

# Computer Vision

Tianlin Pan

October 28, 2025

# Contents

1 数字图像基础 .....	3
1.1 数字图像的表示 .....	3
1.2 图像分辨率 .....	3
1.3 视觉动态范围 .....	3
1.4 数字图像的基本操作 .....	3
2 图像变换 .....	4
2.1 空间域变换 .....	4
2.2 频域变换 .....	5
3 图像滤波与数字滤波器 .....	6
3.1 图像卷积与相关 .....	6
3.2 经典数字滤波器 .....	6
4 形态学与基于深度学习的图像去噪 .....	9
4.1 结构元素 .....	9
4.2 二值图像的膨胀 .....	9
4.3 二值图像的腐蚀 .....	9
4.4 开运算与闭运算 .....	10
5 图像增强 .....	11
5.1 基于空域的图像增强 .....	11
5.2 基于频域的图像增强 .....	13
6 图像退化和复原 .....	15
6.1 图像退化 .....	15
6.2 图像复原 .....	16
6.3 无约束图像复原 .....	17
6.4 有约束图像复原 .....	18
7 图像压缩 .....	19
7.1 图像压缩背景 .....	19
7.2 无损压缩方法 .....	20
7.3 有损压缩方法 .....	21
8 图像物体表达 .....	23
8.1 面向识别任务的物体表达 .....	23
8.2 神经场景表达 .....	24

# 1 数字图像基础

## 1.1 数字图像的表达

图像是一个二维函数  $f(x, y)$ , 其幅值称为 **强度** 或 **灰度**. 数字图像是由有限数量的元素组成的, 每个元素具有离散的数值. 这些元素称为 **像素** (picture element). 数字图像可以表示为一个二维矩阵, 其中每个元素对应于图像中的一个像素.

## 1.2 图像分辨率

- 空间分辨率 (PPI): 每英寸像素数, 描述图像的细节程度. 比如说, 300 PPI 意味着每英寸有 300 个像素, 2 英寸乘 2 英寸的图像将包含 600 乘 600 个像素.
- 设备分辨率 (DPI): 每英寸点数, 描述打印机或显示器的输出质量. 比如说, 600 DPI 意味着每英寸可以打印或显示 600 个独立的点.

## 1.3 视觉动态范围

视觉动态范围是指人眼能够感知的亮度范围. 人眼可以适应非常宽的亮度范围, 从非常暗的环境 (如月光下) 到非常亮的环境 (如阳光直射). 典型情况下, 人眼可以感知的亮度范围约为  $10^6 : 1$ .

**高动态范围图像的合成:** 通过拍摄同一场景的多张不同曝光的照片, 然后将它们合成为一张高动态范围图像. 这种方法可以捕捉到更多的细节, 特别是在亮部和暗部.

## 1.4 数字图像的基本操作

### 1.4.1 点运算

点运算是指对图像中的每个像素独立进行操作. 常见的点运算包括:

- 亮度调整: 通过增加或减少像素值来调整图像的亮度.
- 对比度调整: 通过拉伸或压缩像素值的范围来调整图像的对比度.

### 1.4.2 代数运算

代数运算是指对两幅图像的对应像素进行操作. 常见的代数运算包括加法、乘法和减法. 这些操作可以用于图像的融合、差异检测等.

### 1.4.3 逻辑运算

逻辑运算是指对图像的每个像素进行逻辑运算. 常见的逻辑运算包括与、或、非等. 这些操作可以用于图像的分割和特征提取.

## 2 图像变换

### 2.1 空间域变换

#### 2.1.1 齐次坐标

齐次坐标是对笛卡尔坐标的一种扩展, 允许我们使用矩阵运算来表示各种几何变换. 对于二维空间中的点  $(x, y)$ , 其齐次坐标表示为  $(x, y, w)$ , 其中  $w$  是一个非零的缩放因子. 通常情况下, 我们可以将  $w$  设为 1, 这样点  $(x, y)$  在齐次坐标中表示为  $(x, y, 1)$

#### 2.1.2 欧式变换

欧式变换包括平移和旋转, 它们保持距离和角度不变. 在二维空间中, 欧式变换可以表示为一个  $3 \times 3$  矩阵, 其形式如下:

$$\begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

它的自由度为 3, 包括一个旋转角度  $\theta$  和两个平移参数  $t_x, t_y$ .

#### 2.1.3 相似变换

相似变换包括欧式变换和缩放, 它们保持形状但不保持大小. 在二维空间中, 相似变换可以表示为一个  $3 \times 3$  矩阵, 其形式如下:

$$\begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

它的自由度为 4, 包括一个缩放因子  $s$ , 一个旋转角度  $\theta$  和两个平移参数  $t_x, t_y$ .

#### 2.1.4 仿射变换

仿射变换包括相似变换和剪切, 它们保持直线和平行关系. 在二维空间中, 仿射变换可以表示为一个  $3 \times 3$  矩阵, 其形式如下:

$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

它的自由度为 6, 包括四个线性变换参数  $a_{11}, a_{12}, a_{21}, a_{22}$  和两个平移参数  $t_x, t_y$ .

