

《生物医学图像处理》复习笔记

知识点梳理

2025 年 6 月 19 日

目录

1	数字图像基础	9
1.1	采样 (Sampling)	9
1.1.1	从连续到离散	9
1.1.2	矩阵大小与空间分辨率	9
1.2	量化 (Quantization)	9
1.2.1	灰度级 (Gray Level)	10
1.2.2	位深度与图像质量	10
1.2.3	动态范围与存储	10
1.3	插值 (Interpolation)	10
1.3.1	为什么需要插值?	10
1.3.2	邻域 (Neighborhood)	11
1.3.3	常用插值方法	11
2	空间域运算	12
2.1	基本数学工具 (Basic Mathematical Tools)	12
2.1.1	算术运算 (Arithmetic Operations)	12
2.1.2	集合与逻辑运算 (Set and Logical Operations)	12
2.2	几何变换 (Geometric Transform)	13
2.2.1	仿射变换 (Affine Transformation)	13
2.2.2	非线性变换 (Non-linear Transform)	15
2.2.3	几何变换的应用: 图像配准 (Image Registration)	15
2.3	强度变换 (Intensity Transformations)	15
2.3.1	点处理 (Point Processing)	15
2.3.2	直方图的统计矩 (Statistical Moments)	17
2.3.3	直方图处理 (Histogram Processing)	17

3	空间域滤波	23
3.1	空间滤波基础 (Basics of Spatial Filtering)	23
3.1.1	核心操作：相关与卷积	23
3.1.2	空间域与频率域的关系	24
3.2	低通滤波器 (Low Pass Filter)	24
3.2.1	线性低通滤波器	24
3.2.2	非线性低通滤波器（排序统计滤波器）	25
3.2.3	池化层 (Pooling Layer)	26
3.3	高通滤波器 (High Pass Filter)	26
3.3.1	一阶微分与二阶微分	27
3.3.2	基于二阶微分的滤波器：拉普拉斯算子	27
3.3.3	基于一阶微分的滤波器：梯度算子	29
3.4	利用空间滤波进行图像增强	29
3.4.1	拉普拉斯增强	29
3.4.2	非锐化掩模 (Unsharp Masking)	30
3.4.3	组合空间增强	30
3.5	带通滤波器 (Band Pass Filter)	31
4	频率域运算	32
4.1	傅里叶变换基础 (Fourier Transform Fundamentals)	32
4.1.1	傅里叶级数与傅里叶变换	32
4.2	采样定理与混叠 (The Sampling Theorem and Aliasing)	32
4.2.1	采样的频率域效应	32
4.2.2	采样定理 (Nyquist Criterion)	33
4.2.3	混叠 (Aliasing)	33
4.3	二维离散傅里叶变换 (2D-DFT)	33
4.3.1	2D-DFT 变换对	33
4.3.2	图像频率域 (k 空间) 的理解	33
4.4	2D-DFT 的重要性质	34
4.4.1	可分离性 (Separability)	34
4.4.2	平移与中心化 (Translation & Centering)	34
4.4.3	共轭对称性 (Conjugate Symmetry)	34
4.4.4	卷积定理 (Convolution Theorem)	34
4.5	快速傅里叶变换 (Fast Fourier Transform - FFT)	35
4.6	小波变换简介 (Introduction to Wavelet Transform)	35
4.6.1	动机	35
4.6.2	核心思想	35
4.6.3	优势	35

4.6.4	离散小波变换 (DWT)	35
5	频率域滤波	36
5.1	频域滤波基础 (Fundamentals)	36
5.1.1	空间域与频率域的对应关系	36
5.1.2	频域滤波的基本流程	36
5.2	频域低通滤波器 (Low-Pass Filters)	37
5.2.1	理想低通滤波器 (Ideal Low-Pass Filter, ILPF)	37
5.2.2	巴特沃斯低通滤波器 (Butterworth Low-Pass Filter, BLPF)	39
5.2.3	高斯低通滤波器 (Gaussian Low-Pass Filter, GLPF)	40
5.3	频域高通滤波器 (High-Pass Filters)	42
5.3.1	基本关系	42
5.3.2	主要类型	42
5.3.3	频域拉普拉斯滤波器	43
5.3.4	高频增强滤波 (High-Frequency-Emphasis Filtering)	44
5.4	选择性滤波 (Selective Filtering)	45
5.4.1	带阻/带通滤波器 (Band-reject/Band-pass)	45
5.4.2	陷波滤波器 (Notch Filter)	46
6	图像复原	48
6.1	图像复原基础 (Image Restoration Fundamentals)	48
6.1.1	图像复原 vs. 图像增强	48
6.1.2	图像退化/复原模型	48
6.2	噪声模型 (Noise Models)	49
6.2.1	常见噪声类型	50
6.2.2	噪声参数的估计	52
6.3	仅有噪声的图像复原 (Spatial Filtering Restoration)	53
6.3.1	均值滤波器 (Mean Filters)	53
6.3.2	排序统计滤波器 (Order-Statistics Filters)	56
6.3.3	自适应滤波器 (Adaptive Filters)	59
6.3.4	非局部均值滤波器 (Non-local Means Filter)	60
6.4	图像反卷积 (Image Deconvolution)	60
6.4.1	估计退化函数 H	60
6.4.2	反滤波 (Inverse Filtering)	60
6.4.3	维纳滤波 (Wiener Filtering)	62
6.4.4	约束最小二乘方滤波 (CLS Filtering)	67
6.5	图像复原例题集	68
6.5.1	例题 1: 线性位置不变系统分析	68

6.5.2	例题 2: 噪声统计特性与去噪	68
6.5.3	例题 3: 高级维纳滤波设计	69
6.5.4	例题 4: 约束最小二乘方滤波	70
6.5.5	例题 5: 复合噪声环境下的最优滤波	71
7	形态学操作	73
7.1	基础概念	73
7.1.1	基本集合运算	73
7.1.2	形态学特有操作	73
7.2	二值图像的基本形态学操作	73
7.2.1	腐蚀 (Erosion)	74
7.2.2	膨胀 (Dilation)	75
7.2.3	对偶性 (Duality)	75
7.3	复合形态学操作	75
7.3.1	开运算 (Opening)	76
7.3.2	闭运算 (Closing)	77
7.4	基本形态学算法	78
7.4.1	边界提取 (Boundary Extraction)	78
7.4.2	孔洞填充 (Hole Filling)	79
7.4.3	连通分量提取 (Connected Components Extraction)	79
7.4.4	细化 (Thinning)	80
7.4.5	骨架化 (Skeletonization)	81
7.4.6	击中或击不中变换 (Hit-or-Miss Transform)	82
7.5	灰度图像形态学	83
7.5.1	灰度膨胀	83
7.5.2	灰度腐蚀	84
7.5.3	灰度开运算和闭运算	86
8	图像分割基础	88
8.1	图像分割的定义与目标	88
8.2	基于不连续性的分割: 边缘检测	88
8.2.1	基本原理: 利用导数检测不连续点	88
8.2.2	一阶与二阶导数 (考点)	88
8.2.3	点、线检测	90
8.2.4	边缘检测 (Edge Detection)	90
8.2.5	梯度算子 (一阶导数)	91
8.2.6	高斯-拉普拉斯 (LoG) 算子	93
8.2.7	Canny 边缘检测 (非常重要考点)	94

8.3	基于相似性的分割：阈值法	96
8.3.1	阈值处理 (Thresholding) 基础	96
8.3.2	Otsu's 方法 (最佳全局阈值法 - 大津法)	97
8.3.3	阈值法的改进	99
8.3.4	自适应阈值处理	99
8.4	作业重点提示	100
9	图像分割算法	101
9.1	前期内容回顾	101
9.2	图像分割算法概览	101
9.3	基于区域的分割方法 (Region-based Methods)	101
9.3.1	区域生长法 (Region Growing)	102
9.3.2	区域分裂与合并法 (Region Splitting and Merging)	102
9.4	分水岭算法 (Watershed Algorithm)	103
9.4.1	基本思想与地形学类比	103
9.4.2	算法流程：“大坝构建”法	104
9.4.3	具体实现步骤与示例	104
9.4.4	问题与改进：过分割与标记 (Markers)	105
9.5	基于聚类的分割方法 (Clustering-based Methods)	105
9.5.1	K-均值聚类 (K-Means Clustering)	105
9.6	SLIC 超像素 (SLIC Superpixels)	107
9.7	分割评估与验证	107
10	医学图像配准与可视化	108
10.1	图像配准 (Image Registration) 概述	108
10.1.1	什么是图像配准及其重要性	108
10.1.2	图像配准的基本方法分类	108
10.1.3	图像配准基本流程 (Registration Pipeline)	109
10.2	变换模型详解 (Transformation Model)	109
10.2.1	线性变换 (Linear Transformation)	110
10.2.2	非线性变换 (Non-linear Transformation)	117
10.2.3	图像重采样与插值 (Resampling and Interpolation)	118
10.3	相似性度量详解 (Similarity Metric) (重要考点)	119
10.3.1	单模态 (Intra-modality) 配准度量	119
10.3.2	多模态 (Inter-modality) 配准度量：互信息	120
10.3.3	相似性度量方法对比总结 (重要考点)	126
10.4	优化器与配准策略 (Optimizer)	127
10.4.1	优化挑战与策略	127

10.5 非线性配准 (Non-linear Registration)	127
10.5.1 微分同胚变换 (Diffeomorphism) (重要考点)	127
10.5.2 非线性配准算法详解	128
10.5.3 形变场分析: 雅可比行列式与 TBM	133
11 特征提取	136
11.1 特征提取概述 (Background)	136
11.1.1 什么是特征?	136
11.1.2 特征的分类	136
11.2 形状特征 (Shape Feature)	136
11.2.1 基于边界的描述符 (Boundary Descriptors)	136
11.2.2 基于区域的描述符 (Region Descriptors)	139
11.3 一阶特征 (First Order Features)	141
11.4 纹理特征 (Texture Features)	144
11.4.1 灰度共生矩阵 (GLCM) (重要考点)	144
11.4.2 谱方法 (Spectral Approaches)	146
11.5 影像组学 (Radiomics)	146
11.5.1 定义与工作流程	146
11.5.2 应用与挑战	147
11.5.3 图像增强技术	147
12 机器学习与深度学习	152
12.1 引言: 医学中的人工智能 (Introduction: AI in Medicine)	152
12.1.1 发展历程 (Historical Development)	152
12.1.2 大模型时代 (The Era of Large Models)	152
12.1.3 典型的医疗大模型 (Typical Medical Large Models)	152
12.2 机器学习基础 (Machine Learning Fundamentals)	153
12.2.1 传统方法: 特征工程 (Traditional Method: Feature Engineering)	153
12.2.2 从特征工程到机器学习	154
12.2.3 线性不可分问题与核技巧思想	154
12.3 神经网络基础 (Fundamentals of Neural Networks)	154
12.3.1 感知机 (The Perceptron)	154
12.3.2 激活函数 (Activation Functions)	155
12.3.3 XOR 问题: 神经网络非线性能力的核心示例	155
12.3.4 神经网络的训练: 反向传播 (Training NNs: Backpropagation)	156
12.4 卷积神经网络 (Convolutional Neural Networks - CNN)	157
12.4.1 核心思想与动机	157
12.4.2 关键组件	157

- 12.4.3 CNN 基本运算（手动计算） 158
 - 12.4.4 经典 CNN 架构 160
- 12.5 注意力机制 (Attention Mechanisms) 161
 - 12.5.1 通道与空间注意力 161
 - 12.5.2 自注意力与 Transformer 161
- 12.6 生成模型 (Generative Models) 162
 - 12.6.1 生成对抗网络 (Generative Adversarial Networks - GAN) 162
 - 12.6.2 扩散模型 (Diffusion Models) 163

复习说明

知识点梳理完后得从教材后面、布置的作业的习题练习！学海可以作为你的资料...

1 数字图像基础

数字图像处理的第一步是将物理世界中的连续信号转换为计算机可以识别的数字信号。这个过程被称为**数字化** (Digitalization), 它主要包含两个核心步骤: **采样** (Sampling) 和 **量化** (Quantization)。

1.1 采样 (Sampling)

采样是对空间坐标的离散化。想象一下, 你看到的真实世界是连续不断的画面, 为了让计算机”看到”这个画面, 我们必须在画面上取有限个点, 这些点的坐标位置就是采样。

1.1.1 从连续到离散

- 一个连续的图像信号 $f(x, y)$, 其坐标 (x, y) 是连续变化的。
- 采样过程就是用一个网格去”切割”这个连续图像。每个网格交叉点或网格单元就是一个**像素** (Pixel)。
- 采样后, 图像就被表示成一个矩阵 (或二维数组) A 。矩阵中的每个元素 $f(x, y)$ 对应一个特定位置的像素值, 其中 x 和 y 都是整数索引。

1.1.2 矩阵大小与空间分辨率

- **矩阵大小 (Matrix Size):** 指的是图像所包含的像素行数和列数, 例如 256×256 , 128×128 , 64×64 等。
- 在其他条件不变的情况下, 矩阵越大, 采样的点越多, 图像包含的细节就越丰富, 视觉上看起来也越清晰。如果采样点过少 (矩阵太小), 图像会变得模糊不清, 出现马赛克效应。
- **重要区别: 物理分辨率 (Physical Resolution)** 不仅仅由矩阵大小决定。物理分辨率描述的是图像中一个像素代表的真实物理尺寸 (例如: 毫米/像素)。一个 $(256, 256)$ 的图像和一个 $(128, 128)$ 的图像, 如果前者每个像素代表 $1 \times 1\text{mm}$, 后者每个像素代表 $2 \times 2\text{mm}$, 那么它们覆盖的物理区域大小是一样的, 但前者具有更高的物理分辨率。

1.2 量化 (Quantization)

量化是对像素幅值 (亮度/灰度值) 的离散化。在采样获得每个像素点后, 我们需要给这个点的亮度赋予一个整数值。

1.2.1 灰度级 (Gray Level)

- 量化的本质是把连续的亮度范围划分成有限个等级。
- 量化的等级数量由位深度 (bit depth) K 决定。灰度级的总数 L 为 $L = 2^K$ 。
- 例如，一个 8-bit 图像，其 $K = 8$ ，拥有 $L = 2^8 = 256$ 个灰度级。这些灰度级的取值范围通常是 $[0, 255]$ ，其中 0 代表最暗（黑色），255 代表最亮（白色）。

1.2.2 位深度与图像质量

- 位深度 K 越高，可表示的灰度级就越多，图像的色彩或灰度过渡就越平滑，细节层次也越丰富。
- 如果位深度过低（例如 $K = 2$ ，只有 4 个灰度级），图像会出现明显的色带或等高线状的轮廓，这种现象称为假轮廓 (False Contouring)。

1.2.3 动态范围与存储

- 动态范围 (Dynamic Range)**: 图像中可感知的最大和最小灰度级之间的范围。量化水平会影响图像的动态范围。
- 高动态范围 (HDR, High Dynamic Range)** 技术通过合并不同曝光度的图像，来实现在高光和阴影区域都能保留细节，获得更大的动态范围。
- 存储大小**: 一张 $M \times N$ 的图像，如果位深度为 K bits，那么它占用的存储空间为 $M \times N \times K$ 。

1.3 插值 (Interpolation)

插值是一种根据已知数据点来估计未知位置数据点数值的技术。在图像处理中，插值被广泛应用于图像的几何变换，如缩放 (zooming/shrinking)、旋转 (rotation) 等。

1.3.1 为什么需要插值?

- 当你对于一个图像进行旋转或缩放时，原图像中的像素点（整数坐标）经过变换后，其新的坐标位置通常不再是整数。
- 为了在新的坐标系（目标图像）中生成一个完整的图像，我们必须为每一个整数位置的像素点计算其对应的像素值。这个计算过程就是插值。
- 具体做法通常是**反向映射 (Reverse mapping)**: 遍历目标图像的每一个像素点，通过坐标反变换计算出它在源图像中的对应位置（可能是浮点坐标），然后利用源图像中该位置周围的像素点来估计其像素值。

1.3.2 邻域 (Neighborhood)

- 插值计算通常需要用待求点周围的邻近像素。
- 4-邻域 (N_4): 一个像素点 (x, y) 的上、下、左、右四个邻居。
- 8-邻域 (N_8): 包括 4-邻域和对角线上的四个邻居。

1.3.3 常用插值方法

最近邻插值 (Nearest-Neighbor Interpolation)

- 最简单的方法。直接选择在源图像中离它最近的那个像素的灰度值作为新像素的值。
- 优点: 计算速度快。
- 缺点: 会产生明显的块状效应或锯齿 ("blocky" or "jagged" appearance)。

双线性插值 (Bilinear Interpolation)

- 使用待求点在源图像中对应的 4-邻域像素点进行计算。
- 其本质是在 x 和 y 两个方向上分别进行线性插值 (Linear Interpolation)。
- 优点: 效果比最近邻平滑得多, 计算量适中。
- 缺点: 可能会导致图像的一些高频细节 (如边缘) 模糊。

双三次插值 (Bicubic Interpolation)

- 使用待求点在源图像中对应的 16-邻域 (4×4) 像素点进行计算。它基于更复杂的三次多项式来拟合周围的像素点。
- 优点: 能产生比双线性插值更平滑、边缘更清晰的图像, 效果最接近原始图像 (Groundtruth)。
- 缺点: 计算复杂度最高。

好像后面会具体计算双线性插值? 这里需要直接掌握吗? 彩色图片处理做要求吗?

2 空间域运算

2.1 基本数学工具 (Basic Mathematical Tools)

这一部分介绍直接对图像像素值进行操作的基本运算。

2.1.1 算术运算 (Arithmetic Operations)

这些都是逐像素 (Pixel-wise) 的操作，即对两幅图像相同坐标 (x, y) 的像素值进行运算。

加法 $s(x, y) = f_1(x, y) + f_2(x, y)$

- **主要应用:** 降噪。对于一幅被加性随机噪声 $\eta(x, y)$ 污染的图像 $g(x, y) = f(x, y) + \eta(x, y)$ ，如果噪声是零均值的，我们可以通过对多幅带噪图像求平均来有效降低噪声。
- **原理:** 对 K 幅独立的带噪图像求平均， $\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y)$ 。根据统计学原理，均值的期望等于原始图像 $f(x, y)$ ，而均值的方差是原始噪声方差的 $\frac{1}{K}$ ($\sigma_{\bar{g}(x, y)}^2 = \frac{1}{K} \sigma_{\eta(x, y)}^2$)。 K 越大，降噪效果越明显。

减法 $d(x, y) = f_1(x, y) - f_2(x, y)$

- **主要应用:** 变化检测，即找出两幅图像之间的差异。
- **经典案例:** 数字减影血管造影 (DSA)。通过将包含造影剂的图像减去没有造影剂的背景图像 (Mask)，可以清晰地凸显出血管结构。

乘法 $p(x, y) = f_1(x, y) \times f_2(x, y)$

- **主要应用:** 掩模操作 (Masking) 或感兴趣区域 (ROI) 提取。通过创建一个二值掩模图像 (ROI 内为 1, ROI 外为 0)，与原图相乘，可以只保留感兴趣区域的内容。

除法 $v(x, y) = f_1(x, y) \div f_2(x, y)$

- **主要应用:** 校正/归一化。例如，校正由于传感器或照明不均匀造成的影响。

2.1.2 集合与逻辑运算 (Set and Logical Operations)

主要用于处理二值图像 (Binary Image)。

- **集合运算:** 包括交集 ($A \cap B$)、并集 ($A \cup B$)、补集 (A^c)、差集 ($A - B$) 等。
- **逻辑运算:** 对像素进行位运算，如 NOT (非)、AND (与)、OR (或)、XOR (异或)。这些运算在图像分割、特征提取等领域非常有用。

2.2 几何变换 (Geometric Transform)

几何变换改变的是像素的空间位置（坐标），而不是像素的灰度值。它通常需要与插值 (Interpolation) 结合使用，因为变换后的坐标可能不是整数。

2.2.1 仿射变换 (Affine Transformation)

仿射变换是一种重要的线性变换，它保持了图像的”平直性”（直线变换后仍是直线）和”平行性”（平行线变换后仍平行）。

2D 仿射变换 使用齐次坐标 (Homogeneous Coordinates) 可以将平移、旋转、缩放等操作统一用矩阵乘法表示：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

刚体变换 (Rigid-body Transform)：只包含平移 (Translation) 和旋转 (Rotation)。这种变换保持了物体的大小和形状。

通用仿射变换矩阵 A 如下：

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

变换公式为：

$$\begin{cases} x' = a_{11}x + a_{12}y + a_{13} \\ y' = a_{21}x + a_{22}y + a_{23} \end{cases}$$

平移 (Translation)

- 由参数 a_{13} 和 a_{23} 控制。
- a_{13} (t_x) 控制沿 x 轴的平移距离。
- a_{23} (t_y) 控制沿 y 轴的平移距离。

平移矩阵：

$$A = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

缩放 (Scaling)

- 由对角线上的参数 a_{11} 和 a_{22} 控制。
- a_{11} (c_x) 控制沿 x 轴的缩放比例。
- a_{22} (c_y) 控制沿 y 轴的缩放比例。
- 如果 c_x 或 c_y 为负值, 则会产生**反射 (镜像)** 效果。

缩放矩阵:

$$A = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

旋转 (Rotation) 围绕原点逆时针旋转 θ 角度, 由 $a_{11}, a_{12}, a_{21}, a_{22}$ 四个参数共同决定:

$$a_{11} = \cos \theta \quad (1)$$

$$a_{12} = -\sin \theta \quad (2)$$

$$a_{21} = \sin \theta \quad (3)$$

$$a_{22} = \cos \theta \quad (4)$$

旋转矩阵:

$$A = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

剪切/错切 (Shearing)

- 由非对角线上的参数 a_{12} 和 a_{21} 在没有旋转的情况下控制。
- a_{12} 控制水平方向的剪切。
- a_{21} 控制垂直方向的剪切。

通过组合这些参数, 可以实现任意复杂的仿射变换。例如, 先旋转再平移, 只需将相应的矩阵相乘即可。需要注意的是, 矩阵乘法的顺序至关重要, 不同的操作顺序会导致不同的变换结果。

3D 仿射变换 原理与 2D 类似, 但更复杂。

- **3D 刚体变换:** 由 3 个平移 (沿 X, Y, Z 轴) 和 3 个旋转 (绕 X, Y, Z 轴, 分别称为 Pitch, Roll, Yaw) 组成, 共 6 个自由度。
- **变换顺序很重要:** 不同的旋转顺序 (如 $R_z R_y R_x$ vs $R_x R_y R_z$) 会得到完全不同的结果。

2.2.2 非线性变换 (Non-linear Transform)

当仿射变换不足以对齐两幅图像时（例如，由于器官变形），就需要使用更复杂的非线性变换。

形变场 (Deformation Field) 或 流场 (Flow Field)：非线性变换为图像中的每个像素都定义了一个独立的位移向量，可以实现局部的、复杂的形变。常见的有弹性 (Elastic) 和微分同胚 (Diffeomorphic) 变换。

2.2.3 几何变换的应用：图像配准 (Image Registration)

图像配准是将两幅或多幅图像在空间上对齐的过程，是医学图像分析中的关键步骤。

目的

- **个体间比较**：将不同病人的大脑图像配准到同一个标准空间（如 MNI 模板），以进行组分析。
- **多模态融合**：对齐来自同一个病人的不同模态图像（如 CT 和 MRI），用于放射治疗计划等。

方法 可以通过匹配控制点（如 MRI 标记物）或基于图像强度信息来实现。

2.3 强度变换 (Intensity Transformations)

强度变换操作直接作用于像素的灰度值，其通用形式为 $g(x, y) = T[f(x, y)]$ ，其中 T 是一个变换函数。

2.3.1 点处理 (Point Processing)

点处理是一种最简单的强度变换，新像素的灰度值仅取决于原始图像中该点自身的灰度值。变换函数的邻域大小为 1×1 。

基本变换函数 线性变换：

- **负片变换 (Negative)**： $s = L - 1 - r$ 。效果是黑变白，白变黑，可以增强嵌入在暗色区域的白色细节。
- **恒等变换 (Identity)**： $s = r$ 。输出等于输入。

非线性变换：

- **对数变换**： $s = c \log(1 + r)$ 。扩展输入图像中的低灰度值区域，同时压缩高灰度值区域。适合增强过暗图像的细节。

- 幂律（伽马）变换： $s = cr^\gamma$
 - 当 $\gamma < 1$ 时，效果类似对数变换，扩展低灰度级。
 - 当 $\gamma > 1$ 时，效果与 $\gamma < 1$ 相反，扩展高灰度级。
 - 伽马校正广泛用于校正显示器的非线性响应。

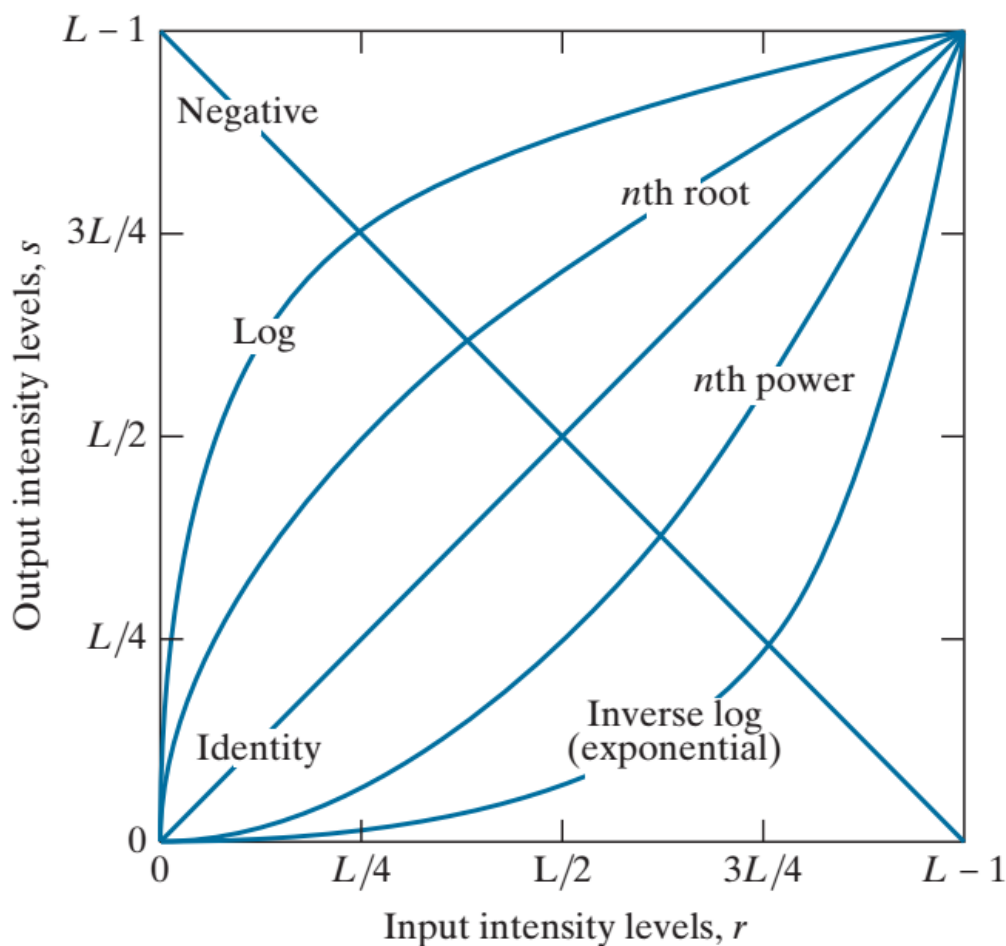


图 1: 强度变换函数示例

分段线性变换 (Piecewise-Linear Transformation)

- 对比度拉伸 (Contrast Stretching): 通过一个分段线性函数，将输入图像中某个灰度范围拉伸到更宽的输出范围，从而提高该范围内的对比度。
- 灰度级切片 (Intensity-level Slicing): 突出图像中特定的灰度范围，可以将其余部分变暗或保持不变。
- 阈值处理 (Thresholding): 一种极端的分段函数，将图像转换为二值图像。

2.3.2 直方图的统计矩 (Statistical Moments)

图像的直方图不仅是一个图形，还包含了重要的统计信息，这些信息可以通过矩来描述。对于一个具有 L 个灰度级、灰度值为 z_k 、对应概率为 $p(z_k)$ 的图像：

一阶矩 (1st order moment) - 均值 (Mean)

- 公式: $E = \sum_{k=0}^{L-1} z_k p(z_k)$
- 物理意义: 代表了图像的平均亮度 (Average Brightness)。均值越高，图像整体越亮。

二阶矩 (2nd order moment) - 方差 (Variance)

- 公式: $\sigma^2 = \sum_{k=0}^{L-1} (z_k - E)^2 p(z_k)$
- 物理意义: 描述了灰度值相对于均值的离散程度，反映了图像的对比度 (Contrast)。方差越大，意味着灰度分布越广，图像的对比度越高。

N 阶矩 (Nth order moment)

- 公式: $\mu_n = \sum_{k=0}^{L-1} (z_k - E)^n p(z_k)$
- 更高阶的矩描述了直方图分布的更细微特征，例如三阶矩与分布的偏斜度 (Skewness) 有关，四阶矩与峰度 (Kurtosis) 有关。

2.3.3 直方图处理 (Histogram Processing)

直方图是图像强度分布的图形化表示，它统计了每个灰度级出现的像素数量。

直方图与图像外观

- 暗图像: 直方图集中在低灰度区域。
- 亮图像: 直方图集中在高灰度区域。
- 低对比度图像: 直方图分布范围很窄。
- 高对比度图像: 直方图分布范围很广，覆盖了大部分灰度级。

直方图均衡化 (Histogram Equalization) 目标: 通过一个变换函数 $s = T(r)$, 自动地将原始图像的直方图重新分布, 使其在整个灰度范围内尽可能均匀 (uniform), 从而增强图像的全局对比度。

核心公式 (离散): 变换函数是原始图像灰度级的累积分布函数 (CDF):

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) = (L - 1) \sum_{j=0}^k \frac{n_j}{MN}$$

其中 r_k 是输入灰度级, $p_r(r_j)$ 是第 j 级灰度的概率, L 是总灰度级数。

注意: 对于有大量背景像素 (如灰度值为 0) 的图像, 通常会忽略背景像素, 从灰度级 1 开始计算, 以避免背景影响均衡化效果。

直方图均衡化例子 假设有一幅 64×64 的图像, 共 4096 个像素, 采用 3-bit 量化, 即灰度级 $L = 8$ (范围从 0 到 7)。

步骤 1: 计算原始直方图和概率密度函数 (PDF)

首先统计每个灰度级 r_k 的像素数 n_k , 然后计算其概率 $p_r(r_k) = n_k / (MN)$ 。

r_k (输入灰度级)	n_k (像素数)	$p_r(r_k)$ (概率)
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

表 1: 原始直方图数据

步骤 2: 计算累积分布函数 (CDF) 并应用变换公式

使用均衡化变换公式 $s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j)$ 。这里 $L - 1 = 7$ 。

$$s_0 = 7 \times p_r(r_0) = 7 \times 0.19 = 1.33 \quad (5)$$

$$s_1 = 7 \times (p_r(r_0) + p_r(r_1)) = 7 \times (0.19 + 0.25) = 3.08 \quad (6)$$

$$s_2 = 7 \times (0.19 + 0.25 + 0.21) = 7 \times 0.65 = 4.55 \quad (7)$$

$$s_3 = 7 \times (0.65 + 0.16) = 7 \times 0.81 = 5.67 \quad (8)$$

$$s_4 = 7 \times (0.81 + 0.08) = 7 \times 0.89 = 6.23 \quad (9)$$

$$s_5 = 7 \times (0.89 + 0.06) = 7 \times 0.95 = 6.65 \quad (10)$$

$$s_6 = 7 \times (0.95 + 0.03) = 7 \times 0.98 = 6.86 \quad (11)$$

$$s_7 = 7 \times (0.98 + 0.02) = 7 \times 1.00 = 7.00 \quad (12)$$

步骤 3: 取整并生成新的直方图

将计算出的 s_k 值四舍五入到最近的整数，得到新的输出灰度级：

原始灰度级 (r_k)	映射到新灰度级 ($s_k \rightarrow \text{round}$)
0	1
1	3
2	5
3	6
4	6
5	7
6	7
7	7

表 2: 直方图均衡化映射结果

结果: 原来灰度级为 0 的 790 个像素，现在灰度级都变为 1。原来为 1 的 1023 个像素，现在都变为 3。原来为 3 和 4 的像素，现在都合并到了灰度级 6。最终，像素会重新分布到新的灰度级上，形成一个更宽、更均匀的直方图，从而增强了对比度。

直方图匹配 (或规定化, Histogram Matching/Specification) **目标:** 与均衡化不同，直方图匹配的目标是将输入图像的直方图变换为某个指定形状的直方图。

应用: 在图像配准前，将待配准图像的直方图匹配到目标图像（如标准模板），可以消除整体亮度差异，有助于提高配准的准确性。

核心原理: 直方图匹配的原理非常巧妙。它并不直接从输入图像变到目标图像，而是利用直方图均衡化作为一座”共同的桥梁”。

1. **输入图像的均衡化 $s = T(r)$:** 我们对输入图像 r 做均衡化，得到一个理论上”平坦”的直方图，其灰度值为 s 。

2. **目标图像的均衡化** $v = G(z)$: 我们也对我们想要模仿的目标直方图 z 做均衡化, 得到另一个理论上”平坦”的直方图, 其灰度值为 v 。

现在, s 和 v 都被映射到了一个共同的、理想化的平坦空间。那么, 如果输入图像中一个像素均衡化后的值为 s_k , 我们只需要在目标图像的均衡化结果中找到一个**最接近 s_k 的值**, 然后看看这个值是由哪个目标灰度 z_q 产生的, 那么这个 z_q 就是我们最终要映射到的值。

