q 在集合 $N_8(p)$ 中时, 称 p 和 q 是 8 邻接的;
- **m 邻接 (混合邻接):** q 在 $N_4(p)$ 中, 或 q 在 $N_D(p)$ 中且 $N_4(p) \cap N_4(q) = \varnothing$, 称 p 和 q 是 m 邻接的.

设像素 p 和 q 的坐标分别为 (x_0, y_0) 和 (x_n, y_n), 若有一系列像素点 (x_{i-1}, y_{i-1}) 和 $(x_i, y_i), i = 1, \cdots, n$ 是邻接的, 则它们构成 p 和 q 的一条**通路**, n 为通路的长度. 特别地, 当 $(x_0, y_0) = (x_n, y_n)$ 时, 通路是**闭合通路**. 根据不同的邻接定义, 可以类似地定义不同的通路. 设 p 和 q 是一个图像子集 S 中的两个元素, 如果存在一条完全由 S 中的像素组成的从 p 和 q 的通路, 则称 p 在 S 中与 q **连通**.

图像里每个连通集构成图像中的一个**区域**, 图像可以看作是这些区域的并集. 区域的**边界**是该区域的一个子集, 同时要满足在其邻域中有不属于该区域的像素, 它将该区域与其他区域分开. 注意, 我们必须指定连接类型才能确定一个像素是否属于边界.

1.2.3 距离测度

距离需要满足**正定性、对称性、三角不等式**. 这里给出两个像素点 $p(x, y), q(u, v)$ 之间的常用距离.

- **欧氏距离:** $D_e(p, q) = [(x-u)^2 + (y-v)^2]^{\frac{1}{2}}$;
- **城区距离:** $D_4 = |x-u| + |y-v|$;
- **棋盘距离:** $D_8 = \max(|x-u|, |y-v|)$;
- **D_m 距离:** 像素点之间的最短 m 通路的长度.

Chapter 2

灰度变换与空间滤波

2.1 一些基本的灰度变换函数

2.1.1 图像反转

灰度级在区间 $[0, L-1]$ 内的反转图像的形式为

$$s = L - 1 - r. \tag{2.1}$$

采用这种方式反转图像的灰度级, 会得到类似于照片底片的结果.

2.1.2 对数变换、幂律 (伽马) 变换和分段线性变换函数

对数变换的通式为

$$s = c\log(1+r), \tag{2.2}$$

其中 c 是一个常数, $r \geq 0$. 这个变换将输入中范围较窄的低灰度值映射为输出中范围较宽的灰度级, 我们使用这类变换来扩展图像中的暗像素值, 同时压缩高灰度级值. 反对数 (指数) 变换的功能正好相反.

幂律 (伽马) 变换的形式为

$$s = cr^{\gamma}, \tag{2.3}$$

其中 c 和 γ 是正常数. 幂律曲线用分数值 γ 将较窄范围的暗输入值映射为较宽范围的输出值, 将高输入值映射为较窄范围的输出值. 当 $\gamma > 1$ 时效果正好相反.

分段线性变换函数即变换是一个分段函数, 其形式可以任意复杂, 但要求输入很多参数. (2.2) 和 (2.3) 和分段线性变换函数的原理同 (2.1) 是类似的, 此处不再赘述.

2.2 直方图处理

令 $r_k, k = 0, 1, 2, \cdots, L-1$ 表示一幅 L 级灰度数字图像 $f(x, y)$ 的灰度. f 的非归一化直方图定义为

$$h(r_k) = n_k, \quad k = 0, 1, 2, \cdots, L-1,$$

其中 n_k 是 f 中灰度为 r_k 的像素的数量, 并且细分的灰度级称为直方图容器. 类似地, f 的归一化直方图定义为

$$p\left(r_k\right)=\frac{h\left(r_k\right)}{MN}=\frac{n_k}{MN},$$

其中 M 和 N 分别是图像的行数和列数. 多数情况下我们处理的是归一化直方图, 我们将这种直方图简单地称为直方图或图像直方图. 对 k 的所有值, $p\left(r_k\right)$ 的和总是 1.