### 2.1.5 投影变换

投影变换包括仿射变换和透视效果, 它们可以改变直线和平行关系. 在二维空间中, 投影变换可以表示为一个  $3 \times 3$  矩阵, 其形式如下:

$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ a_{31} & a_{32} & 1 \end{bmatrix} \quad (4)$$

它的自由度为 8, 包括六个线性变换参数和两个平移参数.

### 2.1.6 3D 空间中的变换的自由度

- 欧式变换: 6 (3 个平移参数 + 3 个旋转参数)
- 相似变换: 7 (6 个欧式变换参数 + 1 个缩放参数)
- 仿射变换: 12 ( $4 \times 4$  的矩阵, 但最后一行通常为  $[0, 0, 0, 1]$ , 因此有 12 个自由度)
- 投影变换: 15 ( $4 \times 4$  的矩阵, 但最后一行通常为  $[h_{\{41\}}, h_{\{42\}}, h_{\{43\}}, 1]$ , 因此有 15 个自由度)

### 2.1.7 灰度变换

灰度变换是指对图像的灰度值进行非线性变换, 以增强图像的对比度或亮度 (点变换). 常见的灰度变换包括:

- 线性变换:  $s = ar + b$ , 其中  $a$  和  $b$  是常数. 如对比度, 亮度调整.
- 对数变换:  $s = c \log(1 + r)$ , 其中  $c$  是常数. 适用于增强暗部细节.
- 伽马变换:  $s = cr^\gamma$ , 其中  $c$  和  $\gamma$  是常数. 适用于调整图像的整体亮度.

## 2.2 频域变换

用于分析图像中的频率成分, 将图像表示为不同频率和振幅的模式之和.

### 2.2.1 连续 Fourier 变换

连续 Fourier 变换将空间域中的图像  $f(x, y)$  转换为频域中的表示  $F(u, v)$ :

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (5)$$

### 2.2.2 离散 Fourier 变换 (DFT)

离散 Fourier 变换将离散的图像  $f[m, n]$  转换为频域中的表示  $F[k, l]$ :

$$F[k, l] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m, n] e^{-j2\pi(k\frac{m}{M}+l\frac{n}{N})} \quad (6)$$

## 3 图像滤波与数字滤波器

### 3.1 图像卷积与相关

#### 3.1.1 定义

对于二维离散信号 (图像)  $f(m, n)$  和  $h(m, n)$ , 它们的相关 (correlation) 和卷积 (convolution) 定义如下:

- 相关:

$$g(m, n) = \sum_{j=-a}^a \sum_{k=-b}^b f(m+j, n+k)h(j, k) \quad (7)$$

- 卷积:

$$g(m, n) = \sum_{j=-a}^a \sum_{k=-b}^b f(m-j, n-k)h(j, k) \quad (8)$$

#### 3.1.2 相关和卷积的关系

相关和卷积之间的关系可以通过翻转滤波器  $h(m, n)$  来表示. 具体来说, 卷积可以看作是相关操作, 但滤波器被翻转了 180 度:

#### 3.1.3 卷积运算的性质

- 交换律:  $f * h = h * f$
- 结合律:  $f * (h * g) = (f * h) * g$
- 分配律:  $f * (h + g) = f * h + f * g$
- 恒等元:  $f * \delta = f$ , 其中  $\delta$  是单位脉冲函数.
- 微分性:  $\frac{\partial}{\partial x}(f * h) = \frac{\partial f}{\partial x} * h = f * \frac{\partial h}{\partial x}$

#### 3.1.4 卷积定理

卷积定理指出, 空间域中的卷积对应于频域中的乘法. 具体来说, 如果  $g(m, n) = f(m, n) * h(m, n)$ , 则其 Fourier 变换满足:

$$G(u, v) = F(u, v)H(u, v) \quad (9)$$

## 3.2 经典数字滤波器

### 3.2.1 滤波和滤波器

- 滤波: 通过某种操作来改变图像的频率成分, 以达到增强或抑制某些特征的目的.

- 滤波器: 用于执行滤波操作的工具或算法, 可以是空间域的 (如卷积核) 或频域的 (如频率响应).

### 3.2.2 图像噪声

1. 高斯噪声: 每一个像素的噪声值 i.i.d. 服从高斯分布, 常见于电子设备产生的噪声.
2. 椒盐噪声: 噪声值随机地取最大值 (灰度值 = 255, 盐噪声) 或最小值 (灰度值 = 0, 椒噪声), 常见于传输错误或故障.
3. 泊松噪声: 噪声值服从泊松分布, 常见于光子计数过程, 如低光照条件下的图像采集.