这就是理论核心: $s_k \approx G(z_q)$, 我们要通过 s_k 反向查找到 z_q 。

算法步骤:

1. 计算输入图像的变换函数 $s_k = T(r_k)$ (即均衡化过程)。
2. 计算目标直方图 $p_z(z)$ 的变换函数 $G(z_q) = (L-1) \sum_{i=0}^q p_z(z_i)$ 。
3. 对于每个均衡化后的值 s_k , 在目标变换函数 G 中找到一个最接近的 $G(z_q)$, 并将原始像素值 r_k 映射到对应的 z_q 。

直方图匹配实例详解 现在, 我们来完成一个完整的匹配推导。我们继续使用前面的例子。

前提: 我们已经完成了上述均衡化过程, 得到了输入图像的均衡化值 s_k 。现在, 我们有一个指定的目标直方图 $p_z(z_q)$ 。

步骤 1: 计算目标直方图的变换函数 $G(z_q)$

与均衡化类似, 我们计算目标直方图的累积分布, 得到其变换函数 $G(z_q) = (L-1) \sum_{i=0}^q p_z(z_i)$ 。

z_q (目标灰度级)	$p_z(z_q)$ (目标概率)	$G(z_q)$ 计算值	$G(z_q)$ 取整
0	0.00	0.00	0
1	0.00	0.00	0
2	0.00	0.00	0
3	0.15	$7 \times (0.15) = 1.05$	1
4	0.20	$7 \times (0.15 + 0.20) = 2.45$	2
5	0.30	$7 \times (0.35 + 0.30) = 4.55$	5
6	0.20	$7 \times (0.65 + 0.20) = 5.95$	6
7	0.15	$7 \times (0.85 + 0.15) = 7.00$	7

表 3: 目标直方图的变换函数计算

步骤 2: 推导过程——一步步完成映射

现在, 我们来完成最关键的一步推导。我们已经拥有了两个关键的列表:

1. **输入图像的均衡化结果 (s_k):** 这是前面例子中计算出来的, 我们只关心**取整后**的值。
 - $s_0 = 1$ (来自原始灰度 $r_0 = 0$)
 - $s_1 = 3$ (来自原始灰度 $r_1 = 1$)

- $s_2 = 5$ (来自原始灰度 $r_2 = 2$)
- $s_3 = 6$ (来自原始灰度 $r_3 = 3$)
- $s_4 = 6$ (来自原始灰度 $r_4 = 4$)
- $s_5 = 7$ (来自原始灰度 $r_5 = 5$)
- $s_6 = 7$ (来自原始灰度 $r_6 = 6$)
- $s_7 = 7$ (来自原始灰度 $r_7 = 7$)

2. 目标直方图的均衡化结果 $G(z_q)$: 这是上表中计算出来的, 我们也只关心取整后的值。这是一个”查找表”。

- $G(z_0 = 0) = 0$
- $G(z_1 = 1) = 0$
- $G(z_2 = 2) = 0$
- $G(z_3 = 3) = 1$
- $G(z_4 = 4) = 2$
- $G(z_5 = 5) = 5$
- $G(z_6 = 6) = 6$
- $G(z_7 = 7) = 7$

关键的匹配步骤 现在, 我们拿着输入图像的每一个均衡化值 s_k , 去目标均衡化的查找表 $G(z_q)$ 中寻找它的”归宿” z_q 。

匹配规则: 对于每一个 s_k , 我们在 $G(z_q)$ 列表中, 找到第一个大于或等于 s_k 的值, 然后将这个 s_k 对应的原始灰度 r_k 映射到那个 $G(z_q)$ 所对应的 z_q 。

- 对于 $s_0 = 1$: 我们在 G 列表中寻找: $\{0, 0, 0, 1, 2, 5, 6, 7\}$ 。第一个大于或等于 1 的值是 $G(z_3) = 1$ 。因此, s_0 匹配到了 z_3 。结论: 原始灰度 $r_0 = 0$ 最终映射到目标灰度 $z = 3$ 。
- 对于 $s_1 = 3$: 我们在 G 列表中寻找: $\{0, 0, 0, 1, 2, 5, 6, 7\}$ 。第一个大于或等于 3 的值是 $G(z_5) = 5$ 。(注意: 虽然 $G(z_4) = 2$ 在数值上更接近 3, 但标准的匹配算法通常采用”大于等于”规则, 以保证映射的单调性)。因此, s_1 匹配到了 z_5 。结论: 原始灰度 $r_1 = 1$ 最终映射到目标灰度 $z = 5$ 。
- 对于 $s_2 = 5$: 我们在 G 列表中寻找: $\{0, 0, 0, 1, 2, 5, 6, 7\}$ 。第一个大于或等于 5 的值是 $G(z_5) = 5$ 。因此, s_2 匹配到了 z_5 。结论: 原始灰度 $r_2 = 2$ 最终映射到目标灰度 $z = 5$ 。
- 对于 $s_3 = 6$ (来自 $r=3$) 和 $s_4 = 6$ (来自 $r=4$): 我们在 G 列表中寻找: $\{0, 0, 0, 1, 2, 5, 6, 7\}$ 。第一个大于或等于 6 的值是 $G(z_6) = 6$ 。因此, s_3 和 s_4 都匹配到了 z_6 。结论: 原始灰度 $r_3 = 3$ 和 $r_4 = 4$ 最终都映射到目标灰度 $z = 6$ 。

- 对于 s_5, s_6, s_7 (值均为 7): 我们在 G 列表中寻找: $\{0, 0, 0, 1, 2, 5, 6, 7\}$ 。第一个大于或等于 7 的值是 $G(z_7) = 7$ 。因此, s_5, s_6, s_7 都匹配到了 z_7 。结论: 原始灰度 r_5, r_6, r_7 最终都映射到目标灰度 $z = 7$ 。

最终的完整映射表 通过上面的推导, 我们得到了从最开始的原始灰度 r 到最终目标灰度 z 的完整映射关系:

原始灰度级 (r_k)	均衡化值 (s_k)	匹配到 $G(z_q)$	最终目标灰度级 (z_q)
0	1	$G(3) = 1$	3
1	3	$G(5) = 5$	5
2	5	$G(5) = 5$	5
3	6	$G(6) = 6$	6
4	6	$G(6) = 6$	6
5	7	$G(7) = 7$	7
6	7	$G(7) = 7$	7
7	7	$G(7) = 7$	7

表 4: 直方图匹配完整映射关系

结果: 原来图像中灰度为 0 的 790 个像素, 在新图像中灰度都会变成 3; 原来灰度为 1 的 1023 个像素, 新灰度会变成 5……以此类推。操作完成后, 新图像的直方图形状就会非常接近我们设定的目标直方图了。

2D 仿射变换肯定要刷题掌握, 但是 3D 仿射变换会做要求吗? 直方图均衡需要掌握直方图匹配需要掌握

3 空间域滤波

本章的核心是利用像素的邻域信息来修改或增强图像，这与第三章只依赖单个像素值的点处理完全不同。

3.1 空间滤波基础 (Basics of Spatial Filtering)

空间滤波是通过一个小的矩阵，即**滤波器** (Filter)，也称为**掩模** (mask)、**核** (kernel)、**模板** (template) 或**窗口** (window)，在图像上进行移动操作的过程。

3.1.1 核心操作：相关与卷积

相关 (Correlation) 相关是滤波器在图像上”滑动”并计算加权和的过程。在每个位置，滤波器中每个系数与其覆盖的图像像素值相乘，然后求和，得到该位置的输出像素值。

一维相关公式：

$$g(x) = \sum_{i=-a}^a w(i)f(i+x)$$

二维相关公式：

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

卷积 (Convolution) 与相关非常相似，但在应用前，需要将滤波器（核）旋转 180 度。

一维卷积公式：

$$g(x) = \sum_{i=-a}^a w(i)f(x-i)$$

重要性质：

- 卷积满足交换律： $f * g = g * f$
- 卷积满足结合律： $f * (g * h) = (f * g) * h$
- 相关不满足这些性质

边界处理 (Handling the edge) 当滤波器移动到图像边缘时，其一部分会超出图像范围。处理方法有：

- **补零** (Zero padding)：在图像边界外填充 0。这是最常见的方法，但可能会在图像边缘产生黑边。
- **镜像** (Mirror) 或 **复制** (Replicated)：将图像边界的像素值镜像或复制到外部。

例题 1：相关运算计算 给定一个 3×3 的图像区域和 3×3 的滤波器，计算相关运算的结果。

图像区域：

$$F = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

滤波器：

$$W = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

解： 相关运算： $g = \sum_{s=-1}^1 \sum_{t=-1}^1 w(s, t) \cdot f(1+s, 1+t)$

$$g = 0 \cdot 1 + (-1) \cdot 2 + 0 \cdot 3 + (-1) \cdot 4 + 5 \cdot 5 + (-1) \cdot 6 + 0 \cdot 7 + (-1) \cdot 8 + 0 \cdot 9$$

$$g = 0 - 2 + 0 - 4 + 25 - 6 + 0 - 8 + 0 = 5$$

3.1.2 空间域与频率域的关系

空间滤波和频率滤波是等价的，它们是同一枚硬币的两面。

卷积定理： 在空间域中进行卷积运算，等同于在频率域中进行乘法运算。

滤波器类型：

- 平滑滤波器（如均值滤波）对应低通滤波器，允许低频分量通过，抑制高频分量。
- 锐化滤波器（如拉普拉斯）对应高通滤波器，允许高频分量通过，抑制低频分量。

3.2 低通滤波器 (Low Pass Filter)

低通滤波器的主要作用是平滑图像和降低噪声，它通过模糊来减弱图像中的尖锐变化（高频信息）。

3.2.1 线性低通滤波器

盒状滤波器 (Box Filter) 也叫均值滤波器。

简单均值： 核内所有系数均为 1，最后除以系数总数（例如， 3×3 核除以 9），即计算邻域内像素的平均值。

3×3 均值滤波器：

$$W = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

加权均值： 给予邻域中心像素更高的权重。

效果： 核越大，模糊效果越明显。

例题 2：均值滤波计算 对下面的图像区域应用 3×3 均值滤波器：

$$\text{图像} = \begin{bmatrix} 100 & 120 & 140 \\ 110 & 200 & 130 \\ 105 & 115 & 125 \end{bmatrix}$$

$$\text{解：均值滤波器：} W = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{中心像素的输出值：} g = \frac{1}{9}(100 + 120 + 140 + 110 + 200 + 130 + 105 + 115 + 125) = \frac{1145}{9} \approx 127$$

高斯滤波器 (Gaussian Filter) 核的系数根据二维高斯函数生成：

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

中心权重最大，向外逐渐减小。

性质：

- σ （标准差）控制了模糊程度， σ 越大，模糊越厉害。
- 通常核的大小取为 $[-3\sigma, 3\sigma]$ 。
- 高斯滤波器具有很好的旋转对称性。
- 连续进行两次高斯滤波等效于一次标准差为 $\sigma_{final} = \sqrt{\sigma_1^2 + \sigma_2^2}$ 的高斯滤波。

例题 3：高斯滤波器设计 设计一个 3×3 的高斯滤波器， $\sigma = 1$ 。

解：对于 3×3 核，坐标为 $\{(-1, -1), (-1, 0), \dots, (1, 1)\}$

计算各点的高斯值：

$$w(-1, -1) = \frac{1}{2\pi} e^{-\frac{(-1)^2 + (-1)^2}{2}} = \frac{1}{2\pi} e^{-1} \approx 0.0596 \quad (13)$$

$$w(-1, 0) = \frac{1}{2\pi} e^{-\frac{(-1)^2 + 0^2}{2}} = \frac{1}{2\pi} e^{-0.5} \approx 0.0965 \quad (14)$$

$$w(0, 0) = \frac{1}{2\pi} e^{-\frac{0^2 + 0^2}{2}} = \frac{1}{2\pi} \approx 0.1592 \quad (15)$$

归一化后的 3×3 高斯核：

$$W \approx \begin{bmatrix} 0.0751 & 0.1238 & 0.0751 \\ 0.1238 & 0.2042 & 0.1238 \\ 0.0751 & 0.1238 & 0.0751 \end{bmatrix}$$

3.2.2 非线性低通滤波器（排序统计滤波器）

这类滤波器是基于邻域内像素值的排序来工作的。

中值滤波器 (Median Filter) 原理：用邻域内所有像素值的中值来替代中心像素的值。

最大优点：在去除脉冲噪声（椒盐噪声）方面效果极佳，并且相比线性滤波器能更好地保留边缘细节，产生的模糊较少。

例题 4：中值滤波计算 对下面的图像区域应用 3×3 中值滤波器：

$$\text{图像} = \begin{bmatrix} 50 & 60 & 55 \\ 45 & 0 & 65 \\ 48 & 58 & 62 \end{bmatrix}$$

（注意：中心像素值为 0，可能是椒盐噪声）

解： 3×3 邻域内的像素值：{50, 60, 55, 45, 0, 65, 48, 58, 62}

排序后：{0, 45, 48, 50, 55, 58, 60, 62, 65}

中值为第 5 个数：55

因此，中心像素的输出值为 55。

最大/最小滤波器 (Max/Min Filter)

- **Max 滤波器：**用邻域内的最大值替换中心像素，用于寻找图像中的亮点或扩大亮区。
- **Min 滤波器：**用邻域内的最小值替换中心像素，用于寻找图像中的暗点或扩大暗区。

例题 5：最大最小滤波计算 对例题 4 的图像区域分别应用最大和最小滤波器。

解： 邻域像素值：{50, 60, 55, 45, 0, 65, 48, 58, 62}

最大滤波器输出：max = 65 最小滤波器输出：min = 0

3.2.3 池化层 (Pooling Layer)

这是深度学习（特别是卷积神经网络）中常见的操作，概念上与低通滤波相关。

作用：对特征图进行降采样 (Downscaling)，减少数据量的同时提取最重要的特征。

类型：

- **最大池化 (Max Pooling)：**在核窗口内取最大值作为输出，能保留最显著的特征。
- **平均池化 (Average Pooling)：**在核窗口内取平均值作为输出，效果类似于均值滤波。

3.3 高通滤波器 (High Pass Filter)

高通滤波器的作用与低通相反，它增强高频分量，主要用于图像锐化和边缘检测。其数学基础是微分。

3.3.1 一阶微分与二阶微分

- 一阶微分 $\frac{\partial f}{\partial x}$ 在边缘（阶跃）和斜坡处会产生非零响应。
- 二阶微分 $\frac{\partial^2 f}{\partial x^2}$ 在边缘的两侧会产生一对正负峰值，并在边缘点上出现零交叉 (Zero crossing)。二阶微分对细节的响应更强，因此更适合图像锐化。

3.3.2 基于二阶微分的滤波器：拉普拉斯算子

拉普拉斯算子 $\nabla^2 f$ 是一个各向同性（旋转不变）的二阶微分算子。

数学推导 二维拉普拉斯算子的定义：

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

步骤 1：一阶偏导数的离散化

对于连续函数的一阶偏导数：

$$\frac{\partial f}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

在离散域中，取 $\Delta x = 1$ ，得到：

$$\frac{\partial f}{\partial x} \approx f(x + 1, y) - f(x, y)$$

类似地：

$$\frac{\partial f}{\partial y} \approx f(x, y + 1) - f(x, y)$$

步骤 2：二阶偏导数的离散化

对 $\frac{\partial f}{\partial x}$ 再求一次偏导：

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x} \left[\frac{\partial f}{\partial x} \right] \approx \frac{\partial}{\partial x} [f(x + 1, y) - f(x, y)]$$

$$= [f(x + 1, y) - f(x, y)] - [f(x, y) - f(x - 1, y)] = f(x + 1, y) - 2f(x, y) + f(x - 1, y)$$

同理：

$$\frac{\partial^2 f}{\partial y^2} \approx f(x, y + 1) - 2f(x, y) + f(x, y - 1)$$

步骤 3：组合得到拉普拉斯算子

$$\nabla^2 f \approx [f(x + 1, y) - 2f(x, y) + f(x - 1, y)] + [f(x, y + 1) - 2f(x, y) + f(x, y - 1)]$$

$$= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

这就对应了 4-邻域拉普拉斯核。

滤波器核推导：

4-邻域核（仅考虑正交方向）：

$$W_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

8-邻域核（包含对角线方向）：为了包含对角线方向的二阶导数，我们可以考虑：

$$\frac{\partial^2 f}{\partial(x+y)^2} \text{ 和 } \frac{\partial^2 f}{\partial(x-y)^2}$$

通过类似的离散化过程，最终得到：

$$W_8 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

拉普拉斯算子的性质

- 各向同性：对图像的旋转不敏感
- 零响应：在均匀区域产生零响应
- 零交叉：边缘位置通常对应拉普拉斯响应的零交叉点
- 符号含义：正值表示比周围暗的点，负值表示比周围亮的点

例题 6：拉普拉斯滤波计算 对下面的图像区域应用 4-邻域拉普拉斯算子：

$$\text{图像} = \begin{bmatrix} 10 & 10 & 10 \\ 10 & 50 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$

$$\text{解：使用 4-邻域拉普拉斯核：} W = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$g = 0 \cdot 10 + 1 \cdot 10 + 0 \cdot 10 + 1 \cdot 10 + (-4) \cdot 50 + 1 \cdot 10 + 0 \cdot 10 + 1 \cdot 10 + 0 \cdot 10$$

$$g = 0 + 10 + 0 + 10 - 200 + 10 + 0 + 10 + 0 = -160$$

3.3.3 基于一阶微分的滤波器：梯度算子

梯度 ∇f 是一个向量，指向图像在该点变化率最大的方向。梯度的大小（模） $M(x, y) = |\nabla f|$ 常用于边缘检测。

Sobel 算子 是一种常用的梯度算子，它在计算 x 和 y 方向的差分，对中心行或列的像素给予了更高的权重（权重为 2），这提供了平滑效果，使其对噪声的敏感度低于简单的差分。

Sobel 算子的核：

x 方向：

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

y 方向：

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

梯度幅值：

$$M(x, y) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

例题 7：Sobel 算子计算 对下面的图像区域应用 Sobel 算子：

$$\text{图像} = \begin{bmatrix} 100 & 100 & 100 \\ 100 & 100 & 200 \\ 100 & 100 & 200 \end{bmatrix}$$

解： 计算 x 方向梯度： $G_x = (-1) \cdot 100 + 0 \cdot 100 + 1 \cdot 100 + (-2) \cdot 100 + 0 \cdot 100 + 2 \cdot 200 + (-1) \cdot 100 + 0 \cdot 100 + 1 \cdot 200$
 $G_x = -100 + 0 + 100 - 200 + 0 + 400 - 100 + 0 + 200 = 300$

计算 y 方向梯度： $G_y = (-1) \cdot 100 + (-2) \cdot 100 + (-1) \cdot 100 + 0 \cdot 100 + 0 \cdot 100 + 0 \cdot 200 + 1 \cdot 100 + 2 \cdot 100 + 1 \cdot 200$
 $G_y = -100 - 200 - 100 + 0 + 0 + 0 + 100 + 200 + 200 = 100$

梯度幅值： $M = \sqrt{300^2 + 100^2} = \sqrt{100000} \approx 316$

3.4 利用空间滤波进行图像增强

图像增强的目标是改善图像的视觉效果，通常是将高通滤波的结果与原图结合。

3.4.1 拉普拉斯增强

将拉普拉斯滤波后的图像（代表边缘信息）按一定比例加回原图。

公式：

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)]$$

其中 c 是增强系数，当拉普拉斯核中心为负时， $c = -1$ ；当中心为正时， $c = 1$ 。
效果：在保留背景信息的同时，使边缘和细节更加突出。

3.4.2 非锐化掩模 (Unsharp Masking)

- 步骤：
- 1. 用低通滤波器（如高斯）对原图 f 进行平滑，得到模糊图像 f_L 。
 - 2. 用原图减去模糊图像，得到模板 $f_h = f - f_L$ ，这个模板只包含图像的高频细节。
 - 3. 将这个模板按一定权重 k 加回原图： $f_{enhance} = f + k \cdot f_h$ 。

3.4.3 组合空间增强

这是一个更复杂的增强流程，结合了多种技术以达到理想效果。
典型流程：

- 1. 用拉普拉斯增强锐化图像。
- 2. 用 Sobel 算子计算梯度图像，并用盒状滤波器平滑它以降低噪声，生成一个掩模。
- 3. 将步骤 1 和 2 的结果相乘，进一步突出共同的强边缘。
- 4. 将步骤 3 的结果加回原图。
- 5. 最后使用伽马变换（幂律拉伸）调整图像的整体对比度和亮度。

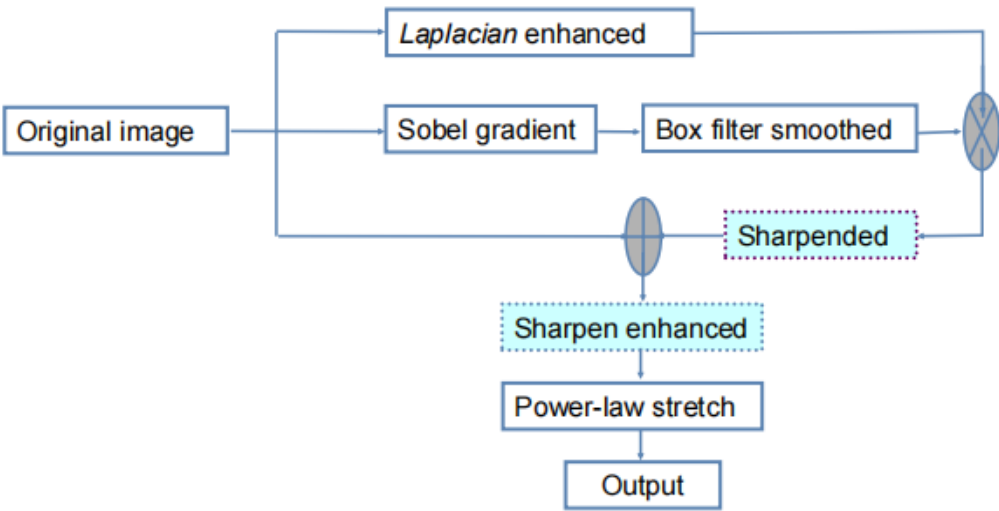


图 2: 组合空间增强的完整流程示例，展示了原图像经过拉普拉斯增强、梯度掩模、图像合成以及伽马变换后的最终增强效果

组合增强技术的核心思想是利用不同滤波器的优势，通过多步骤处理来获得比单一滤波器更好的增强效果。图2展示了这一复杂流程的各个步骤及其视觉效果。

3.5 带通滤波器 (Band Pass Filter)

带通滤波器只允许特定频率范围的信号通过。它们可以由低通和高通滤波器组合而成。

带阻滤波器 (Bandreject):

$$br(x, y) = lp_1(x, y) + hp_2(x, y)$$

带通滤波器 (Bandpass):

$$bp(x, y) = \delta(x, y) - br(x, y)$$

其中 $\delta(x, y)$ 是冲激函数 (单位冲激)。

例题 10: 滤波器类型判断 判断下列滤波器的类型:

$$\text{a) } W_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\text{b) } W_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

解: a) W_1 : 中心系数为正, 周围为负, 且系数和为 $0 + (-1) + 0 + (-1) + 4 + (-1) + 0 + (-1) + 0 = 0$ 。这是一个高通滤波器 (拉普拉斯型)。

b) W_2 : 所有系数为正, 系数和为 $\frac{1}{16}(1 + 2 + 1 + 2 + 4 + 2 + 1 + 2 + 1) = 1$ 。这是一个低通滤波器 (高斯型)。

低通滤波: 平均、高斯 (公式!)、中值、最大最小; 高通滤波: 一阶、二阶 (Laplacian、Sobel); !

CH2 & CH3 & CH4 对应教材中整个第二章的课后习题

4 频率域运算

本章是图像处理从空间域向频率域分析的过渡，是理解高级滤波和图像分析技术的基石。

4.1 傅里叶变换基础 (Fourier Transform Fundamentals)

核心思想：任何函数（信号）都可以表示为不同频率的正弦/余弦波的叠加。傅里叶变换就是将信号从其原始域（如时间域或空间域）转换到频率域的过程。

4.1.1 傅里叶级数与傅里叶变换

傅里叶级数 (Fourier Series) 用于分析周期性函数。它将一个周期信号分解为一系列具有不同频率、幅度和相位的正弦波。

傅里叶变换 (Fourier Transform, FT) 将傅里叶级数的概念推广到非周期性函数。

一维连续傅里叶变换对：

正变换 (Forward Transform)：从空间域 $f(x)$ 到频率域 $F(u)$

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

反变换 (Inverse Transform)：从频率域 $F(u)$ 回到空间域 $f(x)$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

经典范例：空间域中的矩形函数 (rect) 经过傅里叶变换后，在频率域中会得到一个 sinc 函数。

4.2 采样定理与混叠 (The Sampling Theorem and Aliasing)

在将连续信号转换为数字信号时，采样是必不可少的一步，而傅里叶变换揭示了采样的本质后果。

4.2.1 采样的频率域效应

- 在空间域对连续函数 $f(t)$ 以间隔 ΔT 进行采样，等效于用一个脉冲串 (impulse train) 与 $f(t)$ 相乘。
- 这一操作在频率域的效果是，将原始连续信号的频谱 $F(\mu)$ 以 $\frac{1}{\Delta T}$ 为周期进行无限复制。
- 变换后的频谱 $\tilde{F}(\mu)$ 是原始频谱 $F(\mu)$ 无限平移叠加的结果：

$$\tilde{F}(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F\left(\mu - \frac{n}{\Delta T}\right)$$

4.2.2 采样定理 (Nyquist Criterion)

为了能够从采样后的数字信号中无损地恢复出原始连续信号，必须避免复制的频谱发生重叠。

奈奎斯特准则：采样频率必须至少是原始信号最高频率的两倍：

$$\frac{1}{\Delta T} \geq 2\mu_{max}$$

4.2.3 混叠 (Aliasing)

- 如果采样频率不满足奈奎斯特准则（即欠采样, Under-sampling），复制的频谱之间会发生重叠。
- 这种重叠会导致高频信息”伪装”成低频信息，使得原始信号无法被准确恢复，这种现象就是混叠。
- **如何避免：**在对图像进行缩减（降采样）之前，先用低通滤波器进行平滑处理，可以有效减少高频成分，从而减轻混叠现象。

混叠现象在图像处理中的典型表现就是在频域滤波时不进行适当填充而产生的卷绕误差，这在第六章频域滤波中会详细讨论（参见图3）。

4.3 二维离散傅里叶变换 (2D-DFT)

对于计算机处理的数字图像（离散的二维函数），我们使用离散傅里叶变换。

4.3.1 2D-DFT 变换对

正变换 (Forward 2D-DFT):

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

反变换 (Inverse 2D-DFT):

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

4.3.2 图像频率域 (k 空间) 的理解

- **中心区域 (低频)：**代表了图像中缓慢变化的部分，如整体的对比度和亮度。
- **外围区域 (高频)：**代表了图像中剧烈变化的部分，如边缘和细节。
- **例如：**在磁共振成像 (MRI) 中，采集到的原始数据空间就称为 k 空间，它就是 MR 图像的傅里叶变换结果。

4.4 2D-DFT 的重要性质

掌握 2D-DFT 的性质对于在频率域中进行图像处理至关重要。

4.4.1 可分离性 (Separability)

二维 DFT 可以分解为两次一维 DFT 来计算。首先对图像的每一行进行一维 DFT，然后再对结果的每一列进行一维 DFT。这大大简化了计算。

4.4.2 平移与中心化 (Translation & Centering)

- 标准的 DFT 计算会将零频率分量 $F(0,0)$ 放在结果矩阵的左上角。
- 为了便于观察和设计滤波器，我们通常希望将零频点移到频谱图的中心。
- 实现方法：在进行傅里叶变换之前，先将输入图像 $f(x,y)$ 与 $(-1)^{x+y}$ 相乘，即可实现频谱的中心化。

4.4.3 共轭对称性 (Conjugate Symmetry)

- 对于实数输入（所有图像都是实数函数），其傅里叶变换结果具有共轭对称性：

$$F(u,v) = F^*(-u,-v)$$

- 这意味着其频谱（幅度谱）是中心对称的： $|F(u,v)| = |F(-u,-v)|$ 。
- 因此，k 空间中有一半的数据是冗余的，可以根据另一半计算出来。

4.4.4 卷积定理 (Convolution Theorem)

这是傅里叶变换在图像处理中最核心的性质之一。

- 空间域的卷积等同于频率域的乘积：

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v) \cdot H(u,v)$$

- 频率域的卷积等同于空间域的乘积：

$$f(x,y) \cdot h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

- 意义：对于大型的滤波核，在空间域直接进行卷积运算的计算量非常大 ($O(M^2K^2)$)。而通过 FFT 转换到频率域，只需进行一次乘法运算 ($O(M^2)$)，再通过逆 FFT 转回空间域，总计算成本要低得多。

4.5 快速傅里叶变换 (Fast Fourier Transform - FFT)

- FFT 并不是一种新的变换，而是计算 DFT 的一种高效算法。
- 它通过将一个 M 点的 DFT 分解为两个 K 点 ($M = 2K$) 的子问题（分别对偶数和奇数项进行变换）来递归地减少计算量。
- FFT 将 DFT 的计算复杂度从 $O(M^2)$ 显著降低到 $O(M \log_2 M)$ 。
- 对于一个 2048×2048 的图像，FFT 的计算速度比直接计算 DFT 快了数千倍。

4.6 小波变换简介 (Introduction to Wavelet Transform)

4.6.1 动机

傅里叶变换能完美地告诉我们信号中包含哪些频率，但完全丢失了这些频率出现的位置信息。

4.6.2 核心思想

小波变换 (WT) 使用一系列被称为”小波” (wavelets) 的局部化、具有衰减性的波形作为基函数，而不是傅里叶变换中无限延伸的正弦波。

4.6.3 优势

通过对小波基函数进行缩放 (Scaling) 和平移 (Shifting)，小波变换能够同时提供信号的时间和频率信息，实现了多分辨率分析。

4.6.4 离散小波变换 (DWT)

将图像分解为不同层次的子带。

分解 每一层分解都会产生一个近似分量（低频，LL）和三个细节分量（水平、垂直、对角线的高频，LH, HL, HH）。

应用

- 在图像去噪中，可以只对细节分量进行阈值处理，然后重建图像，从而在去除噪声的同时保留大部分图像结构。
- 在边缘检测中，细节分量本身就代表了图像的边缘信息。

FT/FS 回顾（太痛苦了）奈奎斯特定理和混叠 2D-DFT $((-1)^{x+y} \text{ FFT WT}$

5 频率域滤波

本章的核心是利用傅里叶变换，在频率域中对图像进行操作，以达到平滑、锐化或选择性增强/抑制特定信息的目标。

5.1 频域滤波基础 (Fundamentals)

5.1.1 空间域与频率域的对应关系

- **低频分量**：对应图像中强度缓慢变化的部分，如大面积的背景和物体的基本轮廓。频谱的中心区域代表低频。
- **高频分量**：对应图像中强度急剧变化的部分，如边缘、细节和噪声。频谱的外围区域代表高频。
- **傅里叶变换对**：空间域的滤波器核 $h(x, y)$ 和频率域的传递函数 $H(u, v)$ 是一对傅里叶变换对。

根据尺度变换性质 $f(ax, by) \Leftrightarrow \frac{1}{|ab|} F(\frac{u}{a}, \frac{v}{b})$ ，一个在空间域上宽的滤波器（例如，大的高斯核）对应一个在频率域上窄的滤波器，反之亦然。

5.1.2 频域滤波的基本流程

频域滤波的核心是利用卷积定理 ($f * h \Leftrightarrow F \cdot H$)。其基本步骤如下：

步骤 1：预处理 - 填充 (Padding)

- **目的**：避免由 DFT 的周期性假设引起的卷绕误差 (wraparound error)。
- **方法**：将大小为 $M \times N$ 的原图填充到 $P \times Q$ （通常取 $P = 2M, Q = 2N$ ）。



图 3: 卷绕误差示例：左图显示了不进行填充时产生的卷绕误差效果，右图显示了正确填充后的结果。卷绕误差是由于 DFT 的周期性假设导致的图像边界信息错误复制造成的。

步骤 2: 预处理 - 中心化 (Centering)

- 目的: 将傅里叶变换的原点从左上角移动到图像中心, 便于滤波器设计和理解。
- 方法: 将填充后的图像 $f_p(x, y)$ 乘以 $(-1)^{x+y}$ 。

步骤 3: 傅里叶变换 (DFT) 对预处理后的图像进行 DFT, 得到其频谱 $F(u, v)$ 。

步骤 4: 滤波 (Filtering)

- 设计一个尺寸同样为 $P \times Q$ 且中心化的滤波器传递函数 $H(u, v)$ 。
- 将滤波器与图像频谱逐点相乘: $G(u, v) = H(u, v) \cdot F(u, v)$ 。