2.2.1 直方图均衡化

直方图均衡化的主要思想是通过重新分配图像的像素值, 使得图像的灰度分布更加均匀, 从而增强图像的对比度, 其步骤如下:

- 计算原始图像的灰度直方图, 即统计每个灰度级别在图像中的像素数量;
- 计算每个灰度级别在均衡化后图像中的累积分布函数 (CDF), CDF 表示的是在当前灰度级别之前的所有灰度级别的像素数量占总像素数量的比例;
- 将均衡化后图像中的每个像素的灰度值替换为原始图像中对应灰度值的 CDF 值乘以最大灰度级别.

2.2.2 直方图匹配 (规定化)

直方图规定化是指对图像进行对比度调整, 使其直方图匹配给定的参考直方图. 这个技术通常用于标准化不同图像之间的光照条件, 或者将图像的对比度调整到特定的效果. 其步骤可概括如下:

- 统计两幅图像中每个灰度值的像素个数, 分别得到原始图像和目标图像的直方图;
- 分别计算原始图像和目标图像的 CDF;
- 将原始图像的 CDF 与目标图像的 CDF 匹配, 找到每个原始灰度值对应的目标灰度值, 生成灰度值映射表;
- 用映射表将原始图像的灰度值替换为目标灰度值, 生成匹配后的图像.

2.2.3 局部直方图处理

局部直方图处理的主要思想是定义一个邻域, 然后将邻域中心沿着图像从左到右、从上到下一个一个像素的移动, 每移动一个像素, 就针对该像素位置的邻域的灰度进行直方图变换, 但最终只更新邻域中心点的灰度值. 此方法可以在图像的局部区域进行直方图调整, 增强图像的局部对比度, 改善图像细节.

2.2.4 使用直方图统计量增强图像

使用直方图统计量增强图像是基于直方图的统计量信息 (如均值和方差) 对图像的灰度和对比度进行调整. 增强后的图像 $g(x,y)$ 与原始图像 $f(x,y)$ 的修正公式为

$$g(x,y)=\begin{cases}Cf(x,y), & k_0m_G\le m_{S_{xy}}\le k_1m_G, k_2\sigma_G\le\sigma_{S_{xy}}\le k_3\sigma_G,\\ f(x,y), & \text{其他},\end{cases}$$

其中 C,k_0,k_1,k_2 和 k_3 是规定的常数, m_G 是输人图像的全局均值, σ_G 是输人图像的标准差, 参数 $m_{S_{xy}}$ 和 $\sigma_{S_{xy}}$ 分别是局部均值和标准差.

2.3 空间滤波

空间滤波通过把每个像素的值替换为该像素及其邻域的函数值来修改图像. 如果对图像像素执行的运算是线性的, 那么称该滤波器为**线性空间滤波器**, 否则称该滤波器为**非线性空间滤波器**.

2.3.1 线性空间滤波和卷积

线性空间滤波器在图像 f 和滤波器核 w 之间执行乘积之和运算. 核是一个矩阵, 其大小定义了运算的邻域, 其系数决定了该滤波器的性质. 对于大小为 $m \times n$ 的核, 假设 $m = 2a + 1$ 和 $n = 2b + 1$, 则大小为 $m \times n$ 的核对大小为 $M \times N$ 的图像的线性空间滤波可以表示为

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

其中 $w(s, t)$ 为滤波器在 (s, t) 处的系数. 上式右端也称为核与图像 $f(x, y)$ 的**相关**.

线性空间滤波和卷积的概念比较类似, 二者都需要掠过图像中每个像素计算乘积之和, 而卷积只是把滤波器旋转了 $180°$, 即

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

如果空间滤波器核可以写成两个向量的乘积, 即 $G(x, y) = G_1(x) G_2(y)$, 则称该滤波器核是**可分离的**. 由秩不等式可以知道, 一个 (非零的) 列向量和一个 (非零的) 行向量相乘得到的矩阵, 其秩总是 1. 因此, 要确定一个核是否可分离, 只需确定其秩是否为 1.