### 3.2.3 高斯滤波

高斯滤波是一种线性平滑滤波器, 其卷积核由高斯函数定义. 高斯滤波器的形式如下:

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (10)$$

在离散形式中, 高斯滤波器可以表示为一个  $m \times n$  的矩阵, 其中每个元素由高斯函数计算得到. 高斯滤波器的标准差  $\sigma$  控制了滤波器的平滑程度, 较大的  $\sigma$  会导致更强的平滑效果.

高斯滤波可以用于图像降噪, 但是在降低噪声的同时, 也会使得图像变得模糊, 特别是边缘部分.

### 3.2.4 双边滤波

双边滤波在高斯滤波的基础上加入像素值权重项, 既关注像素的位置信息, 同时也考虑像素的灰度 (颜色) 信息. 在像素值权重项中, 像素灰度 (颜色) 越相近, 则权重越大.

考虑像素  $q$  的邻域  $S$ , 双边滤波的计算公式如下:

$$I'(q) = \frac{1}{W} \sum_{p \in S} I(p) f(\|p - q\|) g(|I(p) - I(q)|) \quad (11)$$

其中

1.  $f(\|p - q\|)$  是 **空间权重函数**, 通常采用高斯函数, 用于衡量像素  $p$  与像素  $q$  之间的空间距离.
2.  $g(|I(p) - I(q)|)$  是 **像素值权重函数**, 通常也采用高斯函数, 用于衡量像素  $p$  与像素  $q$  之间的灰度差异.

3.  $W$  是归一化因子, 用于确保权重和为 1.

双边滤波同时关注像素的位置信息和颜色信息, 在滤除噪声的同时, 保留了图像边缘.

### 3.2.5 Wiener 滤波

图像采集过程中造成图像退化, 退化模型为

$$g(x, y) = h(x, y) * f(x, y) + n(x, y) \quad (12)$$

这里的  $h(x, y)$  是退化函数,  $n(x, y)$  是噪声.

图像的空间域滤波主要针对加性噪声, 无法处理图像退化. Wiener 滤波的目标是通过已知的退化函数  $h(x, y)$  和噪声统计特性, 来估计原始图像  $f(x, y)$ . 它通过在频域中对图像进行滤波, 以最小化恢复图像与原始图像之间的均方误差. Wiener 滤波器的频率响应可以表示为:

$$H_{w(u,v)} = \frac{|H(u, v)|^2}{|H(u, v)|^2 + \left(\frac{S_{n(u,v)}}{S_{f(u,v)}}\right)} * \left(\frac{1}{H(u, v)}\right) \quad (13)$$

其中  $H(u, v)$  是退化函数的频率响应,  $S_{n(u,v)}$  是噪声的功率谱密度,  $S_{f(u,v)}$  是原始图像的功率谱密度, 对应的比例系数  $K(u, v) = \frac{S_{n(u,v)}}{S_{f(u,v)}}$  就是信号与噪声的比值 (SNR).



## 4 形态学与基于深度学习的图像去噪

### 4.1 结构元素

结构元素 (structuring element) 是一个小的二值矩阵, 用于定义形态学操作的邻域. 结构元素通常具有对称的形状, 如方形、圆形或十字形. 结构元素的大小和形状会影响形态学操作的结果.

### 4.2 二值图像的膨胀

膨胀操作用于扩展图像中的前景区域 (通常为白色像素). 其基本思想是将结构元素在图像上滑动, 如果结构元素与图像的某个区域有重叠, 则将该区域的中心像素设为前景 (白色).

若原图像为  $A$ , 结构元素为  $B$ , 则膨胀操作定义为:

$$A \oplus B = \left\{ x \mid (\hat{B})_x \cup A \neq \emptyset \right\} \quad (14)$$

其中  $\hat{B}$  是结构元素  $B$  的反射 (i.e. 旋转 180 度),  $B_x$  是结构元素  $B$  的平移.

用算法流程表示就是

1. 使用反射结构元素扫描图像中的每一个像素
2. 将反射结构元素与其覆盖的二值图像进行逻辑与操作
3. 如果覆盖区域内的运算结果都为 0, 则将该像素设为 0; 否则设为 1.

**膨胀的应用:** 填补图像中的小孔洞, 连接断开的前景区域.

### 4.3 二值图像的腐蚀

腐蚀操作用于缩小图像中的前景区域. 其基本思想是将结构元素在图像上滑动, 如果结构元素完全包含在图像的某个区域内, 则将该区域的中心像素设为前景 (白色); 否则设为背景 (黑色).

若原图像为  $A$ , 结构元素为  $B$ , 则腐蚀操作定义为:

$$A \ominus B = \{ x \mid (B)_x \subseteq A \} \quad (15)$$

用算法流程表示就是

1. 使用结构元素扫描图像中的每一个像素
2. 将结构元素与其覆盖的二值图像进行逻辑与操作
3. 如果覆盖区域内的运算结果都为 1, 则将该像素设为 1; 否则设为 0.