步骤 5: 反傅里叶变换 (IDFT) 对滤波后的结果 $G(u, v)$ 进行 IDFT, 得到处理后的空间域图像。

步骤 6: 后处理 取 IDFT 结果的实部, 并再次乘以 $(-1)^{x+y}$ 以抵消第二步的中心化操作。

步骤 7: 裁剪 (Cropping) 从左上角裁剪出 $M \times N$ 大小的区域, 得到最终的滤波后图像。

5.2 频域低通滤波器 (Low-Pass Filters)

低通滤波器用于平滑图像和去除高频噪声。主要有以下三种:

5.2.1 理想低通滤波器 (Ideal Low-Pass Filter, ILPF)

原理 像一堵“砖墙”, 在截止频率 D_0 内, 所有频率分量都无衰减地通过; 在 D_0 外, 所有频率分量都完全被阻断。

传递函数

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

其中 $D(u, v) = \sqrt{(u - u_0)^2 + (v - v_0)^2}$ 是点 (u, v) 到频谱中心的距离。

缺点 其在频域的急剧变化 (矩形函数) 对应于空间域的 sinc 函数, 这会导致严重的振铃效应 (ringing artifacts)。

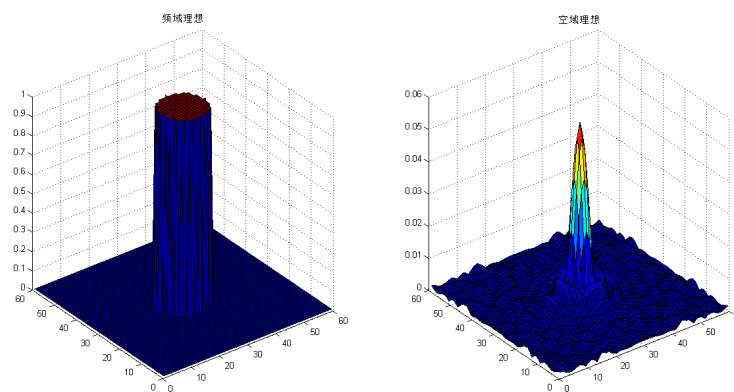


图 4: 理想低通滤波器: 左图为频域传递函数（圆形截止特性），右图为滤波后的图像效果



图 5: 振铃效应示例: 图像处理中, 对一幅图像进行滤波处理, 若选用的频域滤波器具有陡峭的变化, 则会使滤波图像产生”振铃”。所谓”振铃”, 就是指输出图像的灰度剧烈变化处产生的震荡, 就好像钟被敲击后产生的空气震荡。

例题 1: 理想低通滤波器设计与计算 设计一个理想低通滤波器对 64×64 的图像进行滤波, 已知图像的频谱中心位于 $(32, 32)$, 要求保留距离中心 30 个像素范围内的所有频率分量。

题目要求:

1. 写出滤波器的传递函数表达式
2. 计算点 $(40, 32)$ 处的传递函数值
3. 计算点 $(50, 45)$ 处的传递函数值
4. 如果原图某点的频谱值为 $F(25, 35) = 100 + 50j$, 求滤波后该点的频谱值

解:

第 1 步: 传递函数表达式

$$H(u, v) = \begin{cases} 1, & \text{if } D(u, v) \leq 30 \\ 0, & \text{if } D(u, v) > 30 \end{cases}$$

其中 $D(u, v) = \sqrt{(u - 32)^2 + (v - 32)^2}$

第 2 步：计算 $H(40, 32)$ $D(40, 32) = \sqrt{(40 - 32)^2 + (32 - 32)^2} = \sqrt{64} = 8 < 30$

因此 $H(40, 32) = 1$

第 3 步：计算 $H(50, 45)$ $D(50, 45) = \sqrt{(50 - 32)^2 + (45 - 32)^2} = \sqrt{324 + 169} = \sqrt{493} \approx 22.2 < 30$

因此 $H(50, 45) = 1$

第 4 步：滤波后的频谱值先计算 $D(25, 35) = \sqrt{(25 - 32)^2 + (35 - 32)^2} = \sqrt{49 + 9} = \sqrt{58} \approx 7.6 < 30$

因此 $H(25, 35) = 1$

滤波后： $G(25, 35) = H(25, 35) \cdot F(25, 35) = 1 \cdot (100 + 50j) = 100 + 50j$

5.2.2 巴特沃斯低通滤波器 (Butterworth Low-Pass Filter, BLPF)

原理 提供一个从通带到阻带的平滑过渡，而不是像 ILPF 那样急剧变化。

传递函数

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$$

其中 n 是滤波器的阶数，控制着过渡带的陡峭程度。

优点 振铃效应比 ILPF 轻微得多，尤其是当阶数 n 较低时（如 $n = 2$ ）。

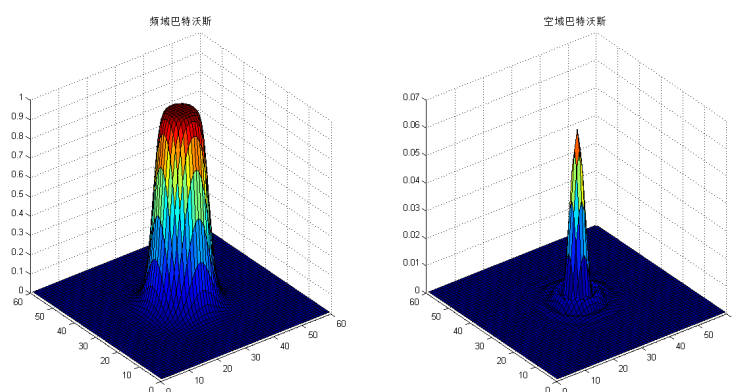


图 6: 巴特沃斯低通滤波器：左图为频域传递函数（平滑过渡特性），右图为滤波后的图像效果，相比理想低通滤波器振铃效应明显减轻

例题 2：巴特沃斯低通滤波器计算 设计一个 2 阶巴特沃斯低通滤波器，截止频率 $D_0 = 20$ ，对 128×128 图像进行滤波。图像频谱中心位于 $(64, 64)$ 。

题目要求：

1. 写出该滤波器的传递函数

2. 计算点 (64, 84) 处的传递函数值
3. 计算点 (84, 84) 处的传递函数值
4. 在截止频率处的传递函数值是多少?
5. 如果某点频谱值为 $F(70, 70) = 80 - 60j$, 计算滤波后的值

解:

第 1 步: 传递函数

$$H(u, v) = \frac{1}{1 + [D(u, v)/20]^4}$$

其中 $D(u, v) = \sqrt{(u - 64)^2 + (v - 64)^2}$

第 2 步: 计算 $H(64, 84)$ $D(64, 84) = \sqrt{(64 - 64)^2 + (84 - 64)^2} = 20$

$$H(64, 84) = \frac{1}{1 + (20/20)^4} = \frac{1}{1 + 1} = 0.5$$

第 3 步: 计算 $H(84, 84)$ $D(84, 84) = \sqrt{(84 - 64)^2 + (84 - 64)^2} = \sqrt{800} = 20\sqrt{2} \approx 28.28$

$$H(84, 84) = \frac{1}{1 + (28.28/20)^4} = \frac{1}{1 + (1.414)^4} = \frac{1}{1 + 4} = 0.2$$

第 4 步: 截止频率处的值在 $D = D_0 = 20$ 处, $H = \frac{1}{1 + 1} = 0.5$

第 5 步: 滤波计算 $D(70, 70) = \sqrt{(70 - 64)^2 + (70 - 64)^2} = \sqrt{72} = 6\sqrt{2} \approx 8.49$

$$H(70, 70) = \frac{1}{1 + (8.49/20)^4} = \frac{1}{1 + (0.425)^4} = \frac{1}{1 + 0.033} = 0.968$$

$$G(70, 70) = 0.968 \times (80 - 60j) = 77.44 - 58.08j$$

5.2.3 高斯低通滤波器 (Gaussian Low-Pass Filter, GLPF)

原理 传递函数本身是一个高斯函数。

传递函数

$$H(u, v) = e^{-D^2(u, v)/(2D_0^2)}$$

核心优点 高斯函数的傅里叶变换仍然是高斯函数, 这意味着它在空间域和频率域都是平滑的, 因此完全没有振铃效应。

应用场景 在不允许任何振铃效应的场合 (如医学诊断图像) 是首选。

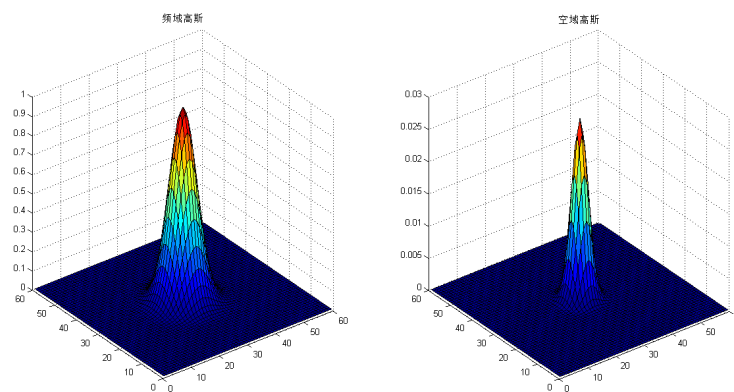


图 7: 高斯低通滤波器: 左图为频域传递函数 (高斯分布特性), 右图为滤波后的图像效果, 完全无振铃效应

三种低通滤波器对比

- 在截止频率处, 高斯滤波器不如巴特沃斯滤波器陡峭
- 当需要在截止频率附近对低频和高频进行严格控制时, 应选择巴特沃斯滤波器
- 当不允许振铃效应时 (如医学诊断中), 应选择高斯滤波器

例题 3: 高斯低通滤波器计算 设计一个高斯低通滤波器, 标准差 $D_0 = 15$, 对 256×256 图像进行滤波。频谱中心位于 $(128, 128)$ 。

题目要求:

1. 写出该滤波器的传递函数
2. 计算点 $(128, 143)$ 处的传递函数值
3. 计算点 $(143, 143)$ 处的传递函数值
4. 距离中心多远处传递函数值降为 $1/e \approx 0.368$?
5. 如果某点频谱值为 $F(135, 140) = 120 + 90j$, 计算滤波后的频谱值和幅度

解:

第 1 步: 传递函数

$$H(u, v) = e^{-D^2(u, v)/(2 \times 15^2)} = e^{-D^2(u, v)/450}$$

其中 $D(u, v) = \sqrt{(u - 128)^2 + (v - 128)^2}$

第 2 步: 计算 $H(128, 143)$ $D(128, 143) = \sqrt{(128 - 128)^2 + (143 - 128)^2} = 15$

$$H(128, 143) = e^{-15^2/450} = e^{-225/450} = e^{-0.5} \approx 0.606$$

第 3 步：计算 $H(143, 143)$ $D(143, 143) = \sqrt{(143 - 128)^2 + (143 - 128)^2} = 15\sqrt{2} \approx 21.21$

$H(143, 143) = e^{-(21.21)^2/450} = e^{-450/450} = e^{-1} \approx 0.368$

第 4 步： $H = 1/e$ 时的距离当 $H = 1/e$ 时： $e^{-D^2/450} = 1/e$

即 $-D^2/450 = -1$ ，所以 $D^2 = 450$ ， $D = \sqrt{450} = 15\sqrt{2} \approx 21.21$

第 5 步：滤波计算 $D(135, 140) = \sqrt{(135 - 128)^2 + (140 - 128)^2} = \sqrt{49 + 144} = \sqrt{193} \approx 13.89$

$H(135, 140) = e^{-193/450} = e^{-0.429} \approx 0.651$

$G(135, 140) = 0.651 \times (120 + 90j) = 78.12 + 58.59j$

幅度： $|G(135, 140)| = \sqrt{78.12^2 + 58.59^2} = \sqrt{9532.4} \approx 97.6$

5.3 频域高通滤波器 (High-Pass Filters)

高通滤波器用于锐化图像，通过增强边缘和细节等高频分量来实现。

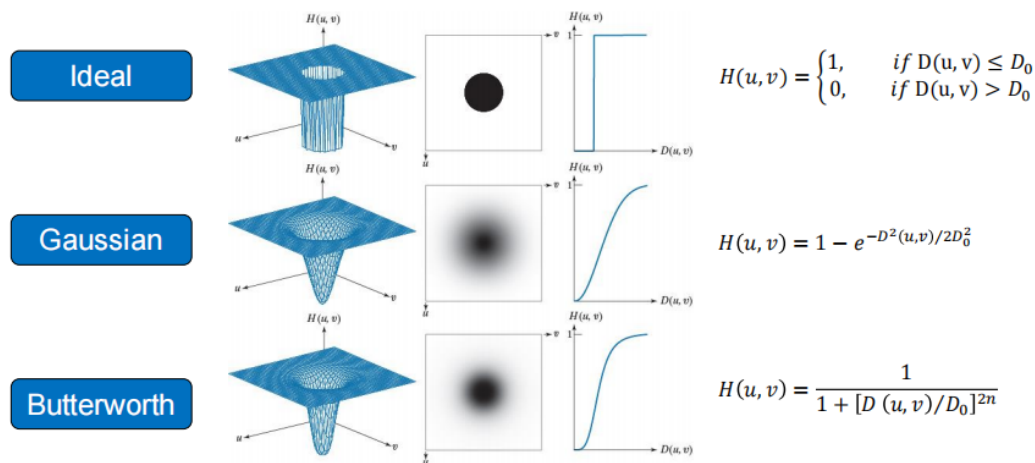


图 8: 频域高通滤波器总览：展示了不同类型高通滤波器（理想、巴特沃斯、高斯）的频域特性和对图像的滤波效果，从左到右分别是原图、理想高通、巴特沃斯高通和高斯高通的结果。

5.3.1 基本关系

高通滤波器可以由对应的低通滤波器导出：

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

这个关系意味着高通滤波器衰减低频，通过高频。

5.3.2 主要类型

同样包括理想 (IHPF)、巴特沃斯 (BHPF) 和高斯 (GHPF) 三种类型，其传递函数都是由对应的低通形式通过 $1 - H_{lp}$ 得到。

例题 4：理想高通滤波器计算 设计一个理想高通滤波器，截止频率 $D_0 = 25$ ，对 128×128 图像进行滤波。

题目要求：

1. 写出理想高通滤波器的传递函数
2. 计算点 $(64, 89)$ 处的传递函数值（中心在 $(64, 64)$ ）
3. 计算点 $(84, 64)$ 处的传递函数值
4. 如果某点的低通滤波器值为 $H_{lp}(70, 80) = 0.8$ ，求对应的高通滤波器值
5. 对于频谱值 $F(50, 50) = 60 + 80j$ ，计算高通滤波后的结果

解：

第 1 步：理想高通滤波器传递函数

$$H_{hp}(u, v) = \begin{cases} 0, & \text{if } D(u, v) \leq 25 \\ 1, & \text{if } D(u, v) > 25 \end{cases}$$

第 2 步：计算 $H_{hp}(64, 89)$ $D(64, 89) = \sqrt{(64 - 64)^2 + (89 - 64)^2} = 25$

由于 $D = 25 = D_0$ ，根据定义 $H_{hp}(64, 89) = 0$

第 3 步：计算 $H_{hp}(84, 64)$ $D(84, 64) = \sqrt{(84 - 64)^2 + (64 - 64)^2} = 20 < 25$

因此 $H_{hp}(84, 64) = 0$

第 4 步：由低通求高通 $H_{hp}(70, 80) = 1 - H_{lp}(70, 80) = 1 - 0.8 = 0.2$

第 5 步：滤波计算 $D(50, 50) = \sqrt{(50 - 64)^2 + (50 - 64)^2} = \sqrt{196 + 196} = 14\sqrt{2} \approx 19.8 < 25$

因此 $H_{hp}(50, 50) = 0$

$G(50, 50) = 0 \times (60 + 80j) = 0 + 0j$

5.3.3 频域拉普拉斯滤波器

算子表达式 拉普拉斯算子在频率域的表达式为：

$$H(u, v) = -4\pi^2 D^2(u, v)$$

拉普拉斯增强 拉普拉斯增强在频率域的实现为：

$$g(x, y) = \mathcal{F}^{-1}\{[1 - c \cdot H_{Laplacian}(u, v)]F(u, v)\}$$

这里通常将 c 合并到 H 中。

例题 5：频域拉普拉斯滤波器计算 使用频域拉普拉斯滤波器对 64×64 图像进行锐化，频谱中心位于 $(32, 32)$ 。

题目要求：

1. 写出频域拉普拉斯算子的传递函数
2. 计算点 $(40, 40)$ 处的拉普拉斯传递函数值
3. 写出拉普拉斯增强的传递函数（增强系数 $c = 1$ ）
4. 如果原频谱值为 $F(35, 45) = 100 + 50j$ ，计算拉普拉斯增强后的频谱值
5. 计算该点对应的空间域坐标的拉普拉斯增强结果（假设逆变换后实部为 150）

解：

第 1 步：频域拉普拉斯传递函数

$$H_{Laplacian}(u, v) = -4\pi^2 D^2(u, v)$$

其中 $D(u, v) = \sqrt{(u - 32)^2 + (v - 32)^2}$

第 2 步：计算 $H_{Laplacian}(40, 40)$ $D(40, 40) = \sqrt{(40 - 32)^2 + (40 - 32)^2} = \sqrt{128} = 8\sqrt{2}$

$$H_{Laplacian}(40, 40) = -4\pi^2 \times 128 = -4 \times 9.87 \times 128 = -5052.16$$

第 3 步：拉普拉斯增强传递函数

$$H_{enhance}(u, v) = 1 - H_{Laplacian}(u, v) = 1 + 4\pi^2 D^2(u, v)$$

第 4 步：计算增强后的频谱值 $D(35, 45) = \sqrt{(35 - 32)^2 + (45 - 32)^2} = \sqrt{9 + 169} = \sqrt{178} \approx 13.34$

$$H_{enhance}(35, 45) = 1 + 4\pi^2 \times 178 = 1 + 4 \times 9.87 \times 178 = 1 + 7026.72 = 7027.72$$

$$G(35, 45) = 7027.72 \times (100 + 50j) = 702772 + 351386j$$

第 5 步：空间域结果假设逆傅里叶变换后，该点的实部为 150，则拉普拉斯增强后的像素值为 150。

由于拉普拉斯增强公式： $g(x, y) = f(x, y) - \nabla^2 f(x, y)$

如果原像素值为 f ，拉普拉斯值为 L ，则： $150 = f - L$

这表明经过拉普拉斯增强后，该像素的边缘信息得到了增强。

5.3.4 高频增强滤波 (High-Frequency-Emphasis Filtering)

这是对非锐化掩模的一种推广。它在保留低频分量的同时，通过一个增益因子 k 来增强高频分量。

公式形式：

$$H_{hfe}(u, v) = a + b \cdot H_{hp}(u, v)$$

其中 a 是偏移量（保留低频）， b 是增强倍数。

例题 6：高频增强滤波器计算 设计一个高频增强滤波器，使用高斯高通滤波器作为基础，参数 $a = 0.5$ ， $b = 2.0$ ， $D_0 = 30$ 。

题目要求：

1. 写出高频增强滤波器的传递函数
2. 计算点 $(64, 94)$ 处的传递函数值（中心在 $(64, 64)$ ）
3. 比较该点的高斯高通值和高频增强值
4. 如果频谱值为 $F(80, 80) = 90 + 120j$ ，计算滤波后的结果
5. 分析 a 和 b 参数对滤波效果的影响

解：

第 1 步： 高频增强滤波器传递函数先计算高斯高通： $H_{hp}(u, v) = 1 - e^{-D^2(u, v)/(2D_0^2)}$

高频增强： $H_{hfe}(u, v) = 0.5 + 2.0 \times H_{hp}(u, v)$

$H_{hfe}(u, v) = 0.5 + 2.0 \times [1 - e^{-D^2(u, v)/1800}]$

第 2 步： 计算 $H_{hfe}(64, 94)$ $D(64, 94) = \sqrt{(64 - 64)^2 + (94 - 64)^2} = 30$

$H_{hp}(64, 94) = 1 - e^{-30^2/1800} = 1 - e^{-0.5} = 1 - 0.606 = 0.394$

$H_{hfe}(64, 94) = 0.5 + 2.0 \times 0.394 = 0.5 + 0.788 = 1.288$

第 3 步： 对比分析高斯高通值： $H_{hp} = 0.394$ （仅保留高频）高频增强值： $H_{hfe} = 1.288$ （保留所有频率并增强高频）

第 4 步： 滤波计算 $D(80, 80) = \sqrt{(80 - 64)^2 + (80 - 64)^2} = 16\sqrt{2} \approx 22.63$

$H_{hp}(80, 80) = 1 - e^{-(22.63)^2/1800} = 1 - e^{-0.284} = 1 - 0.753 = 0.247$

$H_{hfe}(80, 80) = 0.5 + 2.0 \times 0.247 = 0.5 + 0.494 = 0.994$

$G(80, 80) = 0.994 \times (90 + 120j) = 89.46 + 119.28j$

第 5 步： 参数影响分析

- $a = 0.5$ ：保留 50% 的低频成分，避免图像过暗
- $b = 2.0$ ：将高频成分放大 2 倍，增强边缘和细节
- 总效果：既保持图像的基本亮度，又显著增强了细节

5.4 选择性滤波 (Selective Filtering)

选择性滤波用于衰减或通过特定的频率范围或点。

5.4.1 带阻/带通滤波器 (Band-reject/Band-pass)

带阻滤波器 用于去除一个以原点为中心的圆形频率带内的信号。

带通滤波器 则相反，只允许该频率带内的信号通过，其传递函数为：

$$H_{BP}(u, v) = 1 - H_{BR}(u, v)$$

5.4.2 陷波滤波器 (Notch Filter)

用途 专门用于去除特定频率点的噪声，特别是那些由外部干扰产生的周期性噪声（如电网干扰、屏幕刷新率干扰等）。

特征 在频谱图上，周期性噪声通常表现为除原点外的一对或多对对称的亮点。

实现 陷波滤波器通过在这些亮点位置上设置“陷阱”（通常是乘以一个中心移到噪声点的局部高通滤波器）来精确地去除这些噪声点。

陷波滤波器的设计 对于需要去除频率点 (u_k, v_k) 和 $(-u_k, -v_k)$ 的噪声，陷波滤波器可以设计为：

$$H_{notch}(u, v) = \prod_k [1 - H_k(u, v)] [1 - H_{-k}(u, v)]$$

其中 $H_k(u, v)$ 是以点 (u_k, v_k) 为中心的局部低通滤波器。

应用实例

- 去除由于电力线产生的 50Hz 或 60Hz 噪声
- 消除由显示器刷新频率引起的周期性干扰
- 移除由数字化设备采样不当造成的规律性伪影

例题 7：陷波滤波器设计与计算 设计一个陷波滤波器去除 128×128 图像中的周期性噪声，噪声在频谱中表现为位于 $(90, 90)$ 和 $(38, 38)$ 的两个亮点（频谱中心在 $(64, 64)$ ）。

题目要求：

1. 确定需要去除的所有噪声点位置
2. 设计巴特沃斯陷波滤波器，截止距离 $D_0 = 5$ ，阶数 $n = 2$
3. 计算点 $(90, 90)$ 处的传递函数值
4. 计算点 $(88, 92)$ 处的传递函数值
5. 如果噪声点 $(90, 90)$ 的频谱值为 $F = 500 + 300j$ ，计算滤波后的值

解：

第 1 步：确定所有噪声点由于共轭对称性，噪声点包括：

- (90, 90) 和其对称点 (38, 38) ($128 - 90 = 38$)
- 实际上这两个点关于中心 (64, 64) 对称

第 2 步：巴特沃斯陷波滤波器设计对于噪声点 (u_k, v_k) ，陷波滤波器为：

$$H_{notch}(u, v) = \prod_k \frac{1}{1 + [D_0/D_k(u, v)]^{2n}} \times \frac{1}{1 + [D_0/D_{-k}(u, v)]^{2n}}$$

其中： $D_1(u, v) = \sqrt{(u - 90)^2 + (v - 90)^2}$ $D_2(u, v) = \sqrt{(u - 38)^2 + (v - 38)^2}$

$$H_{notch}(u, v) = \frac{1}{1 + [5/D_1(u, v)]^4} \times \frac{1}{1 + [5/D_2(u, v)]^4}$$

第 3 步：计算 $H_{notch}(90, 90)$ $D_1(90, 90) = 0$, $D_2(90, 90) = \sqrt{(90 - 38)^2 + (90 - 38)^2} = 52\sqrt{2} \approx 73.54$

当 $D_1 = 0$ 时，第一项为 0（陷波效果） $H_{notch}(90, 90) = 0 \times \frac{1}{1 + (5/73.54)^4} = 0$

第 4 步：计算 $H_{notch}(88, 92)$ $D_1(88, 92) = \sqrt{(88 - 90)^2 + (92 - 90)^2} = \sqrt{4 + 4} = 2\sqrt{2} \approx 2.83$ $D_2(88, 92) = \sqrt{(88 - 38)^2 + (92 - 38)^2} = \sqrt{2500 + 2916} = \sqrt{5416} \approx 73.56$

$$H_{notch}(88, 92) = \frac{1}{1 + (5/2.83)^4} \times \frac{1}{1 + (5/73.56)^4} = \frac{1}{1 + (1.767)^4} \times \frac{1}{1 + (0.068)^4} = \frac{1}{1 + 9.73} \times \frac{1}{1 + 0.000021} = 0.093 \times 1.0 = 0.093$$

第 5 步：滤波计算 $G(90, 90) = H_{notch}(90, 90) \times F(90, 90) = 0 \times (500 + 300j) = 0 + 0j$
噪声完全被去除。

整个频率滤波的流程：填充-> 中心化->DFT-> 滤波->IDFT-> 后处理-> 裁剪低通滤波：理想、巴特沃斯、高斯高通滤波：理想、巴特沃斯、高斯、Laplacian 带通：Band 还有 Notch !
对应第三章课后习题

6 图像复原

本章的核心是理解图像是如何“变坏”（退化）的，并利用这些知识来尽可能地恢复出原始的、未被破坏的图像。

6.1 图像复原基础 (Image Restoration Fundamentals)

6.1.1 图像复原 vs. 图像增强

这是一个关键的区别：

- **图像增强 (Enhancement)**: 一个主观的过程，目的是让图像在视觉上更好看，或更适合某个特定应用，没有统一的评判标准。
- **图像复原 (Restoration)**: 一个客观的过程，它试图根据已知的图像退化模型，来估计出最接近原始、无损图像的结果。

6.1.2 图像退化/复原模型

这是整个图像复原领域的理论基础。一个退化的图像 $g(x, y)$ 可以被建模为原始图像 $f(x, y)$ 经过一个退化系统 H ，并叠加上一个加性噪声 $\eta(x, y)$ 后的结果。

通用模型

$$g(x, y) = H[f(x, y)] + \eta(x, y)$$

线性空间不变系统 对于线性和空间不变的系统，退化过程可以简化为卷积操作：

空间域模型：

$$g(x, y) = (h * f)(x, y) + \eta(x, y)$$

其中， $h(x, y)$ 是退化函数，也称为点扩散函数 (PSF)。

频率域模型：

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

这个模型清晰地展示了退化（乘积）和噪声（加法）的影响。

图像复原的目标就是从已知的 $g(x, y)$ 和对 $H(u, v)$ 、 $\eta(x, y)$ 的了解，来得到一个对 $f(x, y)$ 的最佳估计 $\hat{f}(x, y)$ 。

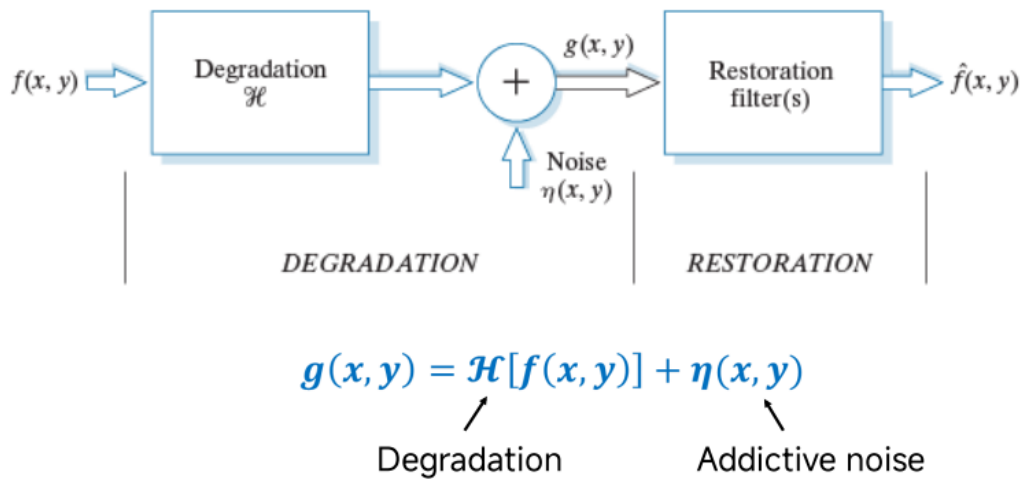


图 9: 图像退化/复原模型: 展示了原始图像经过退化系统 \mathcal{H} 和噪声 $\eta(x, y)$ 后变成退化图像, 以及复原过程中通过估计退化函数来恢复原始图像的完整流程

6.2 噪声模型 (Noise Models)

噪声主要在图像的采集和传输过程中产生。了解不同噪声的统计特性是选择合适复原滤波器的前提。

6.2.1 常见噪声类型

表 5: 常见噪声类型及其特性详解

噪声类型	概率密度函数	参数说明	特性与应用
高斯噪声 (Gaussian)	$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$	μ : 均值 σ : 标准差	特性: 对称钟形分布, 数学处理简单 来源: 电子器件热噪声、放大器噪声 效果: 图像整体模糊
瑞利噪声 (Rayleigh)	$p(z) = \frac{z}{b^2} e^{-\frac{z^2}{2b^2}}, z \geq 0$	b : 尺度参数 均值: $b\sqrt{\pi/2}$	特性: 右偏分布, 只有正值 来源: 雷达波散射、超声回波 效果: 影像偏亮, 对比度降低
爱尔兰噪声 (Erlang/Gamma)	$p(z) = \frac{a^b z^{b-1}}{(b-1)!} e^{-az}, z \geq 0$	$a > 0$: 尺度参数 b : 正整数 (形状) 均值: b/a	特性: 形状可调的右偏分布 来源: 激光成像、光子计数 效果: 低光照下的量子噪声
指数噪声 (Exponential)	$p(z) = ae^{-az}, z \geq 0$	$a > 0$: 衰减率 均值: $1/a$	特性: 爱尔兰噪声特例 ($b = 1$) 来源: 衰减过程、寿命分布 效果: 极度右偏, 突出低灰度
均匀噪声 (Uniform)	$p(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & \text{其他} \end{cases}$	a, b : 区间端点 均值: $(a+b)/2$	特性: 矩形分布, 等概率 来源: A/D 转换量化误差 效果: 灰度级轻微随机化
椒盐噪声 (Impulse)	$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{其他} \end{cases}$	$P_a + P_b \leq 1$ a : 胡椒值 (暗) b : 盐粒值 (亮)	特性: 脉冲性, 只在固定灰度值 来源: 传输错误、开关故障 效果: 随机黑白点, 破坏图像结构

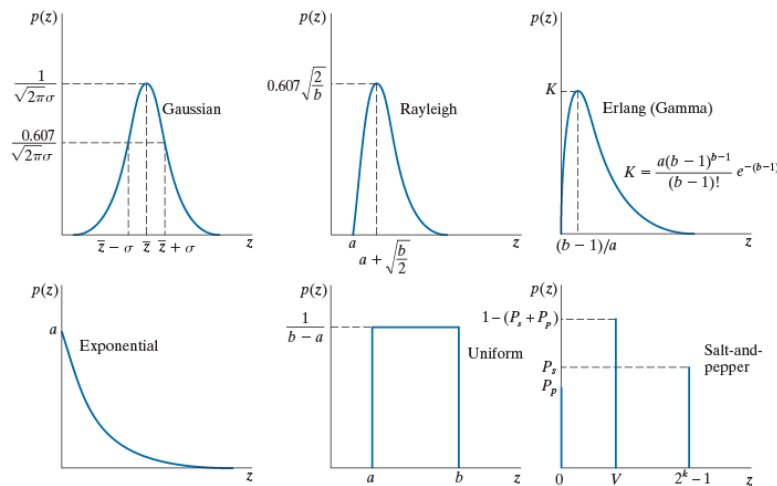


图 10: 常见噪声类型的概率密度函数

噪声类型详细解释 爱尔兰噪声 (Erlang Noise): 爱尔兰噪声是伽马分布的特殊情况，当形状参数为正整数时称为爱尔兰分布。它描述了 b 个独立指数随机变量的和的分布。在图像处理中，常见于：

- **激光成像系统:** 光子到达时间间隔的累积效应
- **医学成像:** 核素衰变过程中的计数统计
- **天文观测:** 暗电流和热噪声的复合效应

瑞利噪声的物理意义: 瑞利分布描述了两个独立高斯随机变量平方和的平方根分布，在成像中表现为：

- **雷达图像:** 目标回波信号的幅度分布
- **超声成像:** 组织散射回波的包络检波
- **MRI 图像:** 噪声背景区域的信号幅度

椒盐噪声的特殊性: 椒盐噪声是唯一的双值脉冲噪声，具有以下特点：

- **盐粒噪声:** 随机出现的亮点 ($z = z_{max}$)
- **胡椒噪声:** 随机出现的暗点 ($z = z_{min}$)
- **复合椒盐噪声:** 同时包含亮暗脉冲