2.3.2 平滑 (低通) 空间滤波器

平滑空间滤波器用于降低灰度的急剧过渡. 这里主要介绍盒式滤波器和低通高斯滤波器, 它们都是线性滤波器.

- **盒式滤波器核:** 它是最简单的可分离低通滤波器核, 其系数的值相同 (通常为 1), 其前面有一个归一化的常数, 这个常数一般为 1 除以系数值之和.
- **低通高斯滤波器核:** 所选的核具有圆对称性, 它基于高斯函数, 其作用是减少图像中的高频成分, 从而达到平滑图像的效果. 高斯核

  $$w(s, t) = G(s, t) = K \exp\left(-\frac{s^2 + t^2}{2\sigma^2}\right)$$

  是唯一可分离的圆对称核. 两个高斯函数的乘积和卷积也是高斯函数.

2.3.3 统计排序 (非线性) 滤波器

统计排序滤波器是一类非线性滤波器, 通过对像素值进行排序和选择来达到滤波目的, 在去除噪声和保持边缘方面具有显著效果. **中值滤波器**是最常用的统计排序滤波器, 它通过选择窗口内像素值的中值来替代中心像素值, 能够有效去除椒盐噪声 (图像中白点和黑点形式的噪声), 同时保留图像的边缘细节.

相应的也有**最大值滤波器**和**最小值滤波器**, 即选择窗口内的最大 (小) 像素值来替代中心像素值, 它们能够增强图像中的亮 (暗) 点.

2.3.4 锐化 (高通) 空间滤波器

锐化的作用是突出灰度中的过渡, 用于增强图像的细节和边缘, 使图像看起来更加清晰. 这里首先说明, 数字函数的导数是用差分来定义的. 如 $f(x)$ 的一阶导数和二阶导数定义为

$$\frac{\partial f}{\partial x} = f(x+1) - f(x), \quad \frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x).$$

下面介绍一些常见的锐化空间滤波器:

- **Laplace 滤波器:** 它通过计算像素值的二阶导数来检测图像中的边缘, 核使用了 Laplace 算子

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y).$$

- **钝化掩蔽和高提升滤波:** 钝化掩蔽是一种通过减少图像的低频成分 (平滑部分) 来增强图像细节的方法, 它由如下步骤组成: 模糊原图像; 从原图像减去模糊后的图像 (产生的差称为模板); 将模板与原图像相加. 令 $\bar{f}(x,y)$ 表示模糊后的图像, 公式形式的模板为

$$g_{\text{mask}}(x,y) = f(x,y) - \bar{f}(x,y),$$

将加权后的模板与原图像相加:

$$g(x,y) = f(x,y) + k g_{\text{mask}}(x,y).$$

$k=1$ 时, 它是钝化掩蔽. $k>1$ 时, 这个过程称为高提升滤波.

- **Roberts 交叉梯度算子:** 通过计算图像在对角方向上的梯度来检测边缘, 它使用两个 2×2 的卷积核, 一个检测对角方向的梯度, 一个检测反对角方向的梯度. 具体地, 对于一个 3×3 的图像区域, 设其从左往右, 从上往下数的位置像素值为 $z_i, i=1,\cdots,9$(如中心点为 z_5, 其右边为 z_6), Roberts 算子为

$$g_x = z_9 - z_5, \quad g_y = z_8 - z_6.$$

则梯度图像表示为

$$M(x,y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]^{\frac{1}{2}}.$$

- **Sobel 算子:** 通过计算图像在水平方向和垂直方向上的一阶导数来检测边缘. 它比 Roberts 交叉梯度算子更复杂, 但对噪声有更好的抑制能力. 记号同上, Sobel 算子为

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3), g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7).$$

梯度图像表示为

$$M(x,y) = \left[((z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3))^2 + ((z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7))^2\right]^{\frac{1}{2}}.$$

通常, 给定的任务需要用到几种互补技术才能得到可以令人接受的结果.

Chapter 3

频率域滤波