## 4.4 开运算与闭运算

- **开运算 (Opening):** 先进行腐蚀操作, 然后进行膨胀操作. 开运算可以去除小的前景噪声, 同时保持较大的前景区域的形状.

$$A \circ B = (A \ominus B) \oplus B \quad (16)$$

- **闭运算 (Closing):** 先进行膨胀操作, 然后进行腐蚀操作. 闭运算可以填补小的前景孔洞, 同时保持较大的前景区域的形状.

$$A \bullet B = (A \oplus B) \ominus B \quad (17)$$

## 5 图像增强

### 5.1 基于空域的图像增强

#### 5.1.1 灰度直方图

灰度直方图是图像中各个灰度级别的像素数量的统计. 它可以帮助我们了解图像的对比度、亮度和动态范围. 高对比度的图像通常具有较宽的灰度直方图, 而低对比度的图像则具有较窄的灰度直方图.

#### 5.1.2 直方图均衡化

直方图均衡化是一种增强图像对比度的方法. 其基本思想是通过重新分配图像的灰度级别, 使得灰度直方图更加均匀分布. 直方图均衡化可以增强图像的细节, 特别是在亮度范围较窄的图像中.

直方图均衡化的步骤如下:

1. 统计各灰度级的像素数目  $n_i, i = 0, 1, \dots, L - 1$ .
2. 计算原始图像直方图各灰度级的频数  $p_i = \frac{n_i}{M \times N}$ , 其中  $M \times N$  是图像的总像素数.
3. 计算累积分布函数 (CDF):  $CDF(i) = \sum_{j=0}^i p_j, i = 0, 1, \dots, L - 1$ .
4. 计算映射函数:  $s_k = (L - 1) CDF(r_k)$ , 其中  $r_k$  是原始图像的灰度级,  $s_k$  是均衡化后的灰度级.
5. 使用映射函数将原始图像的灰度级转换为均衡化后的灰度级.

#### 5.1.3 局部直方图均衡化

前述直方图均衡化是基于全图的, 适用于整体增强, 但当目的是增强图像中几个小区域的细节时, 通常就会失败. 这是因为在这些小区域中, 像素的数量对计算全局变换的影响可以忽略.

局部直方图均衡化 (LHE) 是一种基于局部区域的图像增强方法. 其基本思想是对图像的每个像素, 使用其邻域内的像素来计算局部直方图, 然后应用直方图均衡化. 这样可以增强图像中的局部细节, 特别是在亮度变化较大的区域.

#### 5.1.4 直方图匹配 (规定化)

直方图匹配是一种将图像的灰度分布调整为指定分布的方法. 其基本思想是通过计算原始图像和目标分布的累积分布函数 (CDF), 然后使用这些 CDF 来重新映射图像的灰度级别.

直方图匹配的步骤如下:

1. 计算原始图像和目标图像的灰度直方图和累积分布函数  $CDF_{\text{source}}(i)$  和  $CDF_{\text{target}}(i)$ .
2. 对于原始图像的每个灰度级  $r_k$ , 找到目标图像中使得  $CDF_{\text{target}}(s_k) \approx CDF_{\text{source}}(r_k)$  的灰度级  $s_k$ .
3. 使用映射函数将原始图像的灰度级转换为目标图像的灰度级.

### 5.1.5 空间域滤波器

**线性滤波器:** 通过卷积操作实现, 可以用于平滑 (如均值滤波) 或锐化 (如拉普拉斯滤波)

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (18)$$

这里的  $w(s, t)$  是滤波器的权重矩阵 (卷积核).

**非线性滤波器:** 不依赖于线性卷积, 可以更有效地处理某些类型的噪声 (如中值滤波)

### 5.1.6 空间域平滑

**均值滤波器:** 通过计算邻域内像素的平均值来平滑图像, 可以有效地减少高斯噪声, 但会模糊图像细节.

$$g(x, y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^a \sum_{t=-b}^b f(x + s, y + t) \quad (19)$$

**超限像素平滑法:** 通过识别和替换异常像素来平滑图像, 可以有效地去除椒盐噪声, 同时保留图像细节.

$$g'(x, y) = \begin{cases} g(x, y) & \text{if } |f(x, y) - g(x, y)| < T \\ f(x, y) & \text{otherwise} \end{cases} \quad (20)$$

其中  $f(x, y)$  是原始图像,  $g(x, y)$  是均值滤波后的图像,  $T$  是阈值.

**中值滤波器:** 通过计算邻域内像素的中值来平滑图像, 可以有效地去除椒盐噪声, 同时保留图像边缘.

$$g(x, y) = \text{median}\{f(x + s, y + t), -a \leq s \leq a, -b \leq t \leq b\} \quad (21)$$

### 5.1.7 空间域锐化

图像锐化指的是对图像的边缘或轮廓进行增强, 以达到增强视觉效果的目的. 由于边缘通常出现在灰度突变的地方, 所以锐化的实现通常依赖于图像的梯度信息.