周期性噪声 (Periodic Noise): 周期性噪声不同于随机噪声，它具有确定的空间频率成分：

$$\eta(x, y) = \sum_i A_i \sin(2\pi u_i x + 2\pi v_i y + \phi_i)$$

- **来源：**电磁干扰、电源纹波、扫描设备的机械振动
- **特征：**傅里叶频谱中出现除原点外的对称亮点
- **去除方法：**陷波滤波器、带阻滤波器
- **实例：**CRT 显示器的行扫描干扰、荧光灯频闪效应

6.2.2 噪声参数的估计

噪声参数的准确估计是选择合适复原方法的关键前提。

周期性噪声估计 通过观察图像的傅里叶频谱来识别。周期性噪声在频谱中表现为除原点外的对称亮点。

随机噪声估计 对于其他类型的随机噪声，采用以下标准流程：

1. **选择均匀区域：**在图像中选取灰度值基本恒定的小块区域（平坦区），这些区域应该尽可能不包含边缘和纹理信息。
2. **计算直方图：**对选定区域内的像素灰度值计算直方图，得到噪声的统计分布。
3. **计算统计参数：**计算该区域灰度值的均值和方差：

$$\mu = \frac{1}{N} \sum_{i=1}^N z_i$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (z_i - \mu)^2$$

4. **PDF 匹配：**根据直方图的形状特征，识别最接近的概率密度函数类型：

- 对称钟形 → 高斯噪声
- 右偏分布 → 瑞利或指数噪声
- 均匀分布 → 均匀噪声
- 双峰分布 → 椒盐噪声

5. **参数求解：**利用计算得到的均值和方差，求解所选 PDF 模型的具体参数。例如，对于高斯噪声： $\mu_{noise} = \mu$ ， $\sigma_{noise} = \sigma$ 。

6.3 仅有噪声的图像复原 (Spatial Filtering Restoration)

当图像退化只有噪声时（即 H 是一个单位冲击），模型简化为：

$$g(x, y) = f(x, y) + \eta(x, y)$$

此时，主要使用空间滤波器进行去噪。

6.3.1 均值滤波器 (Mean Filters)

均值滤波器是一类基于统计平均的线性滤波器，通过不同的平均方式来处理各种类型的噪声。根据 PPT 所示，主要有四种类型：

1. 算术均值滤波器 (Arithmetic Mean Filter) 数学表达式：

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

工作原理：

- 对邻域内所有像素值进行简单的算术平均
- 是最基础、最常用的平滑滤波器
- 基于大数定律：随机噪声的均值趋于零

适用场景：

- 高斯噪声：效果最佳，因为高斯噪声均值为零
- 均匀噪声：同样有效
- 一般随机噪声：只要噪声均值接近零

优缺点：

- 优点：实现简单，计算量小，对高斯噪声效果好
- 缺点：会导致图像模糊，特别是边缘细节丢失

2. 几何均值滤波器 (Geometric Mean Filter) 数学表达式：

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

工作原理：

- 计算邻域内所有像素值的几何平均
- 几何平均 = 所有数值乘积的 n 次方根
- 相比算术平均，对极值不敏感

特性分析：

- 平滑效果：介于算术均值和中值滤波之间
- 细节保持：比算术均值更好地保留图像细节
- 数学特性：几何平均 \geq 算术平均（当且仅当所有值相等时等号成立）

适用场景：

- 需要平滑噪声但要保留更多细节的场合
- 处理乘性噪声（如斑点噪声）

3. 谐波均值滤波器 (Harmonic Mean Filter) 数学表达式：

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}}$$

工作原理：

- 计算邻域内所有像素值倒数的算术平均的倒数
- 谐波平均对小值敏感，对大值不敏感
- 数学特性：谐波平均 \leq 几何平均 \leq 算术平均

专门应用：

- 盐粒噪声：特别有效，因为谐波均值趋向于较小值
- 高亮度脉冲噪声：能有效抑制
- 重要限制：对胡椒噪声（暗点）无效，甚至会产生负面效果

注意事项：

- 当邻域内有零值时，整个结果为零
- 主要用于去除亮点噪声，不适用于暗点噪声

4. 逆谐波均值滤波器 (Contra-harmonic Mean Filter) 数学表达式:

$$\hat{f}(x,y) = \frac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^Q}$$

其中 Q 称为滤波器的阶数 (order)。

工作原理:

- 通过调整阶数 Q 来控制滤波器的行为
- Q 的符号决定了滤波器偏向高值还是低值
- 是最灵活的均值滤波器

阶数效应详解:

- $Q > 0$: 滤波器偏向较大的像素值
 - 适合去除胡椒噪声 (暗点)
 - Q 越大, 偏向性越强
- $Q < 0$: 滤波器偏向较小的像素值
 - 适合去除盐粒噪声 (亮点)
 - Q 越小 (绝对值越大), 偏向性越强
- $Q = 0$: 退化为算术均值滤波器
- $Q = -1$: 退化为谐波均值滤波器

极限情况:

- $Q \rightarrow +\infty$: 最大值滤波器 (选择邻域内最大值)
- $Q \rightarrow -\infty$: 最小值滤波器 (选择邻域内最小值)

表 6: 四种均值滤波器特性比较

滤波器类型	平均方式	适用噪声	细节保持	特殊优势
算术均值	简单算术平均	高斯噪声、均匀噪声	一般	实现简单, 通用性强
几何均值	几何平均	一般随机噪声	较好	保留细节, 对极值不敏感
谐波均值	谐波平均	盐粒噪声 (亮点)	较好	专门去除亮点噪声
逆谐波均值	加权幂平均	椒盐噪声 (可调)	可调	参数可调, 适应性强

四种均值滤波器的比较总结 选择原则:

1. **噪声类型识别**：首先确定图像中的主要噪声类型
2. **处理目标**：明确是要去噪还是要保留细节
3. **参数调优**：特别是逆谐波滤波器的阶数选择
4. **效果评估**：结合视觉效果和定量指标进行综合评价

6.3.2 排序统计滤波器 (Order-Statistics Filters)

排序统计滤波器是一类非线性滤波器，它们首先对邻域内的像素值进行排序，然后根据排序结果选择输出值。根据 PPT 所示，主要包括以下四种类型：

1. 中值滤波器 (Median Filter) 数学表达式：

$$\hat{f}(x, y) = \text{median}_{(s,t) \in S_{xy}} \{g(s, t)\}$$

工作原理：

- 将邻域内所有像素值按大小排序
- 选择排序后的中位数作为输出值
- 关键特性：原始像素值包含在计算中，但不直接影响结果

椒盐噪声处理的核心优势：

- 极值免疫性：中值对极值（椒盐噪声）不敏感
- 保边特性：能够很好地保留图像边缘和细节
- 迭代稳定性：可以多次应用而不会过度模糊

2. 最大值/最小值滤波器 (Max/Min Filter) 数学表达式：

$$\text{最大值滤波器: } \hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (16)$$

$$\text{最小值滤波器: } \hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (17)$$

专门应用：

- **最大值滤波器**：专门去除胡椒噪声（黑点）
 - 原理：选择邻域内最大值，自然排除了异常小的暗点
 - 副作用：会使图像整体变亮，细线可能断裂
- **最小值滤波器**：专门去除盐粒噪声（白点）
 - 原理：选择邻域内最小值，自然排除了异常大的亮点
 - 副作用：会使图像整体变暗，细线可能消失

3. 中点滤波器 (Midpoint Filter) 数学表达式:

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right]$$

工作原理:

- 结合排序统计和平均方法
- 计算邻域内最大值和最小值的平均
- 对随机分布噪声（如高斯噪声、均匀噪声）效果最佳

适用场景:

- 高斯噪声和均匀噪声
- 需要保持图像亮度平衡的场合
- 不适合椒盐噪声（会被极值影响）

4. Alpha-修剪均值滤波器 (Alpha-trimmed Mean Filter) 数学表达式:

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{i=d/2+1}^{mn-d/2} g_i(s, t)$$

其中 $g_i(s, t)$ 表示邻域内按升序排列的第 i 个像素值, d 为修剪的像素数量。

工作原理:

- 先对邻域像素值排序
- 去掉 $d/2$ 个最小值和 $d/2$ 个最大值
- 对剩余像素求算术平均

参数效应:

- $d = 0$: 退化为算术均值滤波器
- $d = mn - 1$: 退化为中值滤波器
- d 适中: 结合了均值和中值的优点

混合噪声处理:

- 特别适合处理高斯噪声和椒盐噪声的混合
- 通过修剪去除极值（椒盐噪声）
- 通过平均处理剩余的高斯噪声

为什么排序统计滤波器对椒盐噪声效果特别好？ 1. 椒盐噪声的特殊性质：

- 脉冲性：椒盐噪声表现为孤立的极值点
- 稀疏性：通常只有少数像素被污染
- 极值特性：盐粒噪声 = 最大灰度值，胡椒噪声 = 最小灰度值

2. 排序统计的天然优势：

- 极值鲁棒性：排序统计对极值变化不敏感
- 局部适应性：只要噪声点不占邻域多数，就能有效处理
- 非线性特性：能够处理线性滤波器无法解决的脉冲噪声

3. 具体处理机制分析：

以 3×3 邻域中值滤波为例：

- 噪声密度 $< 50\%$ ：中值必然是正常像素值
- 单个椒盐点：在 9 个值中排序第 5 位，不是极值
- 边缘保持：正常像素的中值仍然代表真实边缘特征

4. 数学证明（简化版）：

假设邻域内有 k 个椒盐噪声点，总共 n 个像素：

- 当 $k < n/2$ 时，中值滤波器输出必然是正常像素值
- 当 $k \geq n/2$ 时，需要增大邻域尺寸或使用 alpha-修剪滤波器

表 7: 排序统计滤波器特性比较

滤波器类型	选择策略	最佳噪声类型	椒盐噪声效果	副作用
中值滤波器	中位数	椒盐噪声	优秀	轻微模糊
最大值滤波器	最大值	胡椒噪声	单向有效	图像变亮
最小值滤波器	最小值	盐粒噪声	单向有效	图像变暗
中点滤波器	最值平均	高斯/均匀噪声	差	受极值影响
Alpha-修剪	修剪后平均	混合噪声	良好	参数敏感

排序统计滤波器比较总结 实际应用建议：

1. 纯椒盐噪声：首选中值滤波器
2. 单一类型脉冲噪声：使用对应的最值滤波器
3. 混合噪声：考虑 alpha-修剪均值滤波器
4. 高噪声密度：增大滤波器窗口尺寸
5. 边缘敏感应用：优先选择中值滤波器

6.3.3 自适应滤波器 (Adaptive Filters)

这类滤波器的性能优于固定滤波器，因为它们的行为会根据滤波器窗口内的图像局部统计特性（如均值、方差）进行调整。

1. 自适应局部降噪滤波器 (Adaptive Local Noise Reduction Filter, ALNRF) 核心思想：图像噪声的强度在整幅图中可能不是均匀的。平坦区域噪声可能更明显，而纹理复杂区域，噪声可能与细节本身难以区分。因此，降噪的强度应该是自适应的。

一句话总结：ALNRF 通过分析一个局部窗口内的统计特性（均值和方差），来估计该区域的噪声强度，并据此调整滤波强度，对每个像素进行”定制化”的降噪处理。

数学表达式：

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_{\eta}^2}{\sigma_L^2} [g(x, y) - m_L]$$

其中：

- σ_{η}^2 : 全局噪声方差
- σ_L^2 : 局部方差 = $\frac{1}{mn} \sum_{(s,t) \in S_{xy}} [g(s, t) - m_L]^2$
- m_L : 局部均值 = $\frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$

算法步骤：

1. **计算局部统计量：**对于以像素 (x, y) 为中心的局部窗口（如 5×5 或 7×7 ），计算窗口内所有像素的局部均值 m_L 和局部方差 σ_L^2 。
2. **估计噪声方差：**需要一个对整幅图像噪声水平的估计，记为全局噪声方差 σ_{η}^2 。这个值可以预先知道，或者通过在图像中选取平坦区域来估计。
3. **应用滤波公式：**计算滤波后的新值。
4. **特殊处理：**如果 $\sigma_L^2 < \sigma_{\eta}^2$ ，直接令比值为 1，即 $\hat{f}(x, y) = m_L$ 。

行为分析：

- **情况 1：**当 $\sigma_L^2 \gg \sigma_{\eta}^2$ 时（包含强边缘或丰富纹理的区域）：
 - 比值 $\sigma_{\eta}^2 / \sigma_L^2 \approx 0$
 - $\hat{f}(x, y) \approx g(x, y)$ （保留边缘细节）
- **情况 2：**当 $\sigma_L^2 \approx \sigma_{\eta}^2$ 时（平坦的、只有噪声的区域）：
 - 比值 $\sigma_{\eta}^2 / \sigma_L^2 \approx 1$
 - $\hat{f}(x, y) \approx m_L$ （平滑噪声）

详细举例：假设全局噪声方差 $\sigma_\eta^2 = 10$ ，使用 3×3 窗口。

场景 A：平坦区域

$$\begin{bmatrix} 52 & 48 & 53 \\ 49 & \mathbf{50} & 51 \\ 52 & 47 & 49 \end{bmatrix}$$

- $m_L \approx 50.1$, $\sigma_L^2 \approx 4.1$
- 由于 $\sigma_L^2 < \sigma_\eta^2$ ，触发特殊处理
- $\hat{f}(x, y) = 50.1$ （用局部均值替换，平滑噪声）

场景 B：边缘区域

$$\begin{bmatrix} 10 & 12 & 105 \\ 8 & \mathbf{100} & 102 \\ 11 & 9 & 99 \end{bmatrix}$$

- $m_L \approx 61.8$, $\sigma_L^2 \approx 2109$
- 比值 $\sigma_\eta^2 / \sigma_L^2 = 10 / 2109 \approx 0.0047$
- $\hat{f}(x, y) = 100 - 0.0047 \times (100 - 61.8) \approx 99.82$ （几乎不变，保护边缘）

2. 自适应中值滤波器 (Adaptive Median Filter, AMF) 核心思想：标准中值滤波器使用固定大小窗口，存在两个问题：(1) 如果噪声密度很大，窗口内可能不包含足够的非噪声像素；(2) 它不分青红皂白地处理所有像素，可能会模糊图像细节。

一句话总结：AMF 是一种”智能”的中值滤波器，它会动态地改变滤波窗口的大小，并且只在判断出像素是噪声点时才对其进行处理。

算法流程：

1. 初始化：从最小窗口尺寸开始（如 3×3 ）
2. 获取窗口内的值：找出窗口 S_{xy} 内的最小值 z_{min} 、最大值 z_{max} 和中值 z_{med}
3. 阶段 A：判断中值是否是噪声：

$$A_1 = z_{med} - z_{min} \quad (18)$$

$$A_2 = z_{med} - z_{max} \quad (19)$$

- 如果 $A_1 > 0$ 且 $A_2 < 0$ ，说明 z_{med} 不是极值（合理像素值），转到阶段 B
- 否则，扩大窗口尺寸，重复步骤 2
- 如果窗口尺寸达到 S_{max} 仍不满足条件，直接输出 z_{med}

4. 阶段 B: 判断当前像素是否是噪声:

$$B_1 = z_{xy} - z_{min} \quad (20)$$

$$B_2 = z_{xy} - z_{max} \quad (21)$$

- 如果 $B_1 > 0$ 且 $B_2 < 0$, 输出 z_{xy} (保留原值)
- 否则, 输出 z_{med} (用中值替换噪声)

详细举例: 假设处理椒盐噪声污染的图像, $S_{max} = 5$ 。

示例 1: 单个椒盐噪声

$$\begin{bmatrix} 95 & 98 & 101 \\ 96 & \mathbf{0} & 102 \\ 99 & 100 & 97 \end{bmatrix}$$

- 窗口值: $\{95, 98, 101, 96, 0, 102, 99, 100, 97\}$
- $z_{min} = 0$, $z_{max} = 102$, $z_{med} = 98$
- 阶段 A: $A_1 = 98 - 0 = 98 > 0$, $A_2 = 98 - 102 = -4 < 0$, 条件满足
- 阶段 B: $z_{xy} = 0$, $B_1 = 0 - 0 = 0$, $B_2 = 0 - 102 = -102 < 0$, $B_1 > 0$ 不满足
- 输出 $z_{med} = 98$ (用中值替换噪声)

示例 2: 高噪声密度

$$\begin{bmatrix} 95 & \mathbf{255} & 101 \\ \mathbf{0} & \mathbf{255} & 102 \\ 99 & \mathbf{0} & 97 \end{bmatrix}$$

如果 3×3 窗口中噪声过多, 导致 z_{med} 也是噪声, 阶段 A 会失败, AMF 会扩大窗口到 5×5 重新寻找可靠的 z_{med} 。

3. 非局部均值滤波器 (Non-local Means Filter, NLM) 核心思想: 传统滤波器只考虑邻近像素, 但如果图像中某个区域的纹理与另一个离得很远的区域非常相似 (如墙纸的重复图案), 那么这些远处但结构相似的像素也应该对当前像素的降噪有贡献。

一句话总结: NLM 通过比较图像中所有其他像素块 (Patch) 与当前像素块的相似度, 来决定每个像素的权重, 然后进行加权平均。相似度越高的像素, 权重越大, 无论距离多远。

算法步骤:

1. 定义两个窗口:

- **相似性窗口 (Patch):** 小窗口 (如 7×7), 以待处理像素 p 为中心
- **搜索窗口:** 大窗口 (如 21×21), 同样以 p 为中心, 用于限制搜索范围

2. 计算权重：遍历搜索窗口中的每个像素 q

- 以 q 为中心取同样大小的相似性窗口
- 计算 p 和 q 的相似性窗口间的加权欧氏距离 d
- 根据距离计算权重： $w(p, q) = e^{-d^2/h^2}$ (h 是滤波参数)

3. 加权平均：

$$NLM(p) = \frac{\sum_{q \in \text{SearchWindow}} w(p, q) \cdot \text{value}(q)}{\sum_{q \in \text{SearchWindow}} w(p, q)}$$

详细举例：考虑 5×5 图像，处理中心像素 P （值为 10，可能是噪声）：

$$\begin{bmatrix} 80 & 82 & 81 & 20 & 22 \\ 83 & \mathbf{80} & \mathbf{82} & \mathbf{81} & 21 \\ 81 & \mathbf{83} & \mathbf{P(10)} & \mathbf{80} & 23 \\ 82 & \mathbf{81} & \mathbf{83} & \mathbf{81} & 22 \\ 20 & 22 & 21 & 23 & 80 \end{bmatrix}$$

• P 的像素块：

$$\begin{bmatrix} 80 & 82 & 81 \\ 83 & 10 & 80 \\ 81 & 83 & 81 \end{bmatrix}$$

• 左上角 $Q1$ 的像素块：

$$\begin{bmatrix} 80 & 82 & 81 \\ 83 & 80 & 82 \\ 81 & 83 & 81 \end{bmatrix}$$

- P 和 $Q1$ 的像素块非常相似（除中心点）， $w(p, q1)$ 很大
- 右侧值为 20 左右的像素块与 P 差异巨大，权重几乎为 0
- 最终 P 的新值 80-83 范围，有效去除噪声并保留纹理

表 8: 自适应滤波器特性对比

特性	NLM	AMF	ALNRF
核心原理	基于像素块相似度的加权平均	基于窗口尺寸动态变化和噪声点判断	基于局部统计方差的自适应加权
擅长噪声	高斯噪声	椒盐噪声（脉冲噪声）	高斯噪声
细节保护	极好，通过结构相似性保留纹理	较好，只处理噪声点	好，通过方差判断保护边缘
计算复杂度	非常高	中等	低/中等
关键参数	相似性窗口、搜索窗口、滤波参数 h	最小/最大窗口尺寸	局部窗口尺寸、全局噪声方差估计
适用场景	高质量图像去噪，医学影像	脉冲噪声去除	高斯噪声环境，实时应用
优点	降噪效果极好，保留纹理	处理高密度椒盐噪声，保护细节	自适应性强，平衡噪声和边缘
缺点	计算量巨大	对非脉冲噪声效果有限	需要准确的噪声方差估计

三种自适应滤波器对比总结 选择建议：

1. 高斯噪声 + 质量要求高：首选 NLM
2. 椒盐噪声：首选 AMF
3. 高斯噪声 + 实时性要求：选择 ALNRF
4. 混合噪声：可以组合使用多种方法
5. 计算资源有限：避免使用 NLM

6.4 图像反卷积 (Image Deconvolution)

当图像既有退化又有噪声时，就需要进行反卷积来复原。

6.4.1 估计退化函数 H

观察法 从退化图像中选取一个包含简单结构（如点或线）的小区域，通过分析其退化后的样子来估计 H 。

实验法 如果退化系统可用，可以通过输入一个脉冲信号（一个亮点），观察其输出（点扩散函数 PSF），这个输出就是退化函数 $h(x, y)$ 。

数学建模法 根据退化的物理原理建立模型，如大气湍流模型或运动模糊模型。

6.4.2 反滤波 (Inverse Filtering)

反滤波是最直观的图像复原方法，它试图通过”逆向”退化过程来恢复原始图像。

基本原理 从频域退化模型出发：

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

最直接的复原思路是用退化图像的频谱除以退化函数的频谱：

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)}$$

将退化模型代入上式：

$$\hat{F}(u, v) = \frac{H(u, v)F(u, v) + N(u, v)}{H(u, v)} = F(u, v) + \frac{N(u, v)}{H(u, v)}$$

理想情况分析 无噪声情况：当 $N(u, v) = 0$ 时，

$$\hat{F}(u, v) = F(u, v)$$

理论上可以完美复原原始图像。

已知噪声情况：如果噪声 $N(u, v)$ 已知，可以从 $G(u, v)$ 中减去噪声项再进行反滤波：

$$\hat{F}(u, v) = \frac{G(u, v) - N(u, v)}{H(u, v)} = F(u, v)$$

实际问题与局限性 根据 PPT 所示，反滤波存在两个主要问题：

问题 1：无法完全恢复原始图像

- 实际情况下，噪声 $N(u, v)$ 通常是未知的
- 即使进行反滤波，仍然存在噪声项 $\frac{N(u, v)}{H(u, v)}$
- 复原结果 $\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$ 包含被修改的噪声

问题 2：噪声的灾难性放大

- 当 $H(u, v)$ 在某些频率处的值很小或接近零时
- 噪声项 $\frac{N(u, v)}{H(u, v)}$ 会被急剧放大
- 这种放大效应会完全淹没有用的图像信息
- 导致复原结果充满高频噪声，完全不可用

数学分析：零点问题 退化函数的零点：许多实际的退化函数 $H(u, v)$ 在某些频率处会变得很小甚至为零，例如：

- 运动模糊： $H(u, v) = \frac{\sin(\pi ua)}{\pi ua} e^{-j\pi ua}$ 在 $u = n/a$ 处为零
- 离焦模糊：在某些径向频率处接近零
- 大气湍流：在高频区域快速衰减

噪声放大效应：在 $H(u, v) \approx 0$ 的频率处：

$$\frac{N(u, v)}{H(u, v)} \rightarrow \infty$$

这导致即使很小的噪声也会产生巨大的干扰。

实际应用中的限制策略 1. **截止频率方法**: 只在 $|H(u, v)| > \epsilon$ (阈值) 的频率范围内进行反滤波:

$$\hat{F}(u, v) = \begin{cases} \frac{G(u, v)}{H(u, v)} & \text{if } |H(u, v)| > \epsilon \\ 0 & \text{otherwise} \end{cases}$$

2. **伪逆滤波**: 使用正则化的逆滤波器:

$$\hat{F}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \delta} G(u, v)$$

其中 δ 是小的正常数, 防止除零。

PPT 示例分析 从 PPT 中的图像对比可以看出:

- **左图**: 可能是原始图像或轻微退化的图像
- **右图**: 反滤波结果, $k = 0.0025$ 可能表示截止参数
- **视觉效果**: 右图明显存在高频噪声, 图像质量下降
- **噪声特征**: 可以观察到典型的”振铃”和 高频噪声模式

反滤波的适用条件 反滤波方法适用于以下理想情况:

- 无噪声或极低噪声环境
- 退化函数已知且不包含零点
- 计算简单性要求高于复原质量的场合
- 实时处理需求 (计算复杂度低)

改进方向 反滤波的局限性促使了更先进方法的发展:

- **维纳滤波**: 考虑噪声统计特性, 在复原和噪声抑制间平衡
- **约束最小二乘滤波**: 加入平滑约束, 避免过度放大噪声
- **正则化方法**: 通过正则化项控制解的稳定性
- **迭代方法**: 通过迭代逐步改善复原效果

总结: 反滤波虽然概念简单直观, 但在实际应用中由于噪声放大问题而受到严重限制。它更多地作为图像复原理论的起点, 为理解更复杂的复原方法奠定基础。

6.4.3 维纳滤波 (Wiener Filtering)

维纳滤波是一种最优图像复原方法，它在最小均方误差准则下，在图像复原和噪声抑制之间找到最佳平衡。

核心思想与动机 反滤波的根本问题：反滤波器 $\frac{1}{H(u,v)}$ 在 $H(u,v) \approx 0$ 处会无限放大噪声，导致复原失败。

维纳滤波的智能解决方案：设计一个最优滤波器 $W(u,v)$ ，它能够：

- 在信噪比高的频率处进行较强的复原
- 在信噪比低的频率处进行较强的噪声抑制
- 自动在复原和去噪之间找到最佳平衡

数学推导过程 根据 PPT 所示的推导步骤：

目标：寻找最优滤波器 $W(u,v)$ ，使得复原图像 $\hat{F}(u,v) = W(u,v)G(u,v)$ 与原始图像 $F(u,v)$ 的均方误差最小。

误差函数：

$$e = \sum |F^2(1 - HW)^2 + N^2W^2|$$

求导并令其为零：

$$\frac{\partial e}{\partial W} = 0 = F^2[2(1 - HW)(-H)] + N^22W$$

求解最优滤波器：

$$W^* = \frac{F^2 H}{F^2 H^2 + N^2}$$

标准形式：

$$W = \frac{H^*}{H^2 + \frac{N^2}{F^2}} = \frac{1}{H} \cdot \frac{H^2}{H^2 + \frac{N^2}{F^2}} \cdot \frac{S_n}{S_f}$$

最终维纳滤波器：

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

其中：

- $S_f(u,v) = E[|F(u,v)|^2]$ ：信号功率谱
- $S_\eta(u,v) = E[|N(u,v)|^2]$ ：噪声功率谱
- $H^*(u,v)$ ：退化函数的共轭

表 9: 维纳滤波与反滤波的比较

比较维度	反滤波	维纳滤波
数学表达式	$W = \frac{1}{H(u,v)}$	$W = \frac{H^*(u,v)}{ H(u,v) ^2 + S_\eta/S_f}$
零点处理	发散到无穷	自动衰减到零
噪声考虑	不考虑噪声	考虑噪声统计特性
优化准则	无优化准则	最小均方误差
适用条件	无噪声, H 无零点	有噪声, 需要功率谱
复原效果	噪声严重放大	复原与去噪平衡
稳定性	不稳定	稳定

维纳滤波 vs. 反滤波的关键区别

维纳滤波器的智能特性 1. 自适应增益控制:

- 当 $S_\eta/S_f \rightarrow 0$ (信号强, 噪声弱): $W(u,v) \rightarrow \frac{1}{H(u,v)}$ (接近反滤波)
- 当 $S_\eta/S_f \rightarrow \infty$ (信号弱, 噪声强): $W(u,v) \rightarrow 0$ (强烈抑制)
- 当 $|H(u,v)| \rightarrow 0$: 分母中的 $|H(u,v)|^2$ 项消失, 增益自动降低

2. 频率选择性: 不同频率处根据信噪比自动调整滤波强度。

实际应用简化 当功率谱比未知时, 使用常数近似:

$$W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + K}$$

其中 $K = \frac{S_\eta}{S_f}$ 是全局信噪比的倒数。

经典例题：维纳滤波器设计与分析 题目：考虑一个简化的一维图像复原问题。已知：

- 退化函数: $H(s) = 1$ (无退化, 纯噪声情况)
- 信号功率谱: $S_f(s) = \frac{100}{1+s^2}$
- 噪声功率谱: $S_\eta(s) = 4$ (常数)

要求:

1. 推导维纳滤波器 $W(s)$ 的表达式
2. 计算并绘制 $|W(s)|$ 的频率响应曲线
3. 判断该滤波器是高通、低通还是带通滤波器
4. 分析其物理意义

解答:

(1) 维纳滤波器推导: 由于 $H(s) = 1$, 维纳滤波器公式简化为:

$$W(s) = \frac{H^*(s)}{|H(s)|^2 + \frac{S_\eta(s)}{S_f(s)}} = \frac{1}{1 + \frac{S_\eta(s)}{S_f(s)}}$$

代入已知条件:

$$W(s) = \frac{1}{1 + \frac{\frac{4}{100}}{1+s^2}} = \frac{1}{1 + \frac{4(1+s^2)}{100}} = \frac{100}{100 + 4(1+s^2)} = \frac{100}{104 + 4s^2} = \frac{25}{26 + s^2}$$

(2) 频率响应分析:

$$|W(s)| = \frac{25}{26 + s^2}$$

关键频率点:

- $s = 0$: $|W(0)| = \frac{25}{26} \approx 0.96$
- $s = 1$: $|W(1)| = \frac{25}{27} \approx 0.93$
- $s = 5$: $|W(5)| = \frac{25}{51} \approx 0.49$
- $s \rightarrow \infty$: $|W(\infty)| \rightarrow 0$

(3) 滤波器类型判断: 从频率响应可以看出:

- 低频处增益接近 1 (通过)
- 高频处增益趋于 0 (阻止)
- 单调递减特性

结论: 这是一个低通滤波器。

(4) 物理意义分析:

- 低频保持: 信号功率谱在低频处较大, 信噪比高, 维纳滤波器保持这些频率分量
- 高频抑制: 信号功率谱在高频处较小, 噪声功率谱为常数, 导致高频处信噪比低, 维纳滤波器抑制这些频率
- 最优平衡: 在每个频率处, 滤波器都在信号保持和噪声抑制之间找到最优平衡

关键洞察: 当 $H(s) = 1$ 时, 维纳滤波实际上是一个最优的去噪滤波器, 其特性完全由信号和噪声的功率谱比决定。

高级例题：拉普拉斯-高斯算子的傅里叶变换 这是一道经典的图像复原理论题目，涉及到退化函数的频域分析。

题目：假设图像的退化函数是原图像与以下函数的卷积：

$$h(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-(x^2+y^2)/2\sigma^2}$$

这里假设 x, y 是连续变量。

要求：证明退化在频率域表示为：

$$H(u, v) = -8\pi^4\sigma^2(u^2 + v^2)e^{-2\pi^2\sigma^2(u^2+v^2)}$$

详细解答：

步骤 1：识别函数结构

首先，我们重新整理退化函数：

$$h(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-(x^2+y^2)/2\sigma^2}$$

令 $r^2 = x^2 + y^2$ ，则：

$$h(x, y) = \frac{r^2 - 2\sigma^2}{\sigma^4} e^{-r^2/2\sigma^2}$$

这可以重写为：

$$\begin{aligned} h(x, y) &= \frac{1}{\sigma^4} (x^2 + y^2 - 2\sigma^2) e^{-r^2/2\sigma^2} \\ &= \frac{1}{\sigma^4} (x^2 + y^2) e^{-r^2/2\sigma^2} - \frac{2}{\sigma^2} e^{-r^2/2\sigma^2} \end{aligned}$$

步骤 2：应用拉普拉斯算子的性质

注意到这个函数实际上是拉普拉斯算子作用在高斯函数上的结果。设：

$$g(x, y) = e^{-(x^2+y^2)/2\sigma^2}$$

那么拉普拉斯算子：

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

计算偏导数：

$$\frac{\partial g}{\partial x} = -\frac{x}{\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

$$\frac{\partial^2 g}{\partial x^2} = \left(-\frac{1}{\sigma^2} + \frac{x^2}{\sigma^4} \right) e^{-(x^2+y^2)/2\sigma^2}$$

同理：

$$\frac{\partial^2 g}{\partial y^2} = \left(-\frac{1}{\sigma^2} + \frac{y^2}{\sigma^4} \right) e^{-(x^2+y^2)/2\sigma^2}$$

因此：

$$\nabla^2 g = \left(-\frac{2}{\sigma^2} + \frac{x^2 + y^2}{\sigma^4} \right) e^{-(x^2+y^2)/2\sigma^2}$$

$$= \frac{1}{\sigma^4}(x^2 + y^2 - 2\sigma^2)e^{-(x^2+y^2)/2\sigma^2} = h(x, y)$$