### 5.1.8 梯度锐化法

在离散图像  $f(x, y)$  中, 其梯度可以通过以下方式近似计算:

- $G_x = f(x + 1, y) - f(x, y)$
- $G_y = f(x, y + 1) - f(x, y)$

它们可以分别对应两个卷积核:

$$G_x = [1, -1], G_y = [1; -1]^T \quad (22)$$

它们被称为 **梯度算子** (gradient operator). 除梯度算子外, 还有很多其他的算子, 如 Roberts 算子

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (23)$$

为了在锐化边缘的同时减少噪声的影响, Prewitt 加大了边缘增强算子的模板大小, 其算子为

$$G_{\text{hori}} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, G_{\text{vert}} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, G_{\text{diag}} = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad (24)$$

## 5.2 基于频域的图像增强

图像的低频分量, 对应到图像灰度变化平缓的部分; 图像的高频分量, 对应到灰度变化剧烈的区域, 例如细节噪声和边缘. 因此, 通过滤波器来增强或抑制图像的某些频率成分, 可以达到图像增强的目的.

### 5.2.1 理想低通滤波器 ILPF

理想低通滤波器 (Ideal Low Pass Filter, ILPF) 在频域中定义为:

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases} \quad (25)$$

其中  $D(u, v) = \sqrt{(u - u_0)^2 + (v - v_0)^2}$  是频率点  $(u, v)$  到频率中心  $(u_0, v_0)$  的距离,  $D_0$  是截止频率. ILPF 会完全保留低于截止频率的成分, 并完全抑制高于截止频率的成分.

### 5.2.2 Butterworth 低通滤波器 BLPF

Butterworth 低通滤波器 (Butterworth Low Pass Filter, BLPF) 在频域中定义为:

$$H(u, v) = \frac{1}{1 + \left( \frac{D(u, v)}{D_0} \right)^{2n}} \quad (26)$$

其中  $n$  是滤波器的阶数, 控制滤波器的陡峭程度. BLPF 提供了一个平滑的过渡, 相比 ILPF 更少引入振铃效应.

## 6 图像退化和复原

### 6.1 图像退化

#### 6.1.1 图像退化类型

- **规则图案变形**: 由已知的几何变换引起的图像变形, 如旋转、缩放和平移.
- **边缘/运动模糊**: 由于成像系统的点扩散函数 (PSF) 引起的图像模糊, 如运动模糊和散焦模糊.
- **噪声污染**: 由传感器噪声或传输错误引起的图像噪声, 如高斯噪声和椒盐噪声.

#### 6.1.2 图像退化的数学模型

$$g(x, y) = h(x, y) * f(x, y) + n(x, y) \quad (27)$$

在频域中表示为:

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad (28)$$

这里的  $g(x, y)$  是退化后的图像,  $f(x, y)$  是原始图像,  $h(x, y)$  是退化函数,  $n(x, y)$  是噪声.

常见的噪声类型有

- **高斯噪声**: 服从高斯分布的随机噪声, 常见于电子设备产生的噪声.
- **椒盐噪声**: 噪声值随机地取最大值 (灰度值 = 255, 盐噪声) 或最小值 (灰度值 = 0, 椒噪声), 常见于传输错误或故障.
- **瑞利噪声**: 服从瑞利分布的随机噪声, 常见于雷达成像等应用.
- **伽马噪声**: 服从伽马分布的随机噪声, 常见于医学成像等应用.
- **均匀噪声**: 在一定范围内均匀分布的随机噪声.

#### 6.1.3 LTI 系统近似

在图像退化模型中, 退化函数  $h(x, y)$  通常被假设为线性时不变系统 (LTI), 或者为一系列 LTI 系统的组合. 这种假设简化了图像复原问题, 使得我们可以使用卷积和频域分析来处理图像退化.

#### 6.1.4 点扩散函数 (PSF)

点扩散函数 (Point Spread Function, PSF) 描述了成像系统对一个点光源的响应.

$$h(x, \alpha, y, \beta) = H \cdot \delta(x - \alpha, y - \beta) \quad (29)$$

其中  $H$  是系统的增益,  $\delta(x - \alpha, y - \beta)$  是二维单位冲激函数, 表示点光源在位置  $(\alpha, \beta)$  处的响应. 在这种情况下, 退化的数学模型可以表示为:

$$g(x, y) = \iint_{-\infty}^{+\infty} f(\alpha, \beta) h(x, \alpha, y, \beta) d\alpha d\beta + n(x, y) \quad (30)$$

### 6.1.5 离散系统的退化模型

在离散图像处理中, 退化模型可以表示为:

$$g(m, n) = \sum_{s=-a}^a \sum_{t=-b}^b h(s, t) f(m - s, n - t) + n(m, n) \quad (31)$$