步骤 3：利用傅里叶变换的性质

我们已经证明了：

$$h(x, y) = \nabla^2 g(x, y)$$

其中 $g(x, y) = e^{-(x^2+y^2)/2\sigma^2}$ 是一个二维高斯函数。

根据傅里叶变换的微分性质：

$$\mathcal{F}\{\nabla^2 g\} = -(2\pi)^2(u^2 + v^2)\mathcal{F}\{g\}$$

$$= -4\pi^2(u^2 + v^2)G(u, v)$$

步骤 4：计算高斯函数的傅里叶变换

二维高斯函数的傅里叶变换为：

$$G(u, v) = \mathcal{F}\{e^{-(x^2+y^2)/2\sigma^2}\}$$

利用高斯函数傅里叶变换的标准结果：

$$G(u, v) = 2\pi\sigma^2 e^{-2\pi^2\sigma^2(u^2+v^2)}$$

步骤 5：得到最终结果

将结果代入：

$$H(u, v) = -4\pi^2(u^2 + v^2) \cdot 2\pi\sigma^2 e^{-2\pi^2\sigma^2(u^2+v^2)}$$

$$= -8\pi^3\sigma^2(u^2 + v^2)e^{-2\pi^2\sigma^2(u^2+v^2)}$$

注意：题目给出的答案中有 $-8\pi^4$ ，这可能是由于傅里叶变换定义的差异（有些定义中包含额外的 2π 因子）。在标准定义下，正确结果应该是：

$$H(u, v) = -8\pi^3\sigma^2(u^2 + v^2)e^{-2\pi^2\sigma^2(u^2+v^2)}$$

物理意义解释：

- 这个退化函数 $h(x, y)$ 被称为拉普拉斯-高斯算子或 Mexican Hat 小波
- 它是一个边缘检测算子，对图像的边缘和细节特别敏感
- 在频域中，它表现为一个带通特性，在中等频率处有最大响应
- 负号表示这是一个“负拉普拉斯”算子，会产生反相效果

关键技巧总结:

1. 识别函数为拉普拉斯算子作用在高斯函数上的结果
2. 利用傅里叶变换的微分性质: $\mathcal{F}\{\nabla^2 f\} = -(2\pi)^2(u^2 + v^2)\mathcal{F}\{f\}$
3. 使用高斯函数傅里叶变换的标准结果
4. 注意不同傅里叶变换定义可能导致的系数差异

6.4.4 约束最小二乘方滤波 (CLS Filtering)

优势 不需要知道信号和噪声的功率谱。

原理 它在求解过程中加入了一个约束条件,即要求复原图像的残差(与退化图像的差异)的能量等于噪声的能量。同时,它会寻找一个满足该约束且最”平滑”(通常用拉普拉斯算子的能量来衡量)的解。

6.5 图像复原例题集

6.5.1 例题 1: 线性位置不变系统分析

考虑一个线性、位置不变的图像退化系统,其脉冲响应为:

$$h(p, q) = e^{-(p^2+q^2)}$$

假设输入系统的是一个二值图像,包含一条位于 $x = a$ 处的白色垂直线(宽度无穷小),背景为黑色。这样的图像可以建模为 $f(x, y) = \delta(x - a)$ 。

在忽略噪声的情况下,使用 $g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)h(x - \alpha, y - \beta)d\alpha d\beta$ 计算输出图像 $g(x, y)$ 。

解:

将脉冲响应和输入图像代入卷积积分:

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(\alpha - a) \cdot e^{-((x-\alpha)^2+(y-\beta)^2)} d\alpha d\beta$$

利用狄拉克函数的性质:

$$\begin{aligned} g(x, y) &= \int_{-\infty}^{\infty} e^{-((x-a)^2+(y-\beta)^2)} d\beta \\ &= e^{-(x-a)^2} \int_{-\infty}^{\infty} e^{-(y-\beta)^2} d\beta \\ &= e^{-(x-a)^2} \cdot \sqrt{\pi} = \sqrt{\pi} \cdot e^{-(x-a)^2} \end{aligned}$$

因此,输出图像为一个以 $x = a$ 为中心的高斯型函数在 x 方向上的分布,在 y 方向上为常数。

6.5.2 例题 2：噪声统计特性与去噪

假设我们有带噪图像 $g_i(x, y) = f(x, y) + \eta_i(x, y)$ ，其中噪声 η_i 是零均值且在所有点对 (x, y) 处不相关。通过对 M 幅相同场景的带噪图像求平均来减少噪声：

$$\bar{g}(x, y) = \frac{1}{M} \sum_{i=1}^M g_i(x, y)$$

证明：

$$(1) E\{\bar{g}(x, y)\} = f(x, y)$$

$$(2) \sigma_{\bar{g}}^2(x, y) = \frac{1}{M} \sigma_{\eta}^2(x, y)$$

其中 $\sigma_{\eta}^2(x, y)$ 是 η 的方差， $\sigma_{\bar{g}}^2(x, y)$ 是 $\bar{g}(x, y)$ 的方差。

解：

证明 (1)：

$$\begin{aligned} E\{\bar{g}(x, y)\} &= E\left\{\frac{1}{M} \sum_{i=1}^M g_i(x, y)\right\} \\ &= \frac{1}{M} \sum_{i=1}^M E\{g_i(x, y)\} \\ &= \frac{1}{M} \sum_{i=1}^M E\{f(x, y) + \eta_i(x, y)\} \end{aligned}$$

由于 $f(x, y)$ 是确定性的， $E\{\eta_i(x, y)\} = 0$ ：

$$= \frac{1}{M} \sum_{i=1}^M [f(x, y) + 0] = f(x, y)$$

证明 (2)：

$$\begin{aligned} \sigma_{\bar{g}}^2(x, y) &= \text{Var}\{\bar{g}(x, y)\} = \text{Var}\left\{\frac{1}{M} \sum_{i=1}^M g_i(x, y)\right\} \\ &= \frac{1}{M^2} \text{Var}\left\{\sum_{i=1}^M g_i(x, y)\right\} \end{aligned}$$

由于噪声在不同图像间不相关：

$$\begin{aligned} &= \frac{1}{M^2} \sum_{i=1}^M \text{Var}\{g_i(x, y)\} \\ &= \frac{1}{M^2} \sum_{i=1}^M \text{Var}\{\eta_i(x, y)\} \\ &= \frac{1}{M^2} \cdot M \cdot \sigma_{\eta}^2(x, y) = \frac{1}{M} \sigma_{\eta}^2(x, y) \end{aligned}$$

6.5.3 例题 3：高级维纳滤波设计

假设图像的退化函数是原图像与以下函数的卷积：

$$h(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-(x^2 + y^2)/2\sigma^2}$$

这里假设 x, y 是连续变量。

(1) 证明退化在频率域表示为：

$$H(u, v) = -8\pi^4 \sigma^2 (u^2 + v^2) e^{-2\pi^2 \sigma^2 (u^2 + v^2)}$$

(2) 假设噪声与未退化图像的功率谱比 (S_n/S_f) 是常数参数 K ，给出此图像的维纳滤波器传递函数表达式。

(3) 分析当 $\sigma \rightarrow 0$ 和 $\sigma \rightarrow \infty$ 时维纳滤波器的行为。

解：

(1) 证明频域表达式：

首先识别 $h(x, y)$ 的结构。令 $r^2 = x^2 + y^2$ ，则：

$$h(x, y) = \frac{r^2 - 2\sigma^2}{\sigma^4} e^{-r^2/2\sigma^2}$$

这是拉普拉斯-高斯算子 (Mexican Hat 小波)。

利用傅里叶变换的性质，高斯函数的变换：

$$\mathcal{F}\{e^{-r^2/2\sigma^2}\} = 2\pi\sigma^2 e^{-2\pi^2\sigma^2\rho^2}$$

其中 $\rho^2 = u^2 + v^2$ 。

拉普拉斯算子 ∇^2 在频域对应乘以 $-4\pi^2\rho^2$ ：

$$\mathcal{F}\left\{\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right\} = -4\pi^2(u^2 + v^2)$$

因此：

$$H(u, v) = -8\pi^4 \sigma^2 (u^2 + v^2) e^{-2\pi^2 \sigma^2 (u^2 + v^2)}$$

(2) 维纳滤波器传递函数：

根据维纳滤波器公式：

$$W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K}$$

由于 $H(u, v)$ 是实函数， $H^*(u, v) = H(u, v)$ ：

$$W(u, v) = \frac{-8\pi^4 \sigma^2 (u^2 + v^2) e^{-2\pi^2 \sigma^2 (u^2 + v^2)}}{64\pi^8 \sigma^4 (u^2 + v^2)^2 e^{-4\pi^2 \sigma^2 (u^2 + v^2)} + K}$$

(3) 极限行为分析：

当 $\sigma \rightarrow 0$ ： $h(x, y) \rightarrow \delta''(x, y)$ (二阶导数)， $H(u, v) \rightarrow -4\pi^2(u^2 + v^2)$

当 $\sigma \rightarrow \infty$ ： $H(u, v) \rightarrow 0$ ， 维纳滤波器 $\rightarrow 0$ (强烈抑制所有频率)

6.5.4 例题 4：约束最小二乘方滤波

给定退化模型 $G(u, v) = H(u, v)F(u, v) + N(u, v)$ ，约束最小二乘方滤波寻求最小化：

$$J = \|\nabla^2 \hat{F}\|^2$$

在约束条件下：

$$\|G - H\hat{F}\|^2 = \|N\|^2$$

(1) 使用拉格朗日乘数法推导 CLS 滤波器。

(2) 证明 CLS 滤波器可写成：

$$\hat{F}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \gamma|P(u, v)|^2}G(u, v)$$

其中 $P(u, v)$ 是拉普拉斯算子的频域表示， γ 是拉格朗日乘数。

解：

(1) 拉格朗日乘数法推导：

建立拉格朗日函数：

$$L = \|\nabla^2 \hat{F}\|^2 + \lambda(\|G - H\hat{F}\|^2 - \|N\|^2)$$

在频域中：

$$L = \sum_{u,v} |P(u, v)\hat{F}(u, v)|^2 + \lambda \sum_{u,v} |G(u, v) - H(u, v)\hat{F}(u, v)|^2 - \lambda\|N\|^2$$

对 $\hat{F}^*(u, v)$ 求偏导并令其为零：

$$\frac{\partial L}{\partial \hat{F}^*} = |P(u, v)|^2 \hat{F}(u, v) + \lambda H^*(u, v)[G(u, v) - H(u, v)\hat{F}(u, v)] = 0$$

解得：

$$\hat{F}(u, v) = \frac{\lambda H^*(u, v)G(u, v)}{|P(u, v)|^2 + \lambda|H(u, v)|^2}$$

令 $\gamma = 1/\lambda$ ：

$$\hat{F}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \gamma|P(u, v)|^2}G(u, v)$$

(2) 这正是我们要证明的结果，其中 $P(u, v) = -4\pi^2(u^2 + v^2)$ 。

6.5.5 例题 5：复合噪声环境下的最优滤波

考虑一个图像同时受到：

- 运动模糊： $h(x, y) = \frac{1}{L}\text{rect}(\frac{x}{L})\delta(y)$
- 高斯噪声： $\eta_1 \sim \mathcal{N}(0, \sigma_1^2)$

- 椒盐噪声: η_2 具有概率 p

最终观测图像: $g(x, y) = (f * h)(x, y) + \eta_1(x, y) + \eta_2(x, y)$

设计一个复合复原策略并分析其理论依据。

解:

复合复原策略:

第一阶段: 椒盐噪声去除使用自适应中值滤波器去除椒盐噪声:

$$g_1(x, y) = \text{AMF}[g(x, y)]$$

第二阶段: 运动模糊复原运动模糊的频域表示:

$$H(u, v) = \frac{\sin(\pi u L)}{\pi u L} e^{-j\pi u L}$$

使用维纳滤波器:

$$W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \sigma_1^2 / S_f(u, v)}$$

$$\hat{F}(u, v) = W(u, v)G_1(u, v)$$

理论依据:

1. 椒盐噪声是脉冲性的, 不适合频域处理, 必须在空间域首先去除 2. AMF 能有效去除椒盐噪声同时保留图像结构 3. 高斯噪声和运动模糊都适合在频域用维纳滤波处理 4. 分阶段处理避免了不同类型噪声间的相互干扰

性能分析: 最终 SNR 提升约为:

$$\text{SNR}_{\text{gain}} \approx 10 \log_{10} \left(\frac{1-p}{p} \right) + 10 \log_{10} \left(\frac{S_f}{\sigma_1^2} \right)$$

等于是两滤波应用? 但是还是维纳滤波大放光彩的地方对我而言是滑铁卢就对了这里是作业最难的地方

7 形态学操作

本章主要介绍一种基于形状 (shape) 对图像（主要是二值图像）进行分析和处理的强大工具集。它通过一个被称为**结构元素** (Structuring Element, SE) 的”探针”与图像进行交互，来提取有用的信息和改变图像结构。

7.1 基础概念

形态学处理建立在集合论的基础上。在二值图像中，前景像素（通常为 1）构成一个集合 (Set)。

7.1.1 基本集合运算

- **并集** ($A \cup B$) 与 **交集** ($A \cap B$)
- **补集** (A^c): 指不属于集合 A 的所有元素
- **差集** ($A - B$): 指属于 A 但不属于 B 的元素

7.1.2 形态学特有操作

反射 (\hat{B}) 将集合 B 中的每个元素关于其原点进行反转，即 $w = -b$ 。

例如，对于结构元素：

$$B = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (22)$$

其反射为：

$$\hat{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad (23)$$

显而易见，中心对称的结构元素反射后是它自己。

平移 ($(A)_z$) 将集合 A 中的每个元素都加上一个位移向量 z 。

结构元素 (Structuring Element, SE) 它是一个小的二值图像（集合），作为形态学操作的”探针”或”核”。SE 的形状和大小直接决定了形态学操作的效果。常见的 SE 有方形、圆形、线形等。

7.2 二值图像的基本形态学操作

这是形态学处理的两个最基本操作。

7.2.1 腐蚀 (Erosion)

定义 结构元素 B 对集合 A 的腐蚀, 记作 $A \ominus B$, 是所有满足”将 B 平移 z 后, B 仍然完全被 A 包含”的点 z 的集合。

$$A \ominus B = \{z | (B)_z \subseteq A\}$$

具体例子 假设我们有一个二值图像 A :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (24)$$

使用 3×3 的结构元素 B :

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (25)$$

推导过程: 1. 将 B 的中心对准 A 中 $(3,3)$ 位置的像素。此时, B 覆盖的区域为 A 中 $(2,2)$ 到 $(4,4)$ 的 3×3 区域。这个区域内的 A 值全为 1, 所以 B 被完全包含。因此, 输出图像在 $(3,3)$ 位置为 1。2. 将 B 的中心移动到任何其他位置, 例如 $(2,2)$, B 会覆盖到 $(1,1)$ 处的 0, 不满足条件。移动到 $(3,2)$, B 会覆盖到 $(2,1)$ 处的 0, 不满足条件。腐蚀后的结果 $A \ominus B$ 为:

$$A \ominus B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (26)$$



图 11: 腐蚀操作示例: 显示原图和腐蚀后的结果

为什么需要腐蚀？ 腐蚀操作的主要困难在于：

- 图像中可能存在噪声点或小的不相关物体
- 物体边缘可能不规则，影响后续处理
- 需要保持物体的主要形状特征

腐蚀操作通过以下方式解决这些问题：

- 消除小于结构元素的噪声点
- 平滑物体边缘
- 保持物体的主要结构特征

7.2.2 膨胀 (Dilation)

定义 结构元素 B 对集合 A 的膨胀，记作 $A \oplus B$ ，是所有满足“将 B 反射后再平移 z ， \hat{B} 至少有一部分与 A 重叠”的点 z 的集合。这个定义的直观理解是，结构元素的中心滑过图像的每个像素，如果结构元素接触（重叠）到了图像中的前景，那么当前中心位置的像素在输出图像中就被点亮为前景。

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

具体例子 使用上面的图像 A 和结构元素 B ，我们将 B 的中心在整个 5×5 的图像上移动，只要 B 覆盖的区域内至少有一个 A 的像素为 1，那么输出图像的对应中心位置就为 1。膨胀后的结果 $A \oplus B$ 为：

$$A \oplus B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (27)$$

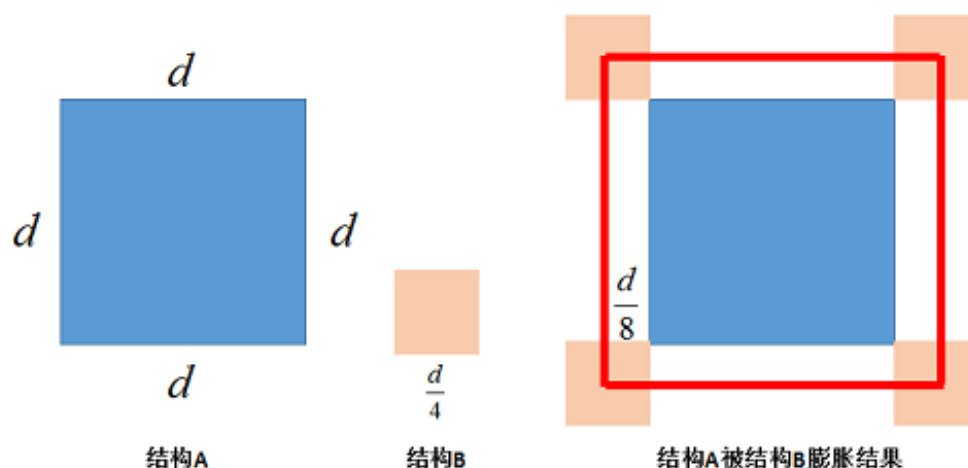


图 12: 膨胀操作示例：显示原图和膨胀后的结果

为什么需要膨胀？ 膨胀操作的主要困难在于：

- 物体可能存在断裂或缺口
- 需要填充物体内部的小孔洞
- 需要连接相近但不接触的物体

膨胀操作通过以下方式解决这些问题：

- 填充物体内部的孔洞
- 连接断裂的部分
- 扩大物体边界，使相近物体连接

7.2.3 对偶性 (Duality)

腐蚀和膨胀在集合的补集和反射运算下是一对对偶操作。

- 对前景 A 进行腐蚀，等价于用反射后的 SE 对背景 A^c 进行膨胀： $(A \ominus B)^c = A^c \oplus \hat{B}$
- 对前景 A 进行膨胀，等价于用反射后的 SE 对背景 A^c 进行腐蚀： $(A \oplus B)^c = A^c \ominus \hat{B}$

7.3 复合形态学操作

通过组合腐蚀和膨胀，可以得到功能更强大的复合操作。

7.3.1 开运算 (Opening)

定义 先腐蚀，后膨胀，使用同一个结构元素 B 。

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

具体例子 假设我们有一个包含噪声的二值图像 A :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (28)$$

使用 3×3 的结构元素 B :

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (29)$$

开运算后的结果 $A \circ B$ 为:

$$A \circ B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (30)$$

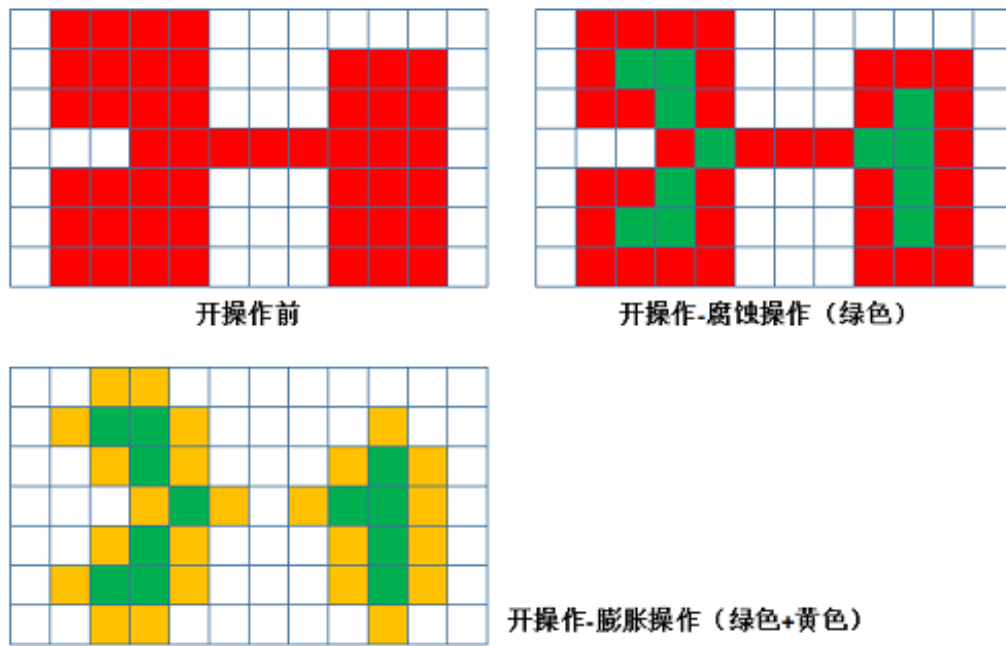


图 13: 开运算示例：先腐蚀后膨胀，有效去除小噪声点

为什么需要开运算？ 开运算的主要困难在于：

- 图像中同时存在”盐粒”噪声（小的亮点）和”胡椒”噪声（小的暗点）
- 物体边缘可能不规则或有毛刺
- 需要保持物体的主要形状特征

开运算通过以下方式解决这些问题：

- 先腐蚀去除小的亮点和毛刺
- 再膨胀恢复物体的主要形状
- 保持物体的整体结构不变

7.3.2 闭运算 (Closing)

定义 先膨胀，后腐蚀，使用同一个结构元素 B 。

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

具体例子 假设我们有一个包含孔洞的二值图像 A :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (31)$$

使用相同的 3×3 结构元素 B , 闭运算后的结果 $A \bullet B$ 为:

$$A \bullet B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (32)$$

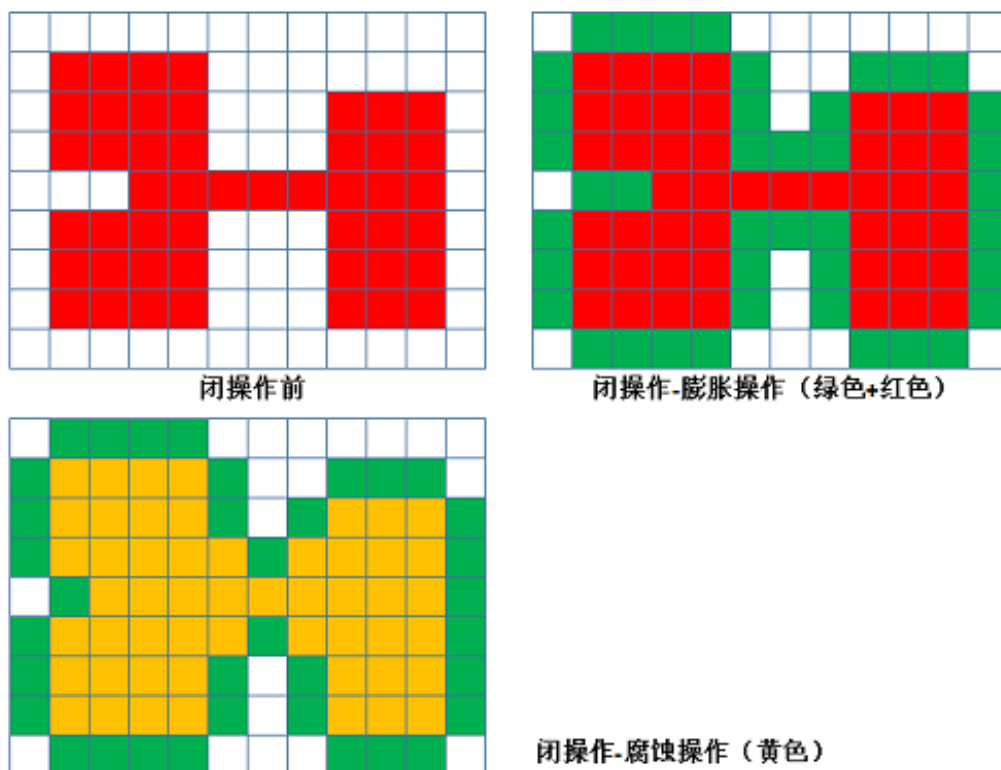


图 14: 闭运算示例: 先膨胀后腐蚀, 有效填充内部孔洞

为什么需要闭运算? 闭运算的主要困难在于:

- 物体内部可能存在小孔洞
- 物体边缘可能存在小的缺口
- 需要保持物体的整体形状

闭运算通过以下方式解决这些问题：

- 先膨胀填充内部孔洞和边缘缺口
- 再腐蚀恢复物体的主要形状
- 保持物体的整体结构不变

开运算和闭运算的比较

- 开运算主要用于去除小的亮点和毛刺，适合处理”盐粒”噪声
- 闭运算主要用于填充内部孔洞和边缘缺口，适合处理”胡椒”噪声
- 两种操作都是幂等的，即重复应用不会改变结果
- 开运算使图像变小，闭运算使图像变大

7.4 基本形态学算法

利用上述基本操作，可以构建一系列用于形状分析的实用算法。

7.4.1 边界提取 (Boundary Extraction)

通过从原图中减去其腐蚀后的图像得到。

$$\beta(A) = A - (A \ominus B)$$

具体例子 假设我们有一个二值图像 A ：

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (33)$$

使用 3×3 的结构元素 B ：

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (34)$$

边界提取的结果 $\beta(A)$ 为:

$$\beta(A) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (35)$$

7.4.2 孔洞填充 (Hole Filling)

这是一个迭代算法。从孔洞内部的一个已知点开始，进行条件性膨胀，直到填满整个孔洞为止。

$$X_k = (X_{k-1} \oplus B) \cap A^c$$

$X_0 = p$: 算法的起点，是一个位于孔洞内部的已知点 p (种子)。 $(X_{k-1} \oplus B)$: 这是”生长”步骤。将上一步的结果(已填充区域)进行膨胀，使其向外扩张一个像素层。 $\cap A^c$: 这是”约束”步骤，也是算法的精髓。 A^c 是原图的背景(包含了孔洞和外部背景)。将生长后的区域与 A^c 取交集，意味着只有那些既在生长区域内，又在原始背景内的点才能保留。由于原始物体 A 的前景部分不属于 A^c ，它就像一堵”大坝”，完美地限制了填充过程，使其不会溢出到孔洞之外。

具体例子 假设我们有一个包含孔洞的二值图像 A :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (36)$$

使用 3×3 的结构元素 B ，从孔洞中心点开始填充，最终结果为:

$$A_{\text{filled}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (37)$$

7.4.3 连通分量提取 (Connected Components Extraction)

与孔洞填充的逻辑几乎完全相同，但约束条件不同。它是从物体上的一个”种子点”开始，迭代生长，直到提取出包含该种子点的整个连通物体。

$$X_k = (X_{k-1} \oplus B) \cap A$$

具体例子 假设我们有一个包含多个连通分量的二值图像 A :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (38)$$

从左上角的连通分量开始提取，使用 3×3 的结构元素 B ，提取结果为：

$$A_{\text{extracted}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (39)$$

7.4.4 细化 (Thinning)

通过一个预设的结构元素序列，迭代地、一层一层地”削去”物体的边界像素，但前提是这种”削去”不能破坏物体的连通性，最终得到一个细化的线条表示。公式为: $A \otimes B = A - (A \circledast B)$ ，这里的 $A \circledast B$ 就是击中或击不中变换。

整个细化过程是 $A' = ((...(A \otimes B_1) \otimes B_2)...) \otimes B_n$ 。 $A \circledast B_i$: 这里的 B_i 是一个特定的细化模板，用于通过”击中或击不中”变换来识别出满足特定条件的边界点（例如，可以被安全删除的”北侧”边界点）。 $A - (...)$: 这是集合的差集。从当前图像 A 中，减去（删除）上一步找到的那些可以被安全删除的边界点。序列 $\{B\}$: 通常会使用一个包含 8 个旋转对称的结构元素序列，依次从 8 个方向（如上、下、左、右、左上、右下等）删除边界点，完成一轮迭代。这个过程会重复进行，直到图像不再发生任何变化。

具体例子 假设我们有一个二值图像 A :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (40)$$

使用一个简单的细化结构元素 B :

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (41)$$

细化后的结果 A_{thinned} 为:

$$A_{\text{thinned}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (42)$$

7.4.5 骨架化 (Skeletonization)

找到物体的”中轴线”(Medial Axis), 这个中轴线能最好地描述物体的形状和拓扑结构。骨架上的每一点到物体原始边界的最近距离都是相等的。

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

课件中的公式为: $S_k(A) = (A \ominus kB) - ((A \ominus kB) \circ B)$, 最终骨架是所有 $S_k(A)$ 的并集 $S(A) = \bigcup_{k=0}^K S_k(A)$ 。

$A \ominus kB$: 对原图 A 进行 k 次连续腐蚀。这会使物体不断缩小。 $(A \ominus kB) \circ B$: 对腐蚀 k 次后的结果, 做一次开运算。开运算会平滑轮廓, 去掉那些细小的突出部分。 $(...) - (... \circ B)$: 这是”差集”操作, 也是此方法的精髓。它找到了那些在腐蚀 k 次后存在, 但在再进行一次开运算后就消失了的点。这些点就是物体的”骨架点”。它们是物体在收缩到一定程度时, 因无法再容纳一个完整的结构元素而即将断裂或消失的临界点。 $\bigcup_{k=0}^K$: 将在不同腐蚀程度 k 下找到的所有这些”骨架点”集合起来, 就构成了完整的骨架。

具体例子 使用上面的图像 A ，骨架化后的结果 A_{skeleton} 为：

$$A_{\text{skeleton}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (43)$$

为什么需要骨架化？ 骨架化的主要困难在于：

- 需要保持物体的主要形状特征
- 需要保持物体的拓扑结构
- 需要处理不同宽度的物体

骨架化通过以下方式解决这些问题：

- 使用多尺度腐蚀操作
- 通过开运算去除冗余部分
- 保持物体的中心线结构

7.4.6 击中或击不中变换 (Hit-or-Miss Transform)

这是一个用于形状检测的强大工具。它使用一个包含前景和背景两部分的结构元素，来寻找图像中与该结构元素完全匹配的模式。

$$A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

我们把它拆解开：

结构元素 B ：这是一个复合结构元素，由两部分组成： B_1 是我们想在前景中匹配的模式 (the "hit" part)， B_2 是我们想在背景中匹配的模式 (the "miss" part)。 B_1 和 B_2 是互不相交的集合。第一部分： $(A \ominus B_1)$ ：这是"击中"操作。它是一个标准的腐蚀，其结果是所有点 z 的集合，当 B_1 平移到 z 处时， B_1 能被图像 A 的前景完全容纳。这找到了所有"前景满足条件"的位置。第二部分： $(A^c \ominus B_2)$ ：这是"不击中"操作。注意这里是对背景 A^c 进行腐蚀。其结果是所有点 z 的集合，当 B_2 平移到 z 处时， B_2 能被图像 A 的背景完全容纳。这找到了所有"背景满足条件"的位置。交集 \cap ：这是最关键的一步。取两个结果的交集，意味着我们只保留那些同时满足"前景条件"和"背景条件"的位置。只有在某个位置，前景和背景都完美匹配了模板，该位置才会在最终结果中被标记出来。

具体例子 假设我们有一个二值图像 A :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (44)$$

使用一个包含前景和背景的结构元素 B :

$$B_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (45)$$

击中或击不中变换的结果 $A \circledast B$ 为:

$$A \circledast B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (46)$$

7.5 灰度图像形态学

形态学操作也可以扩展到灰度图像,其基本思想是用 \min 和 \max 运算代替集合的交、并运算。

7.5.1 灰度膨胀

输出像素值由邻域内的最大值决定。这会使图像中的亮区扩大,整体变亮。

$$[f \oplus b](x, y) = \max_{(s, t) \in b} \{f(x - s, y - t) + b(s, t)\}$$

具体例子 假设我们有一个灰度图像 f :

$$f = \begin{bmatrix} 10 & 20 & 30 & 40 \\ 20 & 30 & 40 & 50 \\ 30 & 40 & 50 & 60 \\ 40 & 50 & 60 & 70 \end{bmatrix} \quad (47)$$

使用一个简单的灰度结构元素 b :

$$b = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (48)$$

计算过程示例: 以位置 $(2, 2)$ 为例 (即原图中值为 50 的位置):

- 将结构元素 b 的中心对准 $f(2, 2)$

- b 覆盖的 f 区域为: $\begin{bmatrix} 30 & 40 & 50 \\ 40 & 50 & 60 \\ 50 & 60 & 70 \end{bmatrix}$

- 对应相加: $\begin{bmatrix} 30+0 & 40+1 & 50+0 \\ 40+1 & 50+1 & 60+1 \\ 50+0 & 60+1 & 70+0 \end{bmatrix} = \begin{bmatrix} 30 & 41 & 50 \\ 41 & 51 & 61 \\ 50 & 61 & 70 \end{bmatrix}$

- 取最大值: $\max(30, 41, 50, 41, 51, 61, 50, 61, 70) = 70$

灰度膨胀后的结果 $f \oplus b$ 为:

$$f \oplus b = \begin{bmatrix} 30 & 40 & 50 & 51 \\ 40 & 50 & 60 & 61 \\ 50 & 60 & 70 & 71 \\ 51 & 61 & 71 & 71 \end{bmatrix} \quad (49)$$

重要说明 需要注意的是, 灰度膨胀的结果不是简单地将原图像的每个像素值加上一个常数。正确的计算必须:

1. 将结构元素中心对准目标像素
2. 计算结构元素覆盖区域内每个像素值与对应结构元素值的和
3. 取所有这些和值中的最大值作为输出

为什么需要灰度膨胀? 灰度膨胀的主要困难在于:

- 需要处理连续的灰度值
- 需要保持图像的亮度特征
- 需要扩大亮区而不影响暗区

灰度膨胀通过以下方式解决这些问题:

- 使用最大值运算扩大亮区
- 通过结构元素控制膨胀范围
- 保持图像的相对亮度关系

7.5.2 灰度腐蚀

输出像素值由邻域内的最小值决定。这会使图像中的暗区扩大，整体变暗。

$$[f \ominus b](x, y) = \min_{(s, t) \in b} \{f(x + s, y + t) - b(s, t)\}$$

具体例子 使用上面的灰度图像 f 和结构元素 b 进行腐蚀。

计算过程示例：以位置 $(2, 2)$ 为例（即原图中值为 50 的位置）：

- 将结构元素 b 的中心对准 $f(2, 2)$

- b 覆盖的 f 区域为：

$$\begin{bmatrix} 30 & 40 & 50 \\ 40 & 50 & 60 \\ 50 & 60 & 70 \end{bmatrix}$$

- 对应相减：

$$\begin{bmatrix} 30 - 0 & 40 - 1 & 50 - 0 \\ 40 - 1 & 50 - 1 & 60 - 1 \\ 50 - 0 & 60 - 1 & 70 - 0 \end{bmatrix} = \begin{bmatrix} 30 & 39 & 50 \\ 39 & 49 & 59 \\ 50 & 59 & 70 \end{bmatrix}$$

- 取最小值： $\min(30, 39, 50, 39, 49, 59, 50, 59, 70) = 30$

让我们再验证一个位置 $(1, 1)$ 的计算：

- 将结构元素 b 的中心对准 $f(1, 1)$ （值为 30）

- b 覆盖的 f 区域为：

$$\begin{bmatrix} 10 & 20 & 30 \\ 20 & 30 & 40 \\ 30 & 40 & 50 \end{bmatrix}$$

- 对应相减：

$$\begin{bmatrix} 10 - 0 & 20 - 1 & 30 - 0 \\ 20 - 1 & 30 - 1 & 40 - 1 \\ 30 - 0 & 40 - 1 & 50 - 0 \end{bmatrix} = \begin{bmatrix} 10 & 19 & 30 \\ 19 & 29 & 39 \\ 30 & 39 & 50 \end{bmatrix}$$

- 取最小值： $\min(10, 19, 30, 19, 29, 39, 30, 39, 50) = 10$

通过逐个位置计算，灰度腐蚀后的**正确结果** $f \ominus b$ 为：

$$f \ominus b = \begin{bmatrix} 9 & 19 & 29 & 29 \\ 19 & 10 & 20 & 30 \\ 29 & 20 & 30 & 40 \\ 29 & 30 & 40 & 50 \end{bmatrix} \quad (50)$$

重要说明 灰度腐蚀同样**不是**简单的减法运算。正确的计算步骤是：

1. 将结构元素中心对准目标像素
2. 计算结构元素覆盖区域内每个像素值与对应结构元素值的差
3. 取所有这些差值中的最小值作为输出

计算验证要点：每个位置都必须严格按照公式计算。例如：

- 位置 (0,0)：只有部分结构元素在图像范围内，需要正确处理边界
- 位置 (1,1)：完整覆盖，计算结果为 10
- 位置 (2,2)：完整覆盖，计算结果为 30

为什么需要灰度腐蚀？ 灰度腐蚀的主要困难在于：

- 需要处理连续的灰度值
- 需要保持图像的亮度特征
- 需要扩大暗区而不影响亮区

灰度腐蚀通过以下方式解决这些问题：

- 使用最小值运算扩大暗区
- 通过结构元素控制腐蚀范围
- 保持图像的相对亮度关系

7.5.3 灰度开运算和闭运算

定义与二值情况类似，同样具有平滑轮廓、去除小亮/暗点的效果。

- 灰度开运算： $f \circ b = (f \ominus b) \oplus b$
- 灰度闭运算： $f \bullet b = (f \oplus b) \ominus b$

具体例子 使用上面的灰度图像 f 和结构元素 b 进行开运算和闭运算。

灰度开运算: $f \circ b = (f \ominus b) \oplus b$

- 首先对上面计算得到的腐蚀结果进行膨胀

$$\bullet \text{ 腐蚀结果: } f \ominus b = \begin{bmatrix} 9 & 19 & 29 & 29 \\ 19 & 10 & 20 & 30 \\ 29 & 20 & 30 & 40 \\ 29 & 30 & 40 & 50 \end{bmatrix}$$

- 对此结果再进行膨胀操作，得到开运算结果

灰度开运算后的结果 $f \circ b$ 为:

$$f \circ b = \begin{bmatrix} 19 & 29 & 39 & 40 \\ 29 & 30 & 40 & 50 \\ 30 & 40 & 50 & 60 \\ 30 & 40 & 50 & 60 \end{bmatrix} \quad (51)$$

灰度闭运算: $f \bullet b = (f \oplus b) \ominus b$

- 首先对原图进行膨胀，然后对结果进行腐蚀

$$\bullet \text{ 膨胀结果: } f \oplus b = \begin{bmatrix} 30 & 40 & 50 & 51 \\ 40 & 50 & 60 & 61 \\ 50 & 60 & 70 & 71 \\ 51 & 61 & 71 & 71 \end{bmatrix}$$

- 对此结果再进行腐蚀操作

灰度闭运算后的结果 $f \bullet b$ 为:

$$f \bullet b = \begin{bmatrix} 20 & 30 & 40 & 50 \\ 30 & 40 & 50 & 60 \\ 40 & 50 & 60 & 70 \\ 50 & 60 & 70 & 70 \end{bmatrix} \quad (52)$$

为什么需要灰度开运算和闭运算? 灰度开运算和闭运算的主要困难在于:

- 需要同时处理亮区和暗区
- 需要保持图像的整体亮度特征
- 需要去除噪声而不影响主要特征

灰度开运算和闭运算通过以下方式解决这些问题:

- 开运算去除小的亮区
- 闭运算去除小的暗区
- 保持图像的主要亮度特征

灰度形态学操作的应用

- 图像平滑和去噪
- 细节增强
- 纹理分析
- 图像分割
- 特征提取

注意力涣散了，来日再战形态学及其后面的内容

8 图像分割基础

8.1 图像分割的定义与目标

图像分割是数字图像处理的核心任务之一。其根本目标是将一幅图像根据某些准则（如灰度、颜色、纹理等）分解成多个互不重叠的区域或对象。简单来说，就是给图像中的每个像素打上一个标签，属于同一对象的像素拥有相同的标签。这个过程是实现高级图像分析（如对象识别、特征提取、场景理解）的基础。

图像分割主要依赖于图像像素强度的两个基本性质：

- **不连续性 (Discontinuity)**: 基于像素值的突变来划分区域，例如图像中物体的边缘。这引出了边缘检测。
- **相似性 (Similarity)**: 基于像素值的相似性将像素分组到同一区域。这引出了阈值法、区域生长等方法。

8.2 基于不连续性的分割：边缘检测

8.2.1 基本原理：利用导数检测不连续点

灰度的不连续性主要表现为孤立点、线和边缘。在离散的数字图像中，导数运算通过空间卷积/滤波来实现。我们将一个小的卷积核（也叫掩模或滤波器）滑过整个图像，实现对每个像素邻域的加权求和。

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

其中 $w(s, t)$ 是卷积核（滤波器）。用于边缘检测的卷积核，其系数之和通常为 0，这保证了在灰度平坦区域的响应为零。

8.2.2 一阶与二阶导数 (考点)

- **一阶导数**:

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x)$$

在灰度斜坡 (Ramp) 的起始点产生响应，并在整个斜坡区域保持非零；在灰度阶跃 (Step) 处产生一个强的尖峰响应。它通常会产生较粗的边缘。

- **二阶导数 (拉普拉斯算子)**:

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

– 对精细细节（如细线、孤立点）响应非常强烈。

- 在阶跃和斜坡边缘的两侧会产生一对方向相反的反应（一正一负），中间点的值接近或等于零。这个特性被称为过零点检测 (Zero-crossing)，可以用来精确定位边缘的中心。
- 二阶导数的符号可以判断边缘的过渡方向：从暗到亮或从亮到暗。

假设有一行像素，其灰度值序列如下：

Intensity: 5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 ...

- **平坦区 (Flat segment)**, e.g., ...0 0 0...:
 - 一阶导数: ...0, 0...
 - 二阶导数: ...0, 0...
 - 结论: 在灰度不变的区域，一阶和二阶导数均为零。
- **阶跃边缘 (Step edge)**, e.g., ...0 7...:
 - 一阶导数: ...0, 7, 0... (在变化处产生一个尖峰)
 - 二阶导数: ...0, 7, -7, 0... (产生一正一负的双边响应)
- **斜坡边缘 (Ramp edge)**, e.g., ...5 4 3 2 1 0...:
 - 一阶导数: ...-1, -1, -1, -1, -1, 0... (在整个斜坡上产生一个平台状的非零响应)
 - 二阶导数: ...0, 0, 0, 0, 1, 0... (仅在斜坡的开始和结束处产生非零响应)
- **孤立点 (Isolated point)**, e.g., ...0 6 0...:
 - 一阶导数: ...6, -6... (产生一正一负的反应)
 - 二阶导数: ...6, -12, 6... (产生一个非常强烈的中心响应和两个较弱的旁瓣，响应比一阶导数强得多)
- **细线 (Line)**, e.g., ...0 1 3 1 0...:
 - 一阶导数: ...1, 2, -2, -1...
 - 二阶导数: ...1, 1, -4, 1, 1... (中心响应非常强烈)

对比总结:

注意: 导数运算对噪声非常敏感，因此在进行边缘检测前通常需要先进行平滑处理。

特性	一阶导数	二阶导数
对阶跃边缘响应	强，产生单个粗边	产生双边响应，可通过过零点定位
对细线/孤立点响应	较弱	响应非常强烈，适合检测精细细节
对噪声敏感度	敏感	更敏感
边缘粗细	产生较粗的边缘	产生较细的边缘

8.2.3 点、线检测

- 点检测:

- 目的: 寻找图像中孤立的、与周围邻域差异极大的像素点。
- 方法: 利用二阶导数（拉普拉斯算子）对孤立点响应最强的特性。一个像素点的响应值 R 通过拉普拉斯掩模计算得到，如果其绝对值 $|R| \geq T$ (T 为预设阈值)，则该点被识别为孤立点。

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

对应的掩模为:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \text{或} \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

第二个掩模考虑了对角线方向，更为常用。

- 线检测: 使用不同方向的模板，对特定方向的线（通常是 1 像素宽）响应最强。

- 目的: 寻找特定方向的细线（通常为 1 像素宽）。
- 方法: 设计对特定方向响应最强的掩模。例如，水平线检测掩模在水平方向上的系数为正（如 2），而在其他方向为负（如-1），这样当掩模中心位于水平线上时会得到最大响应。

水平线	+45° 线	垂直线	-45° 线
$\begin{pmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{pmatrix}$	$\begin{pmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{pmatrix}$	$\begin{pmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$

8.2.4 边缘检测 (Edge Detection)

边缘是图像中两个区域之间的边界像素集合。主要有三种模型：阶跃边缘 (Ideal/step edge)、斜坡边缘 (Ramp edge) 和屋顶边缘 (Roof edge)。

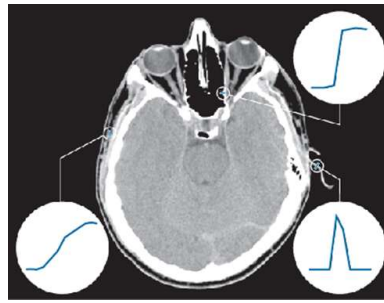


图 15: 边缘模型

- **阶跃边缘 (Ideal/step edge)**: 灰度值瞬间从一个值跳变到另一个值。
- **斜坡边缘 (Ramp edge)**: 灰度值在一段有限的距离内平缓地过渡。这是现实世界中最常见的边缘类型，通常由相机失焦或物体表面弯曲造成。
- **屋顶边缘 (Roof edge)**: 灰度值先上升再下降，形成一个屋顶状。常对应图像中的细线。

8.2.5 梯度算子 (一阶导数)

梯度是一个向量，指向图像灰度值变化最快的方向。简单的 $f(x+1) - f(x)$ 对噪声敏感且只考虑了一个方向。梯度算子通过在一个邻域（如 3×3 ）内计算差分并平均，来获得更稳定、更准确的梯度估计。

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

- **梯度幅值 (Magnitude)**: 表示边缘的强度或”陡峭”程度。幅值越大，边缘越明显。

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

- **梯度方向 (Angle)**: 指向灰度值变化最快的方向，这个方向垂直于边缘的方向。

$$\alpha(x, y) = \tan^{-1} \left(\frac{g_y}{g_x} \right)$$

常用梯度算子 (重要考点):

- **Roberts 算子**: 使用交叉差分，对角线边缘响应好。
 - 优点: 简单，能很好地检测对角线边缘。
 - 缺点: 对噪声非常敏感，定位精度不高。

$$g_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad g_y = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

- **Prewitt 算子**: 在提供梯度近似的同时带有一定的平滑作用。
 - 优点: 在计算梯度的同时, 进行了邻域平均, 起到了一定的平滑作用, 比 Roberts 算子抗噪性好。

$$g_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad g_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

- **Sobel 算子**: Prewitt 算子的改进, 对中心像素赋予更大权重, 平滑效果更好, 是最常用的算子之一。
 - 为什么比 Prewitt 更好?: Sobel 算子是 Prewitt 算子的改进。它在计算梯度时, 对距离中心像素更近的像素赋予了更大的权重 (权重为 2), 而较远的像素权重为 1。这种加权策略提供了更好的平滑效果, 能更有效地抑制噪声, 因此它是最常用的梯度算子之一。

$$g_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad g_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

计算示例: Sobel 算子边缘检测 (模拟考题)

给定一个 5×5 的图像块 I , 使用 Sobel 算子计算所有像素的梯度。

$$I = \begin{pmatrix} 50 & 55 & 60 & 70 & 80 \\ 55 & 60 & 70 & 80 & 90 \\ 0 & 70 & 150 & 90 & 100 \\ 65 & 75 & 100 & 100 & 110 \\ 70 & 80 & 100 & 110 & 120 \end{pmatrix}$$

步骤 1: 零填充边界

为了处理边界像素, 将图像填充到 7×7 :

$$I_{padded} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 55 & 60 & 70 & 80 & 0 \\ 0 & 55 & 60 & 70 & 80 & 90 & 0 \\ 0 & 0 & 70 & 150 & 90 & 100 & 0 \\ 0 & 65 & 75 & 100 & 100 & 110 & 0 \\ 0 & 70 & 80 & 100 & 110 & 120 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

步骤 2: 应用 Sobel 算子

使用 Sobel 水平算子 g_x 和垂直算子 g_y :

$$g_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad g_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

步骤 3: 计算水平梯度矩阵 G_x

对填充后图像的每个 3×3 邻域应用 g_x :

$$G_x = \begin{pmatrix} -50 & -60 & -70 & -80 & -80 \\ 5 & 20 & 20 & 20 & 10 \\ 130 & 150 & 30 & -60 & -10 \\ 70 & 85 & 25 & 10 & 10 \\ 70 & 80 & 100 & 110 & 120 \end{pmatrix}$$

计算示例 (中心像素 (2,2), 即原图中的 150): 邻域为: $\begin{pmatrix} 60 & 70 & 80 \\ 70 & 150 & 90 \\ 75 & 100 & 100 \end{pmatrix}$

$$G_x(2,2) = (-1)(60) + (0)(70) + (1)(75) + (-2)(70) + (0)(150) + (2)(100) + (-1)(80) + (0)(90) + (1)(100) = -60 + 75 - 140 + 200 - 80 + 100 = 95$$

步骤 4: 计算垂直梯度矩阵 G_y

$$G_y = \begin{pmatrix} 105 & 140 & 150 & 160 & 170 \\ -110 & -85 & 80 & 130 & 190 \\ -130 & -75 & 85 & 120 & 210 \\ -135 & -95 & 30 & 100 & 220 \\ -135 & -115 & -20 & 20 & 160 \end{pmatrix}$$

计算示例 (中心像素 (2,2)): $G_y(2,2) = (-1)(60) + (-2)(70) + (-1)(80) + (0)(70) + (0)(150) + (0)(90) + (1)(75) + (2)(100) + (1)(100) = -60 - 140 - 80 + 75 + 200 + 100 = 95$

步骤 5: 计算梯度幅值

对于中心像素 (2,2): - 精确计算: $M = \sqrt{95^2 + 95^2} = \sqrt{18050} \approx 134.4$ - 近似计算: $M \approx |95| + |95| = 190$ - 梯度方向: $\theta = \arctan(95/95) = 45^\circ$

注意: 对于图像边界的像素, 其邻域不完整, 需要进行边界填充 (Padding), 最常用的方法是零填充 (zero-padding), 即在图像外围补充一圈或多圈 0。

8.2.6 高斯-拉普拉斯 (LoG) 算子

我们已经知道, 标准的拉普拉斯 (二阶导数) 算子对边缘定位很准, 但它有两个致命的缺点:

- 对噪声极其敏感: 会将噪声误判为边缘。

- **产生双边响应:** 在一条边缘的两侧都会产生响应，这给边缘的提取带来了麻烦。

LoG 算子的提出就是为了解决这个问题。其核心思想非常直观：先平滑去噪，再检测边缘。它将高斯平滑和拉普拉斯边缘检测这两个步骤完美地结合在了一个算子中。

$$\nabla^2 h(r) = - \left[\frac{r^2 - \sigma^2}{\sigma^4} \right] e^{-\frac{r^2}{2\sigma^2}}, \quad \text{其中 } r^2 = x^2 + y^2$$

这个函数形似一顶“墨西哥草帽”， σ 控制平滑程度。 σ 越大，平滑效果越强，检测到的边缘尺度也越大。

根据卷积的性质，用 LoG 算子对图像进行卷积，等价于先用高斯函数对图像进行平滑，然后再对平滑后的结果应用拉普拉斯算子：

$$\nabla^2 G * f(x, y) = \nabla^2 [G * f(x, y)]$$

这保证了在检测边缘之前，噪声已经被有效地抑制了。边缘则通过寻找 LoG 响应结果中的过零点 (Zero-crossing) 来定位。

其 5×5 的近似掩模为：

$$\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

医学图像例子（肿瘤分割）极好地说明了 σ 的作用：

- **当 σ 很小 (small σ) 时:** 平滑程度低，LoG 算子会检测出肿瘤内部大量的精细纹理和细节，结果显得非常杂乱。
- **当 σ 很大 (large σ) 时:** 平滑程度高，内部的细节被模糊掉，LoG 算子能够检测出肿瘤的宏观轮廓。

结论: σ 的选择决定了检测到的边缘的尺度。没有唯一的“正确” σ 值，它的选择取决于你的应用目标。

8.2.7 Canny 边缘检测 (非常重要考点)

Canny 算子被认为是目前最优的边缘检测算法之一，它力求满足三个准则：

1. **低错误率 (Good Detection):** 尽可能地检测所有真实存在的边缘（高召回率），同时尽可能地避免将非边缘点（如噪声）检测为边缘（高精确率）。
2. **精确定位 (Good Localization):** 检测到的边缘像素应尽可能地靠近真实边缘的中心。

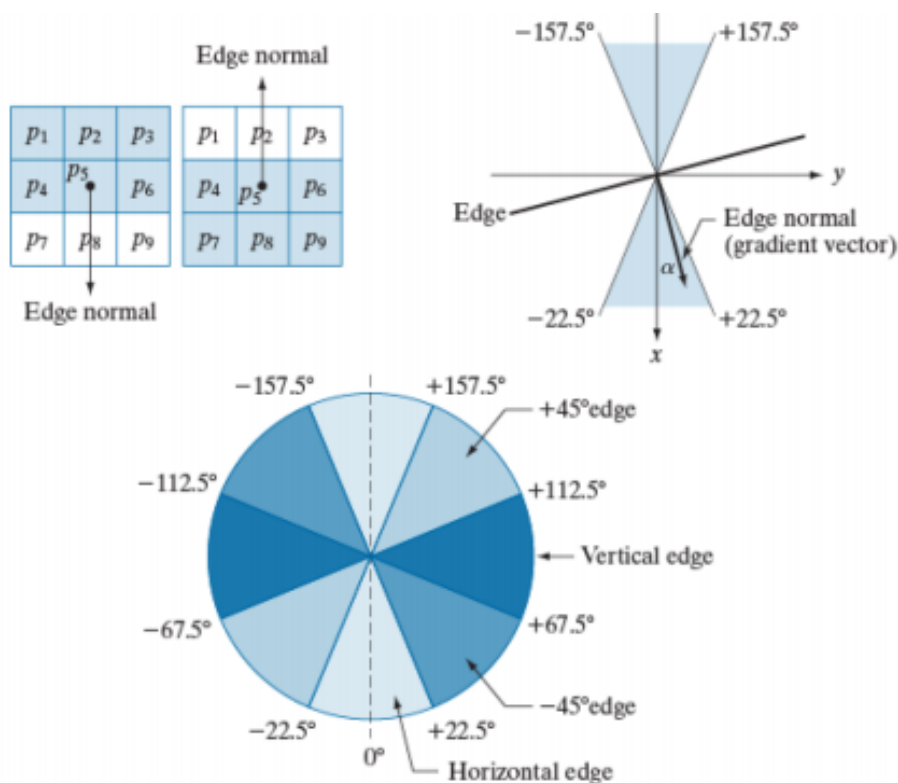


图 16: Canny

3. **单一响应 (Single Response):** 对于单个真实边缘，算法只应产生一个像素宽的响应，避免产生多重响应或粗边缘。

Canny 算法步骤:

1. **高斯平滑:** 使用高斯滤波器平滑图像以去除噪声。
2. **计算梯度:** 使用 Sobel 等算子计算图像的梯度幅值 $M(x, y)$ 和方向 $\alpha(x, y)$ 。
3. **非极大值抑制 (Non-Maximum Suppression):**
 - **目的:** 经过上一步得到的梯度幅值图像中的边缘仍然是模糊和粗壮的。NMS 的目的是将这些粗壮的边缘“瘦身”，使其变为单像素宽的细线，以满足“单一响应”准则。
 - **方向量化过程 (如图所示):**
 - 将连续的梯度方向 $\alpha(x, y)$ 量化到 4 个离散方向:
 - * **水平边缘 (Horizontal edge):** $-22.5^\circ \leq \alpha < 22.5^\circ$ 或 $157.5^\circ \leq \alpha < -157.5^\circ$
 - * **+45° 边缘:** $22.5^\circ \leq \alpha < 67.5^\circ$
 - * **垂直边缘 (Vertical edge):** $67.5^\circ \leq \alpha < 112.5^\circ$
 - * **-45° 边缘:** $112.5^\circ \leq \alpha < 157.5^\circ$
 - 注意: 梯度方向是垂直于边缘方向的，所以水平边缘对应垂直梯度方向。

- 邻居比较与抑制（如图左侧所示）：
 - 对于每个像素 (x, y) ，找到与其梯度方向 $\alpha(x, y)$ 最接近的方向 dk
 - 沿着 dk 方向，比较当前像素的梯度幅值 K 与两个邻居像素的幅值
 - 如果 K 小于沿 dk 方向的一个或两个邻居的值：
 - * $g_N(x, y) = 0$ （抑制）
 - * 否则： $g_N(x, y) = K$ （保留）
- 具体实现：
 - 水平边缘：比较上下邻居 $g(x, y - 1)$ 和 $g(x, y + 1)$
 - 垂直边缘：比较左右邻居 $g(x - 1, y)$ 和 $g(x + 1, y)$
 - $+45^\circ$ 边缘：比较 $g(x - 1, y - 1)$ 和 $g(x + 1, y + 1)$
 - -45° 边缘：比较 $g(x - 1, y + 1)$ 和 $g(x + 1, y - 1)$

4. 双阈值法与边缘连接（滞后阈值）：

- 滞后阈值 (Hysteresis Threshold)：使用两个阈值而非单一阈值
- 设置一个高阈值 T_H 和一个低阈值 T_L （通常 $T_H = 2 \sim 3 \times T_L$ ）
- 梯度幅值 $> T_H$ 的像素被视为**强边缘点** (Strong edges)
- 梯度幅值 $< T_L$ 的像素被认为是噪声，直接抑制（舍弃）
- 梯度幅值在 T_L 和 T_H 之间的像素为**弱边缘点** (Weak edges)
- 只有当弱边缘点与强边缘点 **8 连通**时，才被保留为边缘
- 这一步利用了边缘的连续性，有效去除了噪声引起的假边缘

关键理解：

- 梯度方向 vs 边缘方向：梯度方向垂直于边缘方向，指向灰度变化最快的方向
- 非极大值抑制的本质：沿梯度方向进行”极大值检测”，只保留局部最大值
- 滞后阈值的优势：避免了单一阈值可能造成的边缘断裂或噪声保留问题

8.3 基于相似性的分割：阈值法

8.3.1 阈值处理 (Thresholding) 基础

与寻找”差异”的边缘检测不同，阈值法是一种基于”相似性”的分割技术。它的思想非常简单：将灰度值相似的像素划分为同一类。其基本形式为：

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases}$$

其中 T 是阈值。

- **全局阈值 (Global Thresholding):** T 在整幅图像上是一个常数。
- **局部阈值 (Local Thresholding):** T 的值依赖于像素的邻域属性 $p(x, y)$ 。
- **动态/自适应阈值 (Dynamic/Adaptive Thresholding):** T 的值依赖于像素的空间坐标 (x, y) 。

主要挑战:

- **噪声:** 噪声会“污染”直方图, 使得前景和背景的灰度分布发生重叠, 原本直方图清晰的“双峰”可能会变得模糊, 难以找到一个理想的谷底作为阈值。
- **不均匀光照:** 当图像光照不均时 (如 PPT 中展示的, 图像一半亮一半暗), 背景的灰度值在不同区域差异巨大。此时, 任何一个单一的全局阈值都无法同时正确分割所有区域, 必然会导致部分区域分割失败。

8.3.2 Otsu's 方法 (最佳全局阈值法 - 大津法)

当一幅图像的直方图呈现明显的双峰 (bimodal) 形态时, 我们知道最佳阈值应该在两个峰之间的谷底。但如何自动、精确地找到这个“最佳”位置呢? Otsu 法就是为了解决这个问题。

它的核心思想是: 寻找一个阈值 k , 使得被这个阈值分开的前景类和背景类, 它们两个类之间的方差最大 (Maximize the between-class variance)。类间方差越大, 意味着这两个类的灰度分布差异越大, 分割得越开、越好。

算法步骤:

1. 计算图像的归一化直方图 $p_i = n_i/N$, 其中 n_i 是灰度级 i 的像素数, N 是总像素数。
2. 假设阈值为 k , 将像素分为两组: $C_0 = \{0, 1, \dots, k\}$ 和 $C_1 = \{k+1, \dots, L-1\}$ 。
3. 计算两组的概率 (零阶矩):

$$\omega_0(k) = \sum_{i=0}^k p_i \quad \text{和} \quad \omega_1(k) = \sum_{i=k+1}^{L-1} p_i = 1 - \omega_0(k)$$

4. 计算两组的平均灰度值 (一阶矩):

$$\mu_0(k) = \frac{\sum_{i=0}^k i \cdot p_i}{\omega_0(k)} \quad \text{和} \quad \mu_1(k) = \frac{\sum_{i=k+1}^{L-1} i \cdot p_i}{\omega_1(k)}$$

5. 遍历所有可能的阈值 k from 0 to $L-1$, 计算类间方差 $\sigma_B^2(k)$:

$$\sigma_B^2(k) = \omega_0(k)(\mu_0(k) - \mu_T)^2 + \omega_1(k)(\mu_1(k) - \mu_T)^2 = \omega_0(k)\omega_1(k)(\mu_0(k) - \mu_1(k))^2$$

其中 $\mu_T = \sum_{i=0}^{L-1} i \cdot p_i$ 是整幅图像的总平均灰度。

6. 使得 $\sigma_B^2(k)$ 最大的 k 值即为最佳阈值。

示例：手动计算 Otsu 阈值 (模拟考题)

使用 PPT 作业中的直方图数据进行计算。

灰度级 (i)	0	1	2	3	4	5	6	7
像素数 (n_i)	50	50	40	20	10	10	10	10

Step 1: 准备工作

总像素数 $N = 50 + 50 + 40 + 20 + 10 + 10 + 10 + 10 = 200$ 。

归一化概率 $p_i = n_i/200$: $p = [0.25, 0.25, 0.20, 0.10, 0.05, 0.05, 0.05, 0.05]$

Step 2: 迭代计算类间方差

我们尝试几个可能的阈值 k 。

尝试 $k = 1$:

背景类 $C_0 = \{0, 1\}$, 前景类 $C_1 = \{2, 3, 4, 5, 6, 7\}$

$$\omega_0 = p_0 + p_1 = 0.25 + 0.25 = 0.50 \quad (53)$$

$$\omega_1 = 1 - \omega_0 = 0.50 \quad (54)$$

$$\mu_0 = (0 \cdot 0.25 + 1 \cdot 0.25)/0.50 = 0.5 \quad (55)$$

$$\mu_1 = (2 \cdot 0.20 + 3 \cdot 0.10 + \cdots + 7 \cdot 0.05)/0.50 \quad (56)$$

$$= (0.4 + 0.3 + 0.2 + 0.25 + 0.3 + 0.35)/0.5 = 1.8/0.5 = 3.6 \quad (57)$$

$$\sigma_B^2(1) = 0.5 \cdot 0.5 \cdot (0.5 - 3.6)^2 = 0.25 \cdot (-3.1)^2 = 0.25 \cdot 9.61 = 2.4025 \quad (58)$$

尝试 $k = 2$:

背景类 $C_0 = \{0, 1, 2\}$, 前景类 $C_1 = \{3, 4, 5, 6, 7\}$

$$\omega_0 = p_0 + p_1 + p_2 = 0.25 + 0.25 + 0.20 = 0.70 \quad (59)$$

$$\omega_1 = 1 - \omega_0 = 0.30 \quad (60)$$

$$\mu_0 = (0 \cdot 0.25 + 1 \cdot 0.25 + 2 \cdot 0.20)/0.70 = (0 + 0.25 + 0.4)/0.7 = 0.65/0.7 \approx 0.928 \quad (61)$$

$$\mu_1 = (3 \cdot 0.10 + \cdots + 7 \cdot 0.05)/0.30 = (0.3 + 0.2 + 0.25 + 0.3 + 0.35)/0.3 = 1.4/0.3 \approx 4.667 \quad (62)$$

$$\sigma_B^2(2) = 0.7 \cdot 0.3 \cdot (0.928 - 4.667)^2 = 0.21 \cdot (-3.739)^2 \approx 0.21 \cdot 13.98 = 2.9358 \quad (63)$$

尝试 $k = 3$:

背景类 $C_0 = \{0, 1, 2, 3\}$, 前景类 $C_1 = \{4, 5, 6, 7\}$

$$\omega_0 = 0.70 + 0.10 = 0.80 \quad (64)$$

$$\omega_1 = 1 - \omega_0 = 0.20 \quad (65)$$

$$\mu_0 = (0.65 + 3 \cdot 0.10)/0.80 = 0.95/0.8 = 1.1875 \quad (66)$$

$$\mu_1 = (4 \cdot 0.05 + \dots + 7 \cdot 0.05)/0.20 = (0.2 + 0.25 + 0.3 + 0.35)/0.2 = 1.1/0.2 = 5.5 \quad (67)$$

$$\sigma_B^2(3) = 0.8 \cdot 0.2 \cdot (1.1875 - 5.5)^2 = 0.16 \cdot (-4.3125)^2 \approx 0.16 \cdot 18.6 = 2.976 \quad (68)$$

通过继续计算，我们会发现当 $k = 3$ 时，类间方差达到最大值。

结论：该图像的最佳全局阈值为 $T = 3$ 。

8.3.3 阈值法的改进

问题：如果直方图不是理想的双峰，或者噪声干扰严重，Otsu 法也可能失败。

改进策略：

- **利用平滑：**对原图先进行平滑滤波（如均值滤波），可以有效抑制噪声，使直方图的峰谷特征更明显，从而提高 Otsu 法的准确性。
- **利用边缘信息：**这是一个非常巧妙的方法。我们只选择那些位于物体边缘附近的像素点来计算直方图，因为这些像素最能代表前景和背景的灰度分界。
 - **具体做法：**先用 Sobel 等算子计算图像的梯度图，然后只保留梯度值较高的像素（即边缘像素），最后仅基于这些像素的灰度值来计算直方图并应用 Otsu 法。

8.3.4 自适应阈值处理

当图像光照不均匀时，全局阈值法会失效。自适应阈值法的思想是：

1. 将图像分割成若干个子图像。
2. 对每个子图像独立地计算一个最佳阈值（例如使用 Otsu 方法）。
- 3.（可选）对计算出的阈值曲面进行插值，为原图中每个像素提供一个平滑变化的阈值。

这种方法能很好地适应光照变化，但缺点是可能在子图像的边界处产生不连续的块状效应。更高级的自适应方法会对每个像素计算一个阈值，该阈值由其邻域的统计特性（如均值、标准差）决定，从而得到更平滑的分割结果。

8.4 作业重点提示

- **Otsu 方法计算:** 务必掌握根据给定的直方图, 手动计算类间方差并找出最佳阈值的过程。这是核心考点。
- **修改 Sobel/Prewitt 核:** 理解梯度算子方向性的本质。例如, 要检测 $\pm 45^\circ$ 方向的边缘, 需要将梯度分别定义在对角线方向上。例如, 对于 $+45^\circ$ 边缘, 其梯度方向为 -45° , 因此 g_{+45} 应该在 -45° 方向上进行差分。
- **边缘检测全流程:** 掌握从原图开始, 应用 Sobel 算子 (注意处理边界, 如零填充), 计算梯度幅值和方向, 并进行后续处理 (如阈值化) 的完整步骤。