这里的  $g(m, n)$  是退化后的图像,  $f(m, n)$  是原始图像,  $h(s, t)$  是离散的退化函数,  $n(m, n)$  是离散的噪声. 它可以被写作矩阵形式:

$$[g] = [H][f] + [n] \quad (32)$$

展开来就是

$$\begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(MN - 1) \end{bmatrix} = \begin{bmatrix} H_0 & H_{M-1} & \dots & H_1 \\ H_1 & H_0 & \dots & H_2 \\ \vdots & \vdots & \ddots & \vdots \\ H_{M-1} & H_{M-2} & \dots & H_0 \end{bmatrix} \times \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(MN - 1) \end{bmatrix} + \begin{bmatrix} n(0) \\ n(1) \\ \vdots \\ n(MN - 1) \end{bmatrix} \quad (33)$$

其中  $H_i$  被定义为

$$H_i = \begin{bmatrix} h_e(i, 0) & h_e(i, N - 1) & \dots & h_e(i, 1) \\ h_e(i, 1) & h_e(i, 0) & \dots & h_e(i, 2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(i, N - 1) & h_e(i, N - 2) & \dots & h_e(i, 0) \end{bmatrix} \quad (34)$$

## 6.2 图像复原

图像复原是指通过已知的退化模型和噪声统计特性, 来估计原始图像.



## 图像复原和图像增强的区别

**图像增强** 不考虑图像降质的原因, 只将图像中感兴趣的特征有选择地突出, 从而衰减不需要的特征. 改善后的图像不一定要去逼近原图像; **图像复原** 则是试图去逆转已知的图像降质过程, 以恢复原始图像. 图像复原通常需要对降质过程有一个明确的数学模型, 要建立评价复原好坏的客观标准.

简而言之, 图像增强侧重于改善图像的视觉效果, 而图像复原侧重于恢复初始图像. 对一幅已经退化的图像, 通常的做法是先做图像复原, 然后再做图像增强.

## 6.3 无约束图像复原

### 6.3.1 代数复原方法

假设图像退化模型为

$$n = g - Hf \quad (35)$$

我们希望找到一个  $f$  的估计, 使得误差  $n$  最小. 这可以通过最小化以下目标函数来实现:

$$J(f) = \|g - Hf\|^2 \quad (36)$$

此处的  $J$  不依赖于任何先验知识, 因此称为无约束复原方法. 通过对  $J(f)$  求导并设导数为零, 可以得到最小化  $J(f)$  的解:

$$f = (H^T H)^{-1} H^T g = H^{-1} g \quad (37)$$

### 6.3.2 逆滤波

逆滤波是一种基于频域的图像复原方法. 假设图像退化模型为

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad (38)$$

逆滤波的目标是通过除以退化函数  $H(u, v)$  来恢复原始图像  $F(u, v)$ :

$$F^{u,v} = \frac{1}{H(u, v)} [G(u, v) - N(u, v)] \quad (39)$$

然而, 逆滤波对噪声非常敏感, 特别是在  $H(u, v)$  接近零的频率处. 因此, 逆滤波通常只适用于噪声较小的情况.

## 6.4 有约束图像复原

无约束复原方法中的逆滤波虽然比较简单, 但并没有说明如何处理噪声. 而有约束复原方法则考虑了噪声的影响, 并通过引入先验知识来改善复原效果.

### 6.4.1 代数有约束复原方法

在有约束复原方法中, 我们希望计算函数  $\|Qf\|$  在约束条件  $\|g - Hf\|^2 = \|n\|^2$  下的最小值, 其中  $Q$  是一个正则化矩阵, 用于引入先验知识. 通过拉格朗日乘数法, 可以将其转化为无约束优化问题:

$$J(f) = \|Qf\|^2 + \lambda(\|g - Hf\|^2 - \|n\|^2) \quad (40)$$

可以通过对  $J(f)$  求导并设导数为零, 得到最小化  $J(f)$  的解:

$$f = \left( H^T H + \frac{1}{\lambda} Q^T Q \right)^{-1} H^T g \quad (41)$$

我们也可以设置别的约束条件, 如  $\|g\|^2 = \|f\|^2$ .

## 7 图像压缩

### 7.1 图像压缩背景

#### 7.1.1 图像压缩合理性

- 一般原始图像中存在很大的冗余度
- 用户通常允许图像失真
- 当信道的分辨率不及原始图像的分辨率时, 降低输入的原始图像的分辨率对输出图像分辨率影响不大
- 用户对原始图像的信号不全都感兴趣, 可用特征提取和图像识别的方法, 丢掉大量无用的信息, 提取有用的信息.

#### 7.1.2 图像冗余

**编码冗余** 如果一个图像的灰度级编码, 使用的比特数超过了该图像实际所需的最少比特数, 则称该图像存在编码冗余. 例如, 使用 8 比特来表示灰度级别的图像, 但实际上只使用了 100 个灰度级别, 则存在编码冗余.

**像素冗余** 对于一幅图像, 很多单个像素对视觉的贡献是冗余的. 原始图像越有规则, 各像素之间的相关性越强, 像素冗余就越大. 例如, 在一幅平坦区域的图像中, 相邻像素的灰度值通常非常接近, 因此存在较大的像素冗余.

**心理视觉冗余** 人眼对不同频率和颜色的敏感度不同. 例如, 人眼对高频细节和某些颜色变化不敏感, 因此可以通过去除这些不敏感的信息来实现图像压缩, 这就是心理视觉冗余.

#### 7.1.3 无损压缩和有损压缩

无损压缩是指在压缩过程中不丢失任何信息, 压缩后的图像可以完全恢复为原始图像. 常见的无损压缩方法格式有: PNG, GIF, BMP 等.