9 图像分割算法

9.1 前期内容回顾

在之前的课程中，我们学习了图像分割的基本概念，即把一幅图像根据某些准则分解成其构成部分或对象的过程。图像分割主要依赖于图像像素灰度值的两大基本属性：

- **不连续性 (Discontinuity)**: 这种方法旨在寻找图像灰度值的急剧变化处，即边缘。我们已经学习了多种基于不连续性的分割技术，包括各类边缘检测算子和边缘连接与边界检测方法。
- **相似性 (Similarity)**: 这种方法旨在将具有相似性质（如灰度、颜色、纹理）的像素聚合在一起，形成一个区域。我们已经探讨了其中最基础的方法——阈值法。

本次课程将深入探讨更多基于相似性的高级分割算法，主要包括基于区域的方法和基于聚类的方法。

9.2 图像分割算法概览

图像分割算法可以大致分为监督式和非监督式两大类。

- **监督式分割 (Supervised)**: 需要预先标注的数据进行训练，常见算法有 K-近邻 (KNN)、支持向量机 (SVM)、随机森林、决策树、人工神经网络 (ANN) 等。
- **非监督式分割 (Unsupervised)**: 无需训练数据，直接根据图像自身特征进行分割。这是我们本次课程的重点，主要包含四大类别：
 1. **基于阈值的方法 (Thresholding Methods)**: 如 Otsu 法、自适应阈值法等。
 2. **基于边缘的方法 (Edge-based Methods)**: 如 Sobel、Canny、高斯-拉普拉斯 (LoG) 算子等。
 3. **基于区域的方法 (Region-based Methods)**: 本次讲解的重点之一，包括区域生长法和区域分裂合并法。此外，分水岭算法也是一种重要的基于区域的分割技术。
 4. **基于聚类的方法 (Clustering Methods)**: 本次讲解的另一个重点，包括 K-均值 (K-Means)、模糊 C 均值 (Fuzzy C-Means) 和均值漂移 (Mean-Shift) 等。

本次课程将聚焦于上图用红色圆圈标出的两大核心非监督方法：**基于区域的方法和基于聚类的方法**。

9.3 基于区域的分割方法 (Region-based Methods)

基于区域的分割方法直接根据预定义的相似性准则，将像素或子区域组合成更大的区域。

9.3.1 区域生长法 (Region Growing)

基本思想与关键要素 区域生长是一种程序化的、从“种子”点开始逐步聚合邻近像素的分割方法。其基本流程是：首先选择一组“种子”点，然后将与这些种子点属性相似的邻近像素不断地附加到种子所在的区域中，直至无法再增长为止。

该算法的成功与否高度依赖于以下三个**关键要素**：

1. **种子点的选择 (Selection of the seeds)**: 种子点是区域生长的起点。其选择可以直接由用户交互式指定，也可以通过某些自动化的方法（例如，对图像进行阈值处理后得到的区域质心）来确定。种子点的数量和位置直接影响最终的分割结果。
2. **相似性准则的定义 (Criteria of similarity)**: 这是判断邻域像素是否应该被并入当前区域的依据。相似性准则可以非常灵活，常见的包括：
 - **灰度值**: 要求邻域像素的灰度值与种子点或所在区域的平均灰度值之差在一个预设的阈值内。
 - **颜色/纹理**: 对于彩色图像或纹理图像，可以使用颜色向量距离或纹理描述符的相似度作为准则。
 - **连接性**: 通常使用 4-连接或 8-连接来定义像素的邻域关系。
3. **停止规则的制定 (Formulation of a stopping rule)**: 区域生长过程需要一个明确的停止条件。当遍历完所有邻域像素后，再也没有满足相似性准则的像素可以被添加到任何区域时，算法停止。

区域生长法的一个显著优点是能够生成连通性好、边界平滑的区域，并且其分割准则可以根据具体问题灵活定义。但它的主要缺点是对种子点的选择和噪声非常敏感，不同的种子点可能会导致截然不同的分割结果。

9.3.2 区域分裂与合并法 (Region Splitting and Merging)

这种方法旨在克服区域生长法对种子点选择的依赖性。它采用了一种“自顶向下”再“自底向上”的策略，不依赖任何初始种子点，而是从全局视角审视图像的均匀性。该方法是区域生长的逆过程，它从整幅图像出发，通过递归地分裂不满足均匀性准则的区域，然后再合并相邻的相似区域来完成分割。这种方法通常使用**四叉树 (Quadtree)** 数据结构来实现。

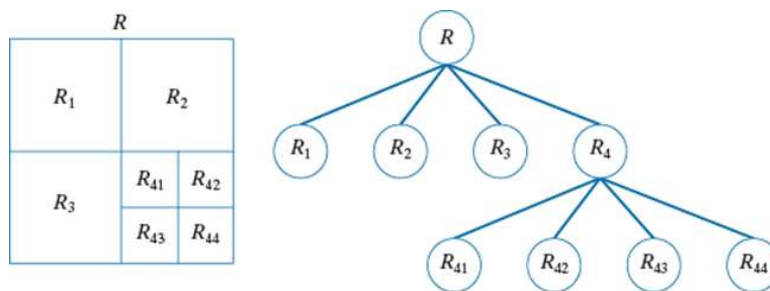


图 17: 区域分裂与合并法流程

算法流程 该算法包含分裂和合并两个阶段:

1. 分裂阶段 (Splitting):

- 从整幅图像 R 开始, 定义一个逻辑谓词 (均匀性准则) $Q(R)$ 。例如, $Q(R)$ 可以定义为” 区域 R 内所有像素的灰度值标准差小于某个阈值”。
- 如果 $Q(R)$ 为 ‘FALSE’, 说明该区域不均匀, 则将其分裂成四个大小相等的子象限 (quadrant) R_1, R_2, R_3, R_4 。
- 对每个子象限递归地重复此分裂过程, 直到所有子区域都满足谓词 Q 为 ‘TRUE’。这个过程的结果可以用一个四叉树来表示, 树的每个叶节点对应一个满足均匀性准则的最小区域块。

2. 合并阶段 (Merging):

- 纯粹的分裂过程可能会导致一些相邻的、属性相同的区域被人为地分开了。因此, 需要一个合并步骤。
- 遍历所有相邻的区域对 (R_i, R_j) , 如果它们的合并区域满足均匀性准则, 即 $Q(R_i \cup R_j)$ 为 ‘TRUE’, 则将它们合并成一个区域。

3. 停止条件: 当一轮遍历中没有任何分裂或合并操作可以再进行时, 算法终止。

相比区域生长, 分裂合并法对种子点的选择不敏感, 因为它是一个” 自顶向下” 再” 自底向上” 的过程。但它的缺点是分裂过程固定为四等分, 可能会导致分割边界呈块状, 不够自然。

9.4 分水岭算法 (Watershed Algorithm)

9.4.1 基本思想与地形学类比

该算法的核心思想是将灰度图像看作一个三维的地形景观, 其中 (x, y) 坐标代表地理位置, 灰度值 $I(x, y)$ 代表该位置的海拔高度。在这个地形上:

- **局部最小值 (Minima):** 对应地形中的” 盆地” 或” 山谷” 的最低点。

- **汇水盆地 (Catchment Basin)**: 如果一滴水落在这个区域的任何一点, 它最终都会汇集到同一个局部最小值点。这个区域就是该最小值的汇水盆地或分水岭。
- **分水岭线 (Watershed Lines)**: 这是地形中的”山脊线”。落在这条线上的水滴, 会等概率地流向两个或多个不同的汇水盆地。

分割的目标就是找到这些分水岭线。这些线构成了不同区域之间的边界。通常, 我们不是在原始图像上直接应用分水岭算法, 而是在其**梯度图像**上应用。在梯度图中, 高梯度值 (边缘) 对应高海拔的”山脊”, 而低梯度值 (平坦区域) 对应低海拔的”盆地”。

9.4.2 算法流程: ”大坝构建”法

分水岭算法的一个直观理解方式是”浸水”或”大坝构建”过程:

1. 想象在每个局部最小值点的位置打一个洞。
2. 让水以均匀的速率从这些洞中涌入, 开始”淹没”整个地形。
3. 随着水位上升, 来自不同盆地的水会逐渐汇合。在两个不同盆地的水即将接触的瞬间, 立即在它们之间修建一道”大坝”以阻止它们合并。
4. 当整个地形都被淹没后, 这些修建起来的大坝就构成了最终的分水岭线, 也就是分割边界。

9.4.3 具体实现步骤与示例

算法的具体实现是一个迭代过程, 按灰度级从小到大处理像素:

1. 首先, 找到所有灰度值最低的像素点 (或梯度图中的梯度最小点)。这些点构成了初始的汇水盆地。
2. 然后, 从最低灰度级 k 开始, 逐级向上处理。
3. 对于当前灰度级 k 的所有像素, 检查其邻域:
 - 如果一个像素只与一个已标记的汇水盆地区域相邻, 则将该像素归入此区域。
 - 如果一个像素与两个或多个不同的已标记区域相邻, 则该点位于”山脊”上, 标记为**边界点** (分水岭线的一部分)。
 - 如果一个像素不与任何已标记区域相邻, 则它形成一个新的局部最小值, 创建一个新的汇水盆地。
4. 重复此过程, 直到处理完所有灰度级的像素。

示例: 分水岭算法手动执行

9.4.4 问题与改进：过分割与标记 (Markers)

分水岭算法的一个主要缺点是**过分割 (Oversegmentation)**。由于噪声或图像局部纹理的细微变化，梯度图中会产生大量无意义的局部最小值，导致算法最终分割出远超预期的、非常多的小区域。

解决过分割问题的有效方法是使用**标记 (Markers)**。标记是一种先验知识，用于告诉算法哪些地方是“确定”的区域内部，哪些是“确定”的背景或边界。

- **内部标记 (Internal Markers)**: 位于我们感兴趣的对象内部的像素点或区域。它们被用来强制指定唯一的局部最小值，从而引导分水岭算法的淹没过程。
- **外部标记 (External Markers)**: 通常是指示背景或对象之间确定边界的像素线。

使用标记的分水岭流程一般如下：

1. 首先，通过一些简单的方法（如阈值处理、形态学操作等）初步确定内部标记和外部标记。例如，可以对原图进行阈值处理得到对象的大致区域作为内部标记。
2. 然后，计算原图的梯度图。
3. 修改梯度图，将标记点的位置强制设为最小值，从而保证水从且仅从这些标记点开始淹没。
4. 在修改后的梯度图上运行分水岭算法，这样分割结果就会受到标记的约束，大大减少过分割现象。

这种基于标记的分水岭方法在医学图像分割中应用非常广泛，能够有效分割出粘连的细胞、肿瘤等复杂结构。

9.5 基于聚类的分割方法 (Clustering-based Methods)

聚类是一种经典的非监督机器学习方法，其目标是将数据集中的样本划分为若干个簇 (cluster)，使得同一个簇内的样本彼此相似，而不同簇的样本相异。在图像分割中，样本就是图像的像素，像素的特征可以是灰度值、RGB 颜色向量、纹理特征，甚至可以包含像素的空间坐标。

9.5.1 K-均值聚类 (K-Means Clustering)

K-均值是最著名的划分式聚类算法之一。它的目标是找到 K 个簇，使得所有样本点到其所属簇的中心（均值）的平方误差和 (Sum of Squared Errors, SSE) 最小。该误差函数定义为：

$$D = \sum_{k=1}^K \sum_{x_i \in C_k} (x_i - m_k)^2$$

其中 K 是预先设定的簇的数量， C_k 是第 k 个簇， m_k 是簇 C_k 的均值（质心）。

算法步骤 K-均值是一个迭代算法，其步骤如下：

1. **初始化**: 随机选择 K 个数据点作为初始的簇中心 m_1, m_2, \dots, m_K 。
2. **分配步骤**: 对于数据集中的每一个样本点 x_i ，计算它到所有 K 个簇中心的距离（通常是欧氏距离），并将其分配给距离最近的那个簇。
3. **更新步骤**: 在所有样本点都被分配完毕后，重新计算每个簇的中心。新的中心就是该簇内所有样本点的均值。

$$m_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

4. **重复与收敛**: 重复第 2 步和第 3 步，直到簇的分配不再发生变化，或者簇中心的变化小于一个很小的阈值，即算法收敛。

优缺点分析 (重要考点) K-均值算法因其简单高效而广受欢迎，但也有其固有的局限性。

优点:

- **简单、快速**: 算法原理简单，易于实现，计算复杂度相对较低，对于大规模数据集处理速度快。
- **收敛性**: 算法保证收敛到一个局部最优解。

缺点/问题:

- **K 值的设定**: 必须预先指定簇的数量 K ，而这个 K 值在实际问题中往往是未知的。
- **对初始中心敏感**: 不同的初始簇中心可能会导致算法收敛到不同的局部最优解，从而得到不同的聚类结果。
- **对异常值敏感**: 异常值 (outliers) 会显著地影响簇中心的计算，从而可能导致错误的聚类结果。
- **只能发现球状簇**: 由于其基于均值和欧氏距离，K-均值倾向于发现大小相似、密度均匀的球状（或凸形）簇，对于非球状、任意形状的簇则无能为力。

K 值的选择与评估 如何选择一个合适的 K 值是 K-均值应用中的一个核心问题。常用的方法有：

- **“肘部”法则 (Elbow Method / L-Curve)**: 尝试多个不同的 K 值，并对每个 K 值计算其最终的平方误差和 D 。将 D 随 K 变化的曲线绘制出来，该曲线通常会急剧下降然后趋于平缓，其“肘部”对应的 K 值就是一个不错的选择。

- **轮廓分析 (Silhouette Analysis):** 轮廓系数是一种衡量样本点与其所属簇的紧密程度以及与其他簇的分离程度的指标。对于样本点 i ，其轮廓系数 s_i 定义为：

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

其中 a_i 是 i 与其所属簇内其他所有点的平均距离（簇内不相似度）， b_i 是 i 与其最近的另一个簇中所有点的平均距离（簇间不相似度）。 s_i 的取值范围是 $[-1, 1]$ ，值越接近 1，说明聚类效果越好。通过计算所有样本的平均轮廓系数，可以评估不同 K 值下的聚类质量。

9.6 SLIC 超像素 (SLIC Superpixels)

在图像分割任务中，直接对像素的颜色或灰度进行 K -均值聚类，会忽略像素的空间位置信息，导致分割结果在空间上非常零碎。SLIC (Simple Linear Iterative Clustering) 算法是对 K -均值的一个巧妙改进，旨在生成紧凑、均匀、大致遵循图像边界的**超像素**。

SLIC 的核心思想是，在聚类时同时考虑像素的**颜色相似性**和**空间邻近性**。它将每个像素的特征向量从单纯的颜色（如 ‘r,g,b’）扩展到了一个五维向量 ‘r,g,b,x,y’，其中 ‘(x,y)’ 是像素的坐标。此外，为了提高效率，SLIC 并不会对每个像素计算其与所有簇中心的距离，而是将搜索范围限制在每个簇中心周围的一个 $2S \times 2S$ 的局部区域内，其中 S 是超像素的预期网格尺寸。这使得 SLIC 的计算复杂度与像素数量成线性关系，非常高效。

9.7 分割评估与验证

分割评估与验证是评估分割算法性能的重要步骤。常用的评估指标包括：

- **Dice 系数:** 用于评估分割结果与真实标签之间的重叠程度。
- **Jaccard 相似系数:** 用于评估分割结果与真实标签之间的重叠程度。
- **豪斯多夫距离 (Hausdorff Distance):** 用于评估分割结果与真实标签之间的距离。

10 医学图像配准与可视化

10.1 图像配准 (Image Registration) 概述

10.1.1 什么是图像配准及其重要性

图像配准是指寻找一个空间变换，将一幅图像（称为**浮动图像/源图像**, Moving/Source Image）上的点，映射到另一幅图像（称为**固定图像/目标图像**, Fixed/Target Image）中的对应点，从而使两幅图像在空间上对齐的过程。在医学图像分析中，配准是不可或缺的关键技术，其应用场景十分广泛：

配准的主要应用场景

- **时间序列分析**: 对同一患者在不同时间点采集的图像进行配准（如今天和一个月前），用于监测疾病进展、评估治疗效果或观察术后恢复情况。这种纵向分析对于慢性疾病的管理和治疗方案的调整具有重要意义。
- **运动伪影校正**: 校正患者在扫描过程中的运动，例如在功能磁共振成像 (fMRI) 中，需要将所有时间点的图像对齐。即使是微小的头动（毫米级别）也会严重影响功能激活的检测，因此运动校正是 fMRI 预处理的必要步骤。
- **多模态图像融合**: 将来自不同成像设备（如 CT、PET、MRI）的图像进行配准，以结合不同模态提供的互补信息（如解剖结构与功能代谢信息）。例如，CT 提供高分辨率的骨骼结构信息，而 PET 提供代谢活动信息，两者结合可以更准确地定位病灶。
- **群体分析与图谱应用**: 将不同患者的图像配准到一个“标准”或“平均”的大脑模板上（如 MNI152 模板），以消除个体解剖差异，从而进行跨被试的群体统计分析。这是现代神经影像学研究的基础。

从数学角度看，图像配准的目标是，给定源图像中的点集 $\{x_i\}$ 和目标图像中的对应点集 $\{x'_i\}$ ，寻找一个最优的变换函数 f 及其参数 p ，使得 $x'_i = f(x_i; p)$ 。

10.1.2 图像配准的基本方法分类

图像配准技术可以根据其依赖的图像特征分为三类：

基于特征的配准方法

1. **基于点的方法 (Point-based)**: 依赖于在两幅图像中手动或自动识别出的同源特征点 (Landmarks)。算法的目标是找到一个能使这些对应特征点之间距离最小化的变换。这种方法直观且计算速度快，但其准确性高度依赖于特征点定位的精度和数量。常用的特征点包括解剖标志点（如前联合、后联合）和几何特征点。

2. **基于表面的方法 (Surface-based)**: 依赖于从图像中分割出的解剖结构表面（如大脑皮层表面、骨骼表面等）。算法通过最小化两个表面之间的距离来寻找最佳变换。这种方法特别适用于具有明显表面特征的结构配准。
3. **基于强度的方法 (Intensity-based)**: 直接利用图像全部或部分体素的灰度值信息，通过优化一个衡量两幅图像强度分布相似性的**度量函数 (Metric)** 来寻找最佳变换。这种方法无需预先分割或特征提取，应用最为广泛，也是本课程的重点。

10.1.3 图像配准基本流程 (Registration Pipeline)

基于强度的图像配准过程是一个迭代优化的循环。其核心流程包含三大基本组件：

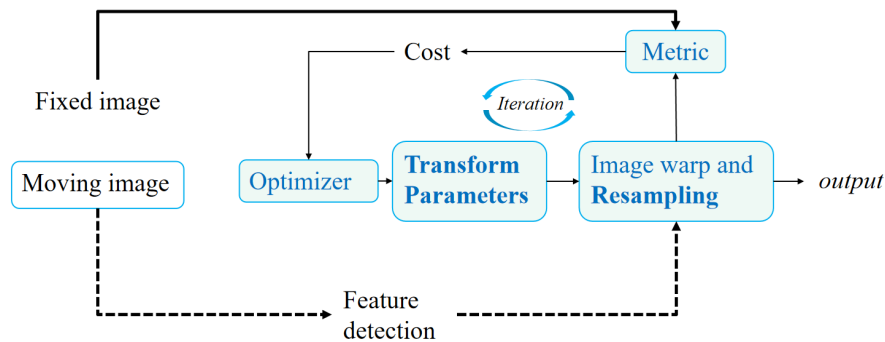


图 18: 图像配准基本流程

配准的三大核心组件

1. **变换模型 (Transformation Model)**: 定义了源图像如何进行空间变形。变换模型的选择决定了配准的灵活性和复杂度。
2. **相似性度量 (Similarity Metric)**: 也称为目标函数或代价函数 (Cost Function)，用于量化当前变换下两幅图像的对齐程度。度量函数的选择主要取决于图像的模态关系。
3. **优化器 (Optimizer)**: 是一种搜索算法，它通过不断调整变换模型的参数，来最大化相似性度量（或最小化代价函数），从而找到最佳的变换参数。

整个流程如下：从一个初始的变换参数开始，将源图像进行**变换和重采样 (Warp and Resampling)**，然后计算变换后的源图像与固定图像之间的相似性度量。优化器根据该度量值（或其梯度）来更新变换参数，并开始下一次迭代，直到度量值达到最优或满足收敛条件为止。

10.2 变换模型详解 (Transformation Model)

变换模型描述了从源图像空间到目标图像空间的映射关系，其复杂性决定了配准的灵活性和自由度。

10.2.1 线性变换 (Linear Transformation)

线性变换保持了图像中直线的”直线性”，可以用一个矩阵来表示。根据变换的复杂程度和保持的几何性质，线性变换形成了一个层次结构。

线性变换的层次结构 线性变换按照其复杂度和自由度可以分为以下几个层次，每个层次都包含了前一层次的所有变换类型：

1. 平移 (Translation): 3 个自由度
2. 刚性变换 (Rigid): 6 个自由度 (平移 + 旋转)
3. 相似变换 (Similarity): 7 个自由度 (刚性 + 均匀缩放)
4. 仿射变换 (Affine): 12 个自由度 (相似 + 非均匀缩放 + 切变)
5. 投影变换 (Projective): 15 个自由度 (仿射 + 投影效应)

仿射变换矩阵结构 (重要考点) 仿射变换是医学图像配准中最常用的线性变换，其通用的 4×4 齐次变换矩阵具有如下结构：

$$\mathbf{T} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \text{缩放} & \text{旋转} & \text{平移} \\ \text{切变} & & \\ 0 & 0 & 1 \end{pmatrix}$$

其中，矩阵的不同区域具有明确的几何意义：

- 左上角 2×2 子矩阵: 包含旋转、缩放和切变信息
- 右上角列向量: 包含平移信息 $(a_{13}, a_{23})^T$
- 底部行: 保持为 $(0, 0, 1)$ 以维持仿射性质

基础几何变换详解

1. 平移变换 (Translation) 平移是最简单的空间变换，仅改变物体的位置，不改变其大小、形状或方向。

数学表示：

$$\mathbf{x}' = \mathbf{x} + \mathbf{t}$$

其中 $\mathbf{t} = (t_x, t_y)^T$ 是平移向量。

变换矩阵：

$$\mathbf{T}_{trans} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

应用场景：校正患者在扫描床上的位置偏移，或者对齐不同时间点扫描时患者的位置差异。

2. 旋转变换 (Rotation) 旋转变换围绕某个固定点（通常是原点）旋转物体，保持物体的大小和形状不变。

二维旋转矩阵（逆时针旋转角度 θ ）：

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

三维旋转：需要分别绕 X、Y、Z 轴的旋转矩阵组合：

$$\mathbf{R}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (69)$$

$$\mathbf{R}_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (70)$$

$$\mathbf{R}_z(\gamma) = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (71)$$

应用场景：校正患者头部的转动，在 fMRI 扫描中校正头动引起的旋转。

3. 缩放变换 (Scaling) 缩放变换改变物体的大小，可以是均匀缩放（各方向缩放比例相同）或非均匀缩放。

均匀缩放矩阵（缩放因子 s ）：

$$\mathbf{S}_{uniform} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

非均匀缩放矩阵（各方向缩放因子 s_x, s_y ）：

$$\mathbf{S}_{non-uniform} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

应用场景：补偿不同扫描仪的像素尺寸差异，或调整不同患者大脑的整体尺寸差异。

4. 切变变换 (Shearing) 切变（错切）变换使得平行线保持平行，但角度发生改变，产生“倾斜”效果。

水平切变矩阵（切变参数 h ）：

$$\mathbf{H}_x = \begin{pmatrix} 1 & h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

垂直切变矩阵（切变参数 v ）：

$$\mathbf{H}_y = \begin{pmatrix} 1 & 0 & 0 \\ v & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

应用场景：校正由于磁场不均匀或梯度线性度问题导致的图像几何畸变。

5. 反射变换 (Reflection) 反射变换产生镜像效果，常用于左右脑的对称性分析。

水平反射矩阵（绕 Y 轴反射）：

$$\mathbf{F}_x = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

垂直反射矩阵（绕 X 轴反射）：

$$\mathbf{F}_y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

应用场景：脑图像的左右半球对称性分析，或处理不同成像协议下的左右翻转问题。

6. 投影变换 (Projection) 投影变换是最通用的线性变换，包含透视效应，主要用于处理相机几何或特殊的成像几何。

投影变换矩阵：

$$\mathbf{P} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

其中底部的 (a_{31}, a_{32}) 不再是零，这引入了透视效应。

应用场景：处理某些特殊的医学成像几何（如 X 光透视成像），或进行图像到物理空间的投影校正。

刚性变换 (Rigid Transformation) 刚性变换是最常用的基础变换，只包含平移和旋转，共 6 个自由度（3 个平移，3 个旋转）。这种变换保持了物体的大小和形状不变，是”最保守”的变换类型。

数学表示：

$$\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t}$$

其中 \mathbf{R} 是 3×3 旋转矩阵， \mathbf{t} 是 3×1 平移向量。

齐次坐标表示：

$$\begin{pmatrix} \mathbf{x}' \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

关键性质：

- 保持距离不变（等距变换）
- 保持角度不变
- 保持体积不变
- 旋转矩阵 \mathbf{R} 是正交矩阵： $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ ， $\det(\mathbf{R}) = 1$

应用场景：

- fMRI 时间序列的运动校正（头动校正）
- 同一患者不同时间点图像的对齐
- PET-CT 等多模态图像的刚性配准初始化

仿射变换 (Affine Transformation) 仿射变换在刚性变换的基础上，增加了缩放和切变，共 12 个自由度。它能够处理更复杂的几何变形，但仍保持平行线的平行性。

TABLE 2.2
Affine transformations based on Eq. (2.6.–23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \sin \theta + w \cos \theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

图 19: 仿射变换示意矩阵

数学表示:

$$\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{t}$$

其中 **A** 是 3×3 仿射矩阵，包含旋转、缩放和切变信息。

矩阵分解: 任何仿射变换矩阵都可以分解为基础变换的组合:

$$\mathbf{A} = \mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{H}$$

其中 **T** 是平移，**R** 是旋转，**S** 是缩放，**H** 是切变。

关键性质:

- 保持平行线的平行性
- 保持线段长度的比例关系
- 变换椭圆形为椭圆形（圆形可能变为椭圆形）
- 不保持角度和距离

应用场景:

- 不同被试大脑向标准空间的线性配准
- 消除扫描仪间的系统性几何差异
- 多模态图像的初步对齐
- 图像预处理中的几何校正

12 个自由度的详细分解:

1. 平移: 3 个参数 (t_x, t_y, t_z)
2. 旋转: 3 个参数 (绕 X、Y、Z 轴的旋转角度)
3. 缩放: 3 个参数 (X、Y、Z 方向的缩放因子)
4. 切变: 3 个参数 (XY、XZ、YZ 平面的切变)

仿射变换的矩阵乘法顺序 (重要考点) 在实际应用中, 仿射变换通常是多个基础变换的复合。理解矩阵乘法的顺序至关重要:

变换顺序规则:

$$\mathbf{x}' = \mathbf{T}_n \cdot \mathbf{T}_{n-1} \cdot \dots \cdot \mathbf{T}_2 \cdot \mathbf{T}_1 \cdot \mathbf{x}$$

核心原则: 先执行的变换放在右边, 后执行的变换放在左边

直观理解:

- 矩阵乘法是从右往左计算的
- \mathbf{T}_1 首先作用于点 \mathbf{x}
- 然后 \mathbf{T}_2 作用于 $\mathbf{T}_1\mathbf{x}$ 的结果
- 以此类推...

常见变换序列示例: 如果要对一个点先进行旋转, 再进行缩放, 最后进行平移, 则变换矩阵为:

$$\mathbf{T}_{total} = \mathbf{T}_{trans} \cdot \mathbf{S}_{scale} \cdot \mathbf{R}_{rotate}$$

仿射变换矩阵的分解与合成 当需要将一个复杂的仿射变换分解为多个简单变换时:

分解原理: 任何仿射变换矩阵都可以分解为:

$$\mathbf{A} = \mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{H}$$

其中变换的执行顺序是: 切变 \rightarrow 缩放 \rightarrow 旋转 \rightarrow 平移

SVD 分解方法: 对于仿射矩阵的 2×2 子矩阵部分:

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

其中:

- \mathbf{U} 和 \mathbf{V}^T 包含旋转信息
- Σ 包含缩放信息

小形变的线性近似：当变换接近恒等变换时，可以使用线性近似：

$$(\mathbf{I} + \delta\mathbf{A}) \cdot (\mathbf{I} + \delta\mathbf{B}) \approx \mathbf{I} + \delta\mathbf{A} + \delta\mathbf{B}$$

这个近似在非线性配准的增量更新中非常有用。

仿射变换计算例题 [仿射变换的复合与分解] 给定以下变换序列：先绕原点逆时针旋转 45° ，然后在 X 方向缩放 2 倍，最后平移 $(3, 4)$ 。

要求计算：

1. 写出各个基础变换矩阵
2. 计算复合变换矩阵
3. 计算点 $(1, 0)$ 变换后的坐标
4. 验证变换顺序的重要性

解题步骤：

步骤 1：各基础变换矩阵

旋转矩阵（ 45° 逆时针）：

$$\mathbf{R} = \begin{pmatrix} \cos(45^\circ) & -\sin(45^\circ) & 0 \\ \sin(45^\circ) & \cos(45^\circ) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

缩放矩阵（X 方向 2 倍）：

$$\mathbf{S} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

平移矩阵（ $(3, 4)$ ）：

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}$$

步骤 2：复合变换矩阵根据变换顺序（旋转 \rightarrow 缩放 \rightarrow 平移），复合矩阵为：

$$\mathbf{M} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{R}$$

先计算 $\mathbf{S} \cdot \mathbf{R}$:

$$\mathbf{S} \cdot \mathbf{R} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & -\sqrt{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

再计算 $\mathbf{T} \cdot (\mathbf{S} \cdot \mathbf{R})$:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & -\sqrt{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & -\sqrt{2} & 3 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 4 \\ 0 & 0 & 1 \end{pmatrix}$$

步骤 3: 点 $(1, 0)$ 的变换

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & -\sqrt{2} & 3 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 4 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} + 3 \\ \frac{\sqrt{2}}{2} + 4 \\ 1 \end{pmatrix}$$

因此变换后的坐标为 $(\sqrt{2} + 3, \frac{\sqrt{2}}{2} + 4) \approx (4.414, 4.707)$

步骤 4: 验证变换顺序如果改变顺序为: 平移 \rightarrow 缩放 \rightarrow 旋转, 则:

$$\mathbf{M}' = \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{T}$$

这会得到完全不同的结果, 说明变换顺序的重要性。

10.2.2 非线性变换 (Non-linear Transformation)

为了对齐大脑沟回等更精细的、高度可变的解剖结构, 需要使用非线性变换。非线性变换允许图像发生局部、不均匀的形变, 其自由度非常高。

形变场表示

- **形变场 (Deformation Field)**: 非线性变换通常用一个形变场 \mathbf{u} 来表示, 该场为图像中的每个点都指定了一个位移向量。
- **弹性模型 (Elastic)** 与 **流体模型 (Fluid)**: 这是两种常见的非线性形变物理模型, 分别模拟固体的弹性变形和流体的流动变形。
- **微分同胚变换 (Diffeomorphic Transformation)**: 这是目前非线性配准研究中的黄金标准, 保证变换的可逆性和拓扑保持性。

10.2.3 图像重采样与插值 (Resampling and Interpolation)

当应用一个空间变换后，源图像网格上的点会被映射到目标图像网格的非整数坐标位置。为了得到变换后图像在目标网格整数坐标上的强度值，必须进行插值。

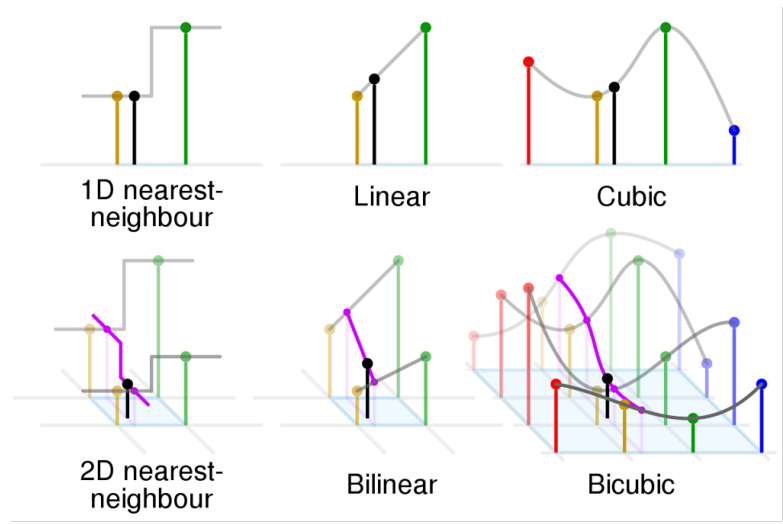


图 20: 插值示意图

常见插值方法 (重要考点)

- **最近邻插值 (Nearest Neighbor):** 将非整数坐标点的灰度值设为其最近的整数坐标点的灰度值。速度最快，但会产生块状效应，适用于标签图像。
- **双线性/三线性插值 (Bilinear/Trilinear):** 使用周围 4 个 (2D) 或 8 个 (3D) 点的灰度值进行线性加权平均。效果平滑，是速度和精度的良好折衷，最常用。
- **三次卷积/B 样条插值 (Cubic/B-spline):** 使用周围 16 个 (2D) 或 64 个 (3D) 点的灰度值进行更高阶的多项式拟合。效果最好，能更好地保留图像细节，但计算成本也最高。