有损压缩是指在压缩过程中丢失部分信息, 压缩后的图像无法完全恢复为原始图像, 但通常可以达到视觉上不可察觉的效果. 常见的有损压缩方法格式有: JPEG, WebP 等.

#### 7.1.4 图像压缩的保真度准则

图像信号在编码和传输过程中会产生误差, 尤其是在有损压缩编码中, 产生的误差应在允许的范围之内. 在这种情况下, 保真度准则可以用来衡量编码方法或系统质量的优劣.

##### 客观保真度准则

- 均方根误差 (RMSE): 衡量原始图像和压缩图像之间的平均误差

$$\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - \hat{f}(x, y)]^2} \quad (42)$$

- 均方信噪比 (SNR): 衡量信号强度与噪声强度的比值

$$\text{SNR} = \left( \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y)]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - \hat{f}(x, y)]^2} \right) \quad (43)$$

信噪比越大, 表示压缩图像质量越好.

**主观保真度准则** 通过人眼对图像的主观评价来衡量图像质量, 例如对比度、清晰度和色彩还原度等.

## 7.2 无损压缩方法

### 7.2.1 压缩率

压缩率 (Compression Ratio, CR) 是衡量图像压缩效果的一个重要指标, 定义为原始图像大小与压缩后图像大小之比:

$$\text{CR} = \frac{\text{Size}_{\text{original}}}{\text{Size}_{\text{compressed}}} \quad (44)$$

无损压缩的压缩率一般较低, 通常在 2 : 1 到 10 : 1 之间, 具体取决于图像的内容和所使用的压缩算法.

### 7.2.2 变长编码: Huffman 编码

Huffman 编码是一种基于字符出现频率的无损压缩方法. 它通过构建一棵二叉树, 将频率较高的字符分配较短的编码, 频率较低的字符分配较长的编码, 从而实现压缩. Huffman 编码的步骤如下:

1. 统计图像中各灰度级的出现频率.
2. 根据频率构建 Huffman 树.
3. 为每个灰度级分配二进制编码.
4. 使用分配的编码对图像进行编码.

Huffman 编码是一种 **最优变长编码**, 在已知符号概率分布的情况下, 可以实现最小的平均编码长度; 也是一种 **快码**, 因为各个信源符号都被解成一组固定次序的码符号; 同时也是一种 **前缀码 (即时码)**, 即任何一个码字都不是另一个码字的前缀, 这样可以确保编码的唯一可解析性, 从而提升解码效率.

### 7.2.3 变长编码: 算数编码

算数编码是一种基于概率模型的无损压缩方法. 它通过将整个消息映射到一个实数区间  $[0, 1)$  上, 并根据符号的概率分布不断缩小该区间, 最终生成一个实数作为编码结果. 算数编码的步骤如下:

1. 统计各字符的出现频率, 并计算其概率分布.
2. 按照各字符的概率分布, 将区间  $[0, 1)$  划分为若干子区间. 比如说, 有三个符号 A, B, C, 其概率分别为 0.5, 0.3, 0.2, 则可以划分为  $[0, 0.5)$ ,  $[0.5, 0.8)$ ,  $[0.8, 1)$ .
3. 对于输入消息, 逐个符号处理, 不断缩小区间. 例如, 对于消息“AB”, 首先处理 A, 将区间缩小到  $[0, 0.5)$ ; 然后处理 B, 将区间缩小到  $[0.25, 0.4)$ .
4. 最终选择区间内的任意一个二进制表示最短的实数作为编码结果.

### 7.2.4 LZW 编码

LZW (Lempel-Ziv-Welch) 编码是一种基于字典的无损压缩方法. 它通过动态构建一个字典, 将输入数据中的重复模式替换为较短的代码, 从而实现压缩. 在编码处理的开始阶段, LZW 编码器初始化一个包含所有可能单字符模式的字典. 对于 8 位单色图像, 字典中前 256 个字被分配给灰度值 0 ~ 255. 当编码器顺序地分析图像像素的时候, 字典中没有包括的灰度级序列由算法决定其出现的位置

## 7.3 有损压缩方法

### 7.3.1 变换编码

将原来在空域描述的图像信号, 通过某种数学变换转换到另外一些正交空间中去, 用变换系数来表示图像信号, 这种方法称为变换编码. 图像变换会使图像信号能量在空间重新分布, 其中低频成分占据能量绝大部分, 而高频成分只占有很小一部分能量.

一般来说在变换域里描述比在空域简单, 因为图像相关性明显下降. 尽管变换本身并不带来数据压缩, 但是由于变换图像的能量大部分只集中于少数几个变换系数上, 因此可以通过对变换系数进行量化和编码来实现图像压缩.

**DFT 编码:** 离散傅里叶变换 (Discrete Fourier Transform, DFT) 将图像从空间域转换到频域. DFT 编码通过保留低频成分并丢弃高频成分来实现压缩. 然而, DFT 编码存在计算复杂度高和阻塞效应明显的问题.

**DCT 编码:** 离散余弦变换 (Discrete Cosine Transform, DCT) 是一种常用的图像变换方法. DCT 编码通过将图像划分为小块 (通常为  $8 \times 8$ ), 对每个块进行 DCT 变换

$$C(u, v) = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \left[ (2x+1)u \frac{\pi}{16} \right] \cos \left[ (2y+1)v \frac{\pi}{16} \right]$$

然后对变换系数进行量化和编码来实现压缩. DCT 编码具有较低的计算复杂度和较好的压缩性能, 广泛应用于 JPEG 图像压缩标准中.

## 8 图像物体表达

### 8.1 面向识别任务的物体表达

- **传统识别任务:** ImageNet 图像分类, 如 AlexNet, VGG, ResNet 等卷积神经网络 (CNN) 架构.
- **人脸识别任务:** DeepID, VS2VI
- **小样本/零样本识别任务:** 给定少量 (或没有, 比如说通过训练几个品种的鸟类数据, 让模型能识别去其他种类的鸟) 的有标注数据, 学习一个可以识别这些类别数据的模型.

#### 8.1.1 零样本识别方法类型

- **基于表征学习的零样本学习方法:** 直接从图像中学习特征表示, 然后将这些特征映射到语义空间中进行分类.
- **基于生成模型的零样本学习方法:** 使用生成模型 (如生成对抗网络, GAN), 先用可见类别的数据训练生成模型, 然后生成不可见类别的样本, 最后使用这些生成的样本进行分类器训练.

#### 8.1.2 GAN

生成对抗网络 (Generative Adversarial Network, GAN) 由两个神经网络组成: 生成器 (Generator) 和判别器 (Discriminator). 生成器负责生成逼真的图像, 判别器负责区分真实图像和生成图像. 通过对抗训练, 生成器不断改进其生成能力, 最终能够生成高质量的图像.

GAN 的训练过程可以表示为一个最小化最大化的问题:

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (46)$$

其中,  $G$  是生成器,  $D$  是判别器,  $p_{\text{data}}(x)$  是真实数据的分布,  $p_z(z)$  是噪声的分布. 对于判别器  $D$  而言, 它的目标是最大化正确分类真实图像和生成图像的概率; 对于生成器  $G$  而言, 它的目标是最小化判别器对生成图像的分类错误率.

**CGAN (Conditional GAN):** 在生成器和判别器中引入条件变量 (如类别标签), 使得生成的图像能够符合特定的条件.

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x|y)] + E_{z \sim p_z(z)} [\log(1 - D(G(z|y)))] \quad (47)$$



### 8.1.3 开集识别任务

开集识别 (Open Set Recognition, OSR) 任务旨在识别训练集中未见过的类别. 传统的闭集识别方法假设测试数据的类别与训练数据的类别完全相同, 而开集识别方法则允许测试数据包含未知类别.

**判别式方法:** 直接训练一个具有良好泛化性能的开集识别神经网络, 将原有的  $k$  分类问题转化为  $k + 1$  分类问题, 其中第  $k + 1$  类表示未知类别. 这可以通过引入一个拒绝选项来实现, 当模型对某个输入的预测概率低于某个阈值时, 将其归类为未知类别.

**生成式方法:** 对已知类别样本的分布进行建模, 根据测试样本是否符合建模的分布. 如果不符合, 则将其归类为未知类别. 这可以通过使用生成对抗网络 (GAN) 或变分自编码器 (VAE) 来实现, 以学习已知类别的特征分布.

## 8.2 神经场景表达

### 8.2.1 NeRF

神经辐射场 (Neural Radiance Fields, NeRF) 是一种用于表示和渲染复杂 3D 场景的神经网络方法. NeRF 使用一个多层感知机 (MLP) 来表示场景中的体积密度和颜色信息. 给定一个 3D 坐标  $(x, y, z)$  和一个视角方向  $(\theta, \varphi)$ , NeRF 输出该位置的颜色  $c$  和体积密度  $\sigma$ :

$$(c, \sigma) = \text{MLP}(x, y, z, \theta, \varphi) \quad (48)$$

通过对场景进行采样和体积渲染, NeRF 可以生成高质量的视图. 具体来说, 对于每个像素, NeRF 沿着视线采样多个点, 计算每个点的颜色和密度, 然后使用体积渲染公式将这些信息合成最终的像素颜色.