双线性插值计算例题 [双线性插值] 给定一个 3×3 矩阵，使用双线性插值计算指定坐标点的灰度值：

y	$x = 0$	$x = 1$	$x = 2$
2	80	90	100
1	40	50	60
0	10	20	30

要求计算：使用双线性插值计算以下点的灰度值：

1. $P_1(0.5, 0.5)$

2. $P_2(1.3, 1.7)$

3. $P_3(1.8, 0.2)$

双线性插值公式：

$$f(x, y) = f(0, 0)(1 - x)(1 - y) + f(1, 0)x(1 - y) + f(0, 1)(1 - x)y + f(1, 1)xy$$

解题步骤：

对于 $P_1(0.5, 0.5)$ ：周围四个格点： $(0, 0) = 10$, $(1, 0) = 20$, $(0, 1) = 40$, $(1, 1) = 50$

$$\begin{aligned} f(0.5, 0.5) &= 10(1 - 0.5)(1 - 0.5) + 20(0.5)(1 - 0.5) + 40(1 - 0.5)(0.5) + 50(0.5)(0.5) \\ &= 10 \times 0.25 + 20 \times 0.25 + 40 \times 0.25 + 50 \times 0.25 = 30 \end{aligned}$$

对于 $P_2(1.3, 1.7)$ ：周围四个格点： $(1, 1) = 50$, $(2, 1) = 60$, $(1, 2) = 90$, $(2, 2) = 100$

$$\begin{aligned} x &= 1.3 - 1 = 0.3, \quad y = 1.7 - 1 = 0.7 \\ f(1.3, 1.7) &= 50(1 - 0.3)(1 - 0.7) + 60(0.3)(1 - 0.7) + 90(1 - 0.3)(0.7) + 100(0.3)(0.7) \\ &= 50 \times 0.21 + 60 \times 0.09 + 90 \times 0.49 + 100 \times 0.21 \\ &= 10.5 + 5.4 + 44.1 + 21 = 81 \end{aligned}$$

对于 $P_3(1.8, 0.2)$ ：周围四个格点： $(1, 0) = 20$, $(2, 0) = 30$, $(1, 1) = 50$, $(2, 1) = 60$

$$\begin{aligned} x &= 1.8 - 1 = 0.8, \quad y = 0.2 - 0 = 0.2 \\ f(1.8, 0.2) &= 20(1 - 0.8)(1 - 0.2) + 30(0.8)(1 - 0.2) + 50(1 - 0.8)(0.2) + 60(0.8)(0.2) \\ &= 20 \times 0.16 + 30 \times 0.64 + 50 \times 0.04 + 60 \times 0.16 \\ &= 3.2 + 19.2 + 2 + 9.6 = 34 \end{aligned}$$

10.3 相似性度量详解 (Similarity Metric) (重要考点)

相似性度量是配准流程的”眼睛”，它告诉优化器当前的对齐效果有多好。其选择主要取决于待配准图像的模态关系。

10.3.1 单模态 (Intra-modality) 配准度量

适用于配准来自相同模态的图像，例如 T1w 到 T1w，或者 fMRI 时间序列的运动校正。这些度量通常假设两幅图像的强度值存在简单的线性关系。

常用单模态度量

- 差值平方和 (Sum of Squared Differences, SSD):

$$SSD = \frac{1}{N} \sum_i^N |A(i) - B'(i)|^2$$

该度量假设两幅图像的强度差异仅由高斯噪声引起。它计算简单，但对强度尺度的变化和噪声非常敏感。

- 比率图像均匀性 (Ratio-Image Uniformity, RIU): 适用于两幅图像之间存在一个乘性增益因子的场景。它通过计算比率图像的标准差与均值的比值来衡量一致性。

计算步骤:

1. 确定体素总数: N

2. 计算比率图像: $R(i) = \frac{B'(i)}{A(i)}$

3. 计算比率图像的均值:

$$\mu_R = \frac{1}{N} \sum_{i=1}^N R(i)$$

4. 计算比率图像的标准差:

$$\sigma_R = \sqrt{\frac{1}{N} \sum_{i=1}^N (R(i) - \mu_R)^2}$$

5. 计算归一化标准差 (RIU):

$$RIU = \frac{\sigma_R}{\mu_R}$$

RIU 值越小，表明两幅图像的比率越均匀，配准效果越好。理想情况下，如果两幅图像完全对齐且只相差一个常数倍数因子，则 RIU 应接近 0。

- 相关系数 (Correlation Coefficient, CC):

$$CC = \frac{\sum_i (A(i) - \bar{A})(B'(i) - \bar{B}')}{\sqrt{\sum_i (A(i) - \bar{A})^2 \sum_i (B'(i) - \bar{B}')^2}}$$

该度量假设两幅图像的强度值之间存在线性关系，对线性的亮度/对比度变化具有鲁棒性，是单模态配准中非常稳健和常用的方法。

10.3.2 多模态 (Inter-modality) 配准度量: 互信息

当配准不同模态的图像时 (如 CT 到 MRI, 或 T1w 到 T2w), 组织在两幅图像中的强度关系不再是简单的线性关系, 甚至可能是复杂的非单调关系。此时, 上述度量均会失效。基于信息论的互信息 (Mutual Information, MI) 是解决该问题的金标准。

信息论基础 (考点)

1. **信息量**: 一个事件所包含的信息量与其发生的概率成反比。一个非常不可能的事件（概率 $p(x)$ 很小）发生了，会带来很大的信息量（惊喜程度高）。信息量定义为 $I(x) = -\log(p(x))$ 。
2. **熵 (Entropy)**: 一个随机变量的熵是其信息量的数学期望，衡量了该变量的不确定性。熵越大，不确定性越高。

$$H(X) = E[I(x)] = - \sum_x p(x) \log p(x)$$

3. 联合熵与条件熵:

- **联合熵** $H(X, Y)$: 衡量一对随机变量 (X, Y) 总的不确定性。
 - **条件熵** $H(Y|X)$: 在已知 X 的条件下， Y 还剩下的不确定性。
4. **互信息 (Mutual Information, MI)**: MI 衡量了两个随机变量之间的相互依赖程度，或者说，一个变量中包含了多少关于另一个变量的信息。其定义式有多种等价形式：

$$\begin{aligned} I(X; Y) &= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned}$$

MI 的直观解释是：知道一个变量 Y 后，另一个变量 X 不确定性的减小量。如果 X 和 Y 相互独立，则 MI 为 0。

MI 在图像配准中的应用

- 将两幅待配准图像的强度值看作两个随机变量 A 和 B 。
- 计算它们的**联合直方图 (Joint Histogram)**，归一化后得到联合概率分布 $p(a, b)$ 。
- 当两幅图像对齐得很好时，一种组织（如灰质）在 A 图中的强度值会稳定地对应于 B 图中的某种强度值。这会使得联合直方图中的点簇非常集中和紧凑。
- 一个高度集中的联合直方图意味着联合熵 $H(A, B)$ 很小。根据公式 $I(A; B) = H(A) + H(B) - H(A, B)$ ，由于 $H(A)$ 和 $H(B)$ （由图像内容决定，不随变换改变）是常数，**最小化联合熵等价于最大化互信息**。
- 因此，图像配准的目标就变成了：**寻找一个空间变换，使得两幅图像的互信息最大化**。

MI 不依赖于图像强度的具体函数关系，只依赖于其统计依赖性，因此对多模态配准极为有效和鲁棒。

互信息计算例题 [信息论计算] 给定两幅图像 A 和 B 如下：

图像 A:

$$\begin{pmatrix} 1 & 3 & 1 \\ 1 & 3 & 1 \\ 1 & 3 & 1 \end{pmatrix}$$

图像 B:

$$\begin{pmatrix} 1 & 2 & 0 \\ 1 & 2 & 0 \\ 1 & 2 & 0 \end{pmatrix}$$

联合熵公式: $H(A, B) = -\sum_a \sum_b p_{AB}(a, b) \log p_{AB}(a, b)$

要求计算:

1. 计算各自的熵: $H(A), H(B)$
2. 计算联合熵: $H(A, B)$
3. 计算互信息: $I(A; B)$
4. 计算灰度分布 (灰度级从 0 到 3), 然后计算 Kullback-Leibler 距离 $K(A||B)$

解题步骤:

步骤 1: 统计灰度分布

- 图像 A: 灰度 1 出现 6 次, 灰度 3 出现 3 次, 总像素 9 个
- 图像 B: 灰度 0 出现 3 次, 灰度 1 出现 3 次, 灰度 2 出现 3 次, 总像素 9 个

步骤 2: 计算概率分布

- $P_A(1) = 6/9 = 2/3, P_A(3) = 3/9 = 1/3$
- $P_B(0) = 3/9 = 1/3, P_B(1) = 3/9 = 1/3, P_B(2) = 3/9 = 1/3$

步骤 3: 计算各自熵

$$H(A) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} \quad (72)$$

$$= -\frac{2}{3}(-0.585) - \frac{1}{3}(-1.585) = 0.390 + 0.528 = 0.918 \text{ bits} \quad (73)$$

$$H(B) = -3 \times \frac{1}{3} \log_2 \frac{1}{3} = -\log_2 \frac{1}{3} = \log_2 3 = 1.585 \text{ bits} \quad (74)$$

步骤 4: 构建联合直方图并计算联合熵联合概率: $(A = 1, B = 0) : 3/9, (A = 1, B = 1) : 3/9, (A = 3, B = 2) : 3/9$

$$H(A, B) = -3 \times \frac{1}{3} \log_2 \frac{1}{3} = 1.585 \text{ bits}$$

步骤 5: 计算互信息

$$I(A; B) = H(A) + H(B) - H(A, B) = 0.918 + 1.585 - 1.585 = 0.918 \text{ bits}$$

步骤 6: 计算 KL 散度

$$K(A||B) = \sum_i P_A(i) \log_2 \frac{P_A(i)}{P_B(i)}$$

由于 A 和 B 的支撑集不同, 需要扩展到完整的灰度空间 [0,3]:

$$K(A||B) = P_A(1) \log_2 \frac{P_A(1)}{P_B(1)} + P_A(3) \log_2 \frac{P_A(3)}{P_B(3)}$$

其中 $P_B(3) = 0$, 使得 KL 散度趋于无穷大。

KL 散度的物理意义: Kullback-Leibler 散度 $K(A||B)$ 衡量了两个概率分布之间的“距离”, 具体表示:

- **信息散度:** 从分布 B 编码分布 A 所需要的额外信息量
- **非对称性:** $K(A||B) \neq K(B||A)$, 因此不是真正的距离度量
- **非负性:** $K(A||B) \geq 0$, 当且仅当 $A = B$ 时等于 0
- **相对熵:** 衡量 A 相对于 B 的信息增益

在图像配准中, KL 散度可以用作非对称的相似性度量, 特别适用于有先验信息的配准场景。

其他相似性度量计算例题 为了更全面地理解各种相似性度量的计算方法, 我们再来看看 SSD、RIU 和相关系数的具体计算过程。

[单模态相似性度量计算] 给定两幅 3×3 图像如下:

固定图像 F:

$$\begin{pmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{pmatrix}$$

浮动图像 M (经过某种变换后):

$$\begin{pmatrix} 12 & 22 & 32 \\ 42 & 52 & 62 \\ 72 & 82 & 92 \end{pmatrix}$$

要求计算:

1. 差值平方和 (SSD)
2. 比率图像均匀性 (RIU)
3. 相关系数 (CC)
4. 分析哪种度量最适合这种图像对

解题步骤:

步骤 1: 计算 SSD

$$SSD = \frac{1}{N} \sum_{i=1}^N |F(i) - M(i)|^2$$

其中 $N = 9$ (总像素数)。

逐像素计算差值平方:

$$(10 - 12)^2 + (20 - 22)^2 + (30 - 32)^2 + (40 - 42)^2 + (50 - 52)^2 \quad (75)$$

$$+ (60 - 62)^2 + (70 - 72)^2 + (80 - 82)^2 + (90 - 92)^2 \quad (76)$$

$$= 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 = 36 \quad (77)$$

$$SSD = \frac{36}{9} = 4$$

步骤 2: 计算 RIU RIU 适用于两图像间存在乘性关系的情况。

计算比率图像:

$$R(i) = \frac{M(i)}{F(i)}$$

比率矩阵:

$$\mathbf{R} = \begin{pmatrix} 1.2 & 1.1 & 1.067 \\ 1.05 & 1.04 & 1.033 \\ 1.029 & 1.025 & 1.022 \end{pmatrix}$$

计算比率图像的均值:

$$\mu_R = \frac{1}{9}(1.2 + 1.1 + 1.067 + 1.05 + 1.04 + 1.033 + 1.029 + 1.025 + 1.022) = \frac{9.566}{9} = 1.063$$

计算比率图像的标准差:

$$\sigma_R^2 = \frac{1}{9} \sum_{i=1}^9 (R(i) - \mu_R)^2$$

逐项计算:

$$(1.2 - 1.063)^2 + (1.1 - 1.063)^2 + (1.067 - 1.063)^2 + \dots \quad (78)$$

$$= 0.0188 + 0.0014 + 0.000016 + 0.0002 + 0.0005 + 0.0009 + 0.0012 + 0.0015 + 0.0017 \quad (79)$$

$$= 0.0263 \quad (80)$$

$$\sigma_R = \sqrt{0.0263} = 0.162$$

计算 RIU:

$$RIU = \frac{\sigma_R}{\mu_R} = \frac{0.162}{1.063} = 0.152$$

步骤 3: 计算相关系数 (CC)

$$CC = \frac{\sum_{i=1}^N (F(i) - \bar{F})(M(i) - \bar{M})}{\sqrt{\sum_{i=1}^N (F(i) - \bar{F})^2 \sum_{i=1}^N (M(i) - \bar{M})^2}}$$

计算均值:

$$\bar{F} = \frac{10 + 20 + 30 + 40 + 50 + 60 + 70 + 80 + 90}{9} = \frac{450}{9} = 50$$

$$\bar{M} = \frac{12 + 22 + 32 + 42 + 52 + 62 + 72 + 82 + 92}{9} = \frac{468}{9} = 52$$

计算分子:

$$\sum_{i=1}^9 (F(i) - \bar{F})(M(i) - \bar{M})$$

逐项计算:

$$(10 - 50)(12 - 52) + (20 - 50)(22 - 52) + (30 - 50)(32 - 52) + \dots \quad (81)$$

$$= (-40)(-40) + (-30)(-30) + (-20)(-20) + (-10)(-10) + (0)(0) \quad (82)$$

$$+ (10)(10) + (20)(20) + (30)(30) + (40)(40) \quad (83)$$

$$= 1600 + 900 + 400 + 100 + 0 + 100 + 400 + 900 + 1600 = 6000 \quad (84)$$

计算分母:

$$\sum_{i=1}^9 (F(i) - \bar{F})^2 = 1600 + 900 + 400 + 100 + 0 + 100 + 400 + 900 + 1600 = 6000$$

$$\sum_{i=1}^9 (M(i) - \bar{M})^2 = 1600 + 900 + 400 + 100 + 0 + 100 + 400 + 900 + 1600 = 6000$$

计算 CC:

$$CC = \frac{6000}{\sqrt{6000 \times 6000}} = \frac{6000}{6000} = 1.0$$

步骤 4: 结果分析

- **SSD = 4:** 表明两图像有较小的强度差异
- **RIU = 0.152:** 比率图像相对均匀, 存在近似的乘性关系
- **CC = 1.0:** 完美的线性相关, 说明两图像存在精确的线性关系

度量选择建议：对于这种具有完美线性关系的图像对，**相关系数（CC）**是最佳选择，因为：

- 1. 它对线性强度变化完全鲁棒
- 2. 不受加性偏移和乘性缩放的影响
- 3. 计算效率相对较高
- 4. 在这种情况下提供了最稳定的相似性度量

[不同噪声水平下的度量比较] 为了进一步说明各种度量的鲁棒性，考虑在上述浮动图像 M 中加入不同水平的高斯噪声：

噪声水平 1（标准差 $\sigma = 1$ ）：

$$M_1 = M + \mathcal{N}(0, 1)$$

噪声水平 2（标准差 $\sigma = 5$ ）：

$$M_2 = M + \mathcal{N}(0, 25)$$

预期结果：

- **SSD**：对噪声最敏感，随噪声增加急剧恶化
- **RIU**：中等敏感，加性噪声会影响比率计算
- **CC**：对噪声相对鲁棒，特别是对于保持线性关系的噪声
- **MI**：对噪声最鲁棒，但计算复杂度最高

这个例子说明了在实际应用中选择合适的相似性度量的重要性。

10.3.3 相似性度量方法对比总结 (重要考点)

下表总结了各种相似性度量方法的特点、适用场景和性能对比：

表 10: 相似性度量方法对比				
代价函数	模态假设	鲁棒性	计算成本	最佳使用场景
SSD / MSE	单模态	低（噪声敏感）	低	相同模态，无强度变化
RIU	单模态（缩放）	中等	中等	相同模态，乘性差异
互相关	单模态/线性	高（线性变化）	中等	相同模态，亮度/对比度变化
互信息	多模态	非常高	高	不同模态

选择指导原则:

- **单模态配准:** 优先选择互相关 (CC), 因其对线性强度变化具有良好的鲁棒性
- **多模态配准:** 必须使用互信息 (MI), 这是唯一可靠的选择
- **计算效率要求高:** 可考虑 SSD, 但需确保图像预处理质量
- **存在乘性增益差异:** 使用 RIU 可以有效处理这类问题

10.4 优化器与配准策略 (Optimizer)

优化器的任务是在由变换参数定义的多维空间中, 系统地搜索能使相似性度量达到最优 (最大值或最小值) 的参数组合。

10.4.1 优化挑战与策略

- **挑战:** 相似性度量的曲面通常非常复杂, 充满了多个**局部最优解**。优化器的目标是找到**全局最优解**。
- **迭代搜索:** 由于通常不存在解析解, 优化过程必须是迭代的。
- **梯度下降法 (Gradient Descent):** 一种常见且基础的优化策略。它从一个初始参数点开始, 沿着代价函数梯度的**负方向**进行小步迭代, 因为这个方向是函数值下降最快的方向。

$$\mathbf{p}_{t+1} = \mathbf{p}_t - \eta \nabla C(\mathbf{p}_t)$$

其中 \mathbf{p} 是参数向量, η 是学习率 (步长), ∇C 是代价函数的梯度。

- **多分辨率策略:** 为了避免陷入局部最优, 一个非常有效的实践策略是采用**多分辨率/多尺度**方法。即先在非常粗糙 (降采样) 的图像上进行配准, 找到一个大致的全局最优解, 然后将此解作为更高分辨率图像配准的初始值, 逐级提高图像分辨率, 直至在原始分辨率上完成配准。这种由粗到精的策略能有效跳出局部陷阱, 提高配准的鲁棒性和效率。

10.5 非线性配准 (Non-linear Registration)

当需要对齐精细的、局部差异大的解剖结构时, 线性变换不足以胜任, 必须使用非线性配准。

10.5.1 微分同胚变换 (Diffeomorphism) (重要考点)

在现代计算解剖学中, **微分同胚 (Diffeomorphism)** 是一种理想的非线性变换模型。它是一个可微的、双射 (一对一) 的映射, 并且其逆映射也是可微的。这意味着:

微分同胚的重要性质

- **保持拓扑结构:** 变换是平滑的, 不会产生撕裂或折叠, 保证了变换前后解剖结构的连接关系不变 (例如, 皮层不会被撕开)。
- **可逆性:** 存在一个唯一的、平滑的逆变换, 可以从目标图像完美地映射回源图像。
- **生物学合理性:** 这些特性对于进行有生物学意义的形态学分析至关重要。

微分同胚变换通常通过将一个大形变分解为一系列无穷小的、平滑的形变的组合 (积分) 来实现。

10.5.2 非线性配准算法详解

Demons 算法 是一种基于光流理论快速非线性配准方法, 通过” 恶魔力” 推动图像像素对齐; **LDDMM 算法** 是基于微分几何的高精度配准方法, 通过测地流实现大形变。两者的核心区别在于 Demons 追求计算效率, 而 LDDMM 追求变形质量和理论严谨性。

Demons 算法 (重要考点)

算法概述与核心思想 Demons 算法是一种广泛使用的非参数可形变配准方法, 由 Thirion 在 1998 年提出。它将图像配准问题类比为物理学中的**光流 (Optical Flow)** 问题, 是一种基于强度梯度的迭代形变估计方法。

核心理念: 将固定图像和浮动图像之间的强度差异看作是一种” 力” 或” 恶魔 (Demons)”, 这些恶魔会” 推动” 浮动图像的像素点向固定图像的对应位置移动, 从而实现图像对齐。

光流理论基础 Demons 算法将待配准的图像视为运动序列:

- **固定图像 F :** $I(x, t - \Delta t)$
- **浮动图像 M :** $I(x, t)$

根据光流约束方程 (**Optical Flow Constraint**):

$$\nabla I(x, t) \cdot v(x, t) + I_t(x, t) = 0$$

其中:

- $\nabla I(x, t)$: 图像的空间梯度
- $v(x, t)$: 像素点的运动速度 (位移场)
- $I_t(x, t)$: 图像的时间梯度

设 $\Delta t = 1$ ，则时间梯度变为：

$$I_t(x, t) = \frac{I(x, t) - I(x, t - \Delta t)}{\Delta t} = M - F$$

因此光流方程简化为：

$$\nabla I(x, t) \cdot v(x, t) = F - M$$

经典 Demons 算法流程 算法步骤：

1. **初始化：** 设置初始位移场 s (通常为恒等变换)
2. **确定活跃点集：** 选择浮动图像 M 中空间梯度非零的所有像素点 D_M
3. **计算基本位移：** 对于 D_M 中的每个点 P ，计算其基本位移 $u(P)$
4. **计算图像力：** 计算推动 M 向 F 移动的力 u ，使得 $M \circ (s + u)$ 更接近 F
5. **更新位移场：** $s \leftarrow$ 对 $s + u$ 进行高斯平滑
6. **迭代：** 重复步骤 2-5 直到收敛

基本位移计算公式：

$$\tilde{f}(P) = \sum_{P' \in S} \frac{K(P, P')}{D(P, P')} P \vec{P'}$$

其中：

- P ：浮动图像中的点
- P' ：固定图像中的点
- K ：相似性准则函数
- D ：距离函数
- $\tilde{f}(P)$ ：点 P 应移动的方向

力的计算：恶魔推动机制 恶魔力的直观理解：

- 如果某点的像素值低于目标值：恶魔沿图像梯度方向推动
- 如果某点的像素值高于目标值：恶魔沿图像梯度负方向推动

优化目标函数：

$$\text{minimize } \frac{1}{2} \mathbf{v}^T \mathbf{v} \quad \text{s.t.} \quad \nabla I \cdot \mathbf{v} = -I_t$$

解得速度场：

$$\hat{\mathbf{v}} = \left(\frac{-I_t}{\|\nabla I\|^2} \right) \nabla I$$

这个公式给出了每个像素点应该移动的方向和幅度。

微分同胚 Demons 算法 为了保证变换的微分同胚性质，现代 Demons 算法采用了改进的更新策略：

改进的目标函数：

$$E_{corr}(u) = ||F - M \circ \exp(v)||^2 + ||\mathbf{v}||^2$$

其中：

- 第一项：数据项 (相似性约束)
- 第二项：正则化项 (平滑性约束)
- $\exp(v)$ ：指数映射，保证微分同胚性

扩散式正则化 (高斯平滑)：

$$s \leftarrow Id + K * (c - Id)$$

其中 K 是高斯核， c 是当前变换， Id 是恒等变换。

算法特点与优势 优势：

- 计算效率高：每次迭代的计算复杂度较低
- 易于实现：算法流程直观，实现相对简单
- 鲁棒性好：对初始化不敏感，收敛性较好
- 适用性广：可用于单模态和多模态配准

局限性：

- 依赖梯度信息：在图像梯度较弱的区域效果不佳
- 易陷入局部最优：可能收敛到局部解
- 对噪声敏感：图像噪声会影响梯度计算的准确性

适用场景：

- 同一被试不同时间点的纵向配准
- 功能图像的运动校正
- 解剖图像的精细对齐
- 作为其他复杂配准算法的初始化步骤

LDDMM 算法 (重要考点)

算法概述与理论基础 LDDMM (Large Deformation Diffeomorphic Metric Mapping) 是一种更为先进的、基于微分几何和流体力学框架的非线性配准算法。它将图像间的形变看作是一个测地流 (Geodesic Flow)，即在所有可能的微分同胚变换群中寻找一条“最短路径”。

核心理念：

- **微分几何视角：** 将图像配准问题转化为在微分同胚群上的测地线问题
- **连续时间演化：** 通过连续的时间演化过程实现大形变
- **能量最小化：** 同时优化形变的平滑性和配准精度

LDDMM 数学框架 目标函数：

$$\mathcal{E} = \frac{1}{2} \int_{t=0}^1 \|\mathbf{L}\mathbf{v}_t\|^2 dt + \frac{1}{2\sigma^2} \|f - \mu(\varphi_1^{-1})\|^2$$

其中：

- **第一项：正则化项** - 形变平滑性的度量
- **第二项：数据项** - 变形模板与个体扫描间的距离
- **L**：微分算子（通常是拉普拉斯算子）
- **v_t**：时间 *t* 处的速度场
- **φ₁**：最终的形变映射
- **σ²**：噪声方差参数

约束条件：

$$\varphi_0 = \text{Id}, \quad \frac{d\varphi}{dt} = \mathbf{v}_t(\varphi_t)$$

这个微分方程描述了形变的时间演化过程。

大形变的分解原理 LDDMM 的关键洞察是将大形变分解为一系列无穷小形变的组合：

$$\varphi_1 = \left(\text{Id} + \frac{\mathbf{v}_{t_{N-1}}}{N}\right) \circ \left(\text{Id} + \frac{\mathbf{v}_{t_{N-2}}}{N}\right) \circ \dots \circ \left(\text{Id} + \frac{\mathbf{v}_{t_1}}{N}\right) \circ \left(\text{Id} + \frac{\mathbf{v}_0}{N}\right)$$

逆变换的计算：

$$\vartheta_1 = \left(\text{Id} - \frac{\mathbf{v}_0}{N}\right) \circ \left(\text{Id} - \frac{\mathbf{v}_{t_1}}{N}\right) \circ \dots \circ \left(\text{Id} - \frac{\mathbf{v}_{t_{N-2}}}{N}\right) \circ \left(\text{Id} - \frac{\mathbf{v}_{t_{N-1}}}{N}\right)$$

小形变近似原理： 当形变接近恒等变换时，我们可以进行线性近似：

- $1.3 \times 0.8 = 1.04$ ，但 $1 + (1.3 - 1) + (0.8 - 1) = 1.1$

- $1.03 \times 0.98 = 1.0094$, 但 $1 + (1.03 - 1) + (0.98 - 1) = 1.01$
- $1.003 \times 0.998 = 1.000994$, 但 $1 + (1.003 - 1) + (0.998 - 1) = 1.001$

可见, 当形变很小时, 乘法可以用加法近似, 这为数值实现提供了基础。

Hilbert 梯度优化 LDDMM 通过 **Hilbert 梯度下降**进行优化:

$$\mathbf{v}_t^{(i+1)} = \mathbf{v}_t^{(i)} - c\mathbf{v}_t^{(i)} - cK \left(|\mathbf{D}\vartheta^{(i)}| (\mathbf{D}\vartheta^{(i)})^T \left(\frac{|\mathbf{D}\varphi_1^{(i)}|}{2} (\nabla\mu)(f \circ \varphi_t^{(i)} - \mu) \right) \circ \vartheta^{(i)} \right)$$

其中:

- c : 学习率参数
- K : 核函数 (通常为高斯核)
- \mathbf{D} : 雅可比矩阵
- $\nabla\mu$: 模板图像的梯度

算法特点与核心概念 三大核心特性:

1. 微分算子 (Differential Operator):

- 使用拉普拉斯算子或更高阶的微分算子
- 确保速度场的平滑性
- 控制形变的正则化程度

2. 大形变 (Large Deformations):

- 能够处理极大的非线性形变
- 通过时间积分避免雅可比行列式为零
- 保证变换的双射性质

3. 测地距离 (Geodesic Distances):

- 在微分同胚群上定义黎曼度量
- 寻找最短的形变路径
- 提供形态学上有意义的距离度量

时变速度场演化: LDDMM 的核心输出是一个随时间平滑演化的速度场, 它描述了从源图像到目标图像的连续变形过程。这个过程可以可视化为图像在时间轴上的”流动”。

表 11: Demons 算法与 LDDMM 算法对比

比较维度	Demons 算法	LDDMM 算法
理论基础	光流理论，基于图像梯度的局部优化	微分几何理论，基于测地流的全局优化
形变模型	离散的位移场更新	连续时间的速度场演化
微分同胚性	需要特殊策略保证（如指数映射）	天然保证微分同胚性质
计算复杂度	低，每次迭代计算简单	高，需要求解时变 PDE
收敛速度	快，通常几十次迭代	慢，可能需要数百次迭代
形变质量	中等，可能出现折叠	高，严格避免拓扑错误
可逆性	近似可逆	精确可逆
参数敏感性	相对鲁棒	对正则化参数较敏感
实现难度	简单，易于实现	复杂，需要专业数值方法
适用场景	实时应用，快速配准	高精度研究，形态学分析

Demons vs LDDMM 算法对比 (重要考点)

10.5.3 形变场分析：雅可比行列式与 TBM

非线性配准得到的核心结果是一个密集的形变场。通过分析这个形变场，我们可以量化大脑局部的形态差异。

雅可比分析 (重要考点)

- **雅可比矩阵 (Jacobian Matrix):** 形变场在某一点的雅可比矩阵描述了该点邻域的线性近似变换。
- **雅可比行列式 (Jacobian Determinant):** 雅可比矩阵的行列式 $|J|$ 度量了变换在该点引起的局部体积变化：
 - $|J| > 1$: 体积扩张 (expansion)
 - $|J| = 1$: 体积不变 (no change)
 - $|J| < 1$: 体积收缩 (contraction)
- **基于张量的形态学分析 (Tensor-Based Morphometry, TBM):** TBM 就是通过计算和分析雅可比行列式图，来在体素水平上定量比较不同组别之间大脑的局部体积差异。例如，在老年痴呆症研究中，可以通过 TBM 发现患者大脑中特定区域（如海马体）相比健康人存在显著的体积萎缩。

雅可比矩阵计算例题 [空间变换的雅可比分析] 给定空间变换如下：

$$\begin{cases} x' = e^{(x+3y)/4} \\ y' = e^{(3x+y)/4} \end{cases}$$

要求计算：

1. 计算雅可比矩阵
2. 计算雅可比行列式
3. 计算逆变换的雅可比矩阵
4. 计算逆变换的雅可比行列式

解题步骤：

步骤 1：计算雅可比矩阵雅可比矩阵定义为：

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial y} \\ \frac{\partial y'}{\partial x} & \frac{\partial y'}{\partial y} \end{pmatrix}$$

计算各偏导数：

$$\frac{\partial x'}{\partial x} = \frac{\partial}{\partial x} e^{(x+3y)/4} = \frac{1}{4} e^{(x+3y)/4} \quad (85)$$

$$\frac{\partial x'}{\partial y} = \frac{\partial}{\partial y} e^{(x+3y)/4} = \frac{3}{4} e^{(x+3y)/4} \quad (86)$$

$$\frac{\partial y'}{\partial x} = \frac{\partial}{\partial x} e^{(3x+y)/4} = \frac{3}{4} e^{(3x+y)/4} \quad (87)$$

$$\frac{\partial y'}{\partial y} = \frac{\partial}{\partial y} e^{(3x+y)/4} = \frac{1}{4} e^{(3x+y)/4} \quad (88)$$

因此雅可比矩阵为：

$$\mathbf{J} = \begin{pmatrix} \frac{1}{4} e^{(x+3y)/4} & \frac{3}{4} e^{(x+3y)/4} \\ \frac{3}{4} e^{(3x+y)/4} & \frac{1}{4} e^{(3x+y)/4} \end{pmatrix}$$

步骤 2：计算雅可比行列式

$$\begin{aligned} |\mathbf{J}| &= \frac{1}{4} e^{(x+3y)/4} \cdot \frac{1}{4} e^{(3x+y)/4} - \frac{3}{4} e^{(x+3y)/4} \cdot \frac{3}{4} e^{(3x+y)/4} \\ &= \frac{1}{16} e^{(x+3y+3x+y)/4} - \frac{9}{16} e^{(x+3y+3x+y)/4} \\ &= \frac{1-9}{16} e^{(4x+4y)/4} = -\frac{8}{16} e^{x+y} = -\frac{1}{2} e^{x+y} \end{aligned}$$

步骤 3：计算逆变换设逆变换为 $(u, v) \rightarrow (x, y)$ ，其中 $u = x', v = y'$

从原变换方程:

$$u = e^{(x+3y)/4}, \quad v = e^{(3x+y)/4}$$

取对数:

$$\ln u = \frac{x+3y}{4}, \quad \ln v = \frac{3x+y}{4}$$

解这个线性方程组:

$$4 \ln u = x + 3y, \quad 4 \ln v = 3x + y$$

解得:

$$x = \frac{4 \ln v - 3 \cdot 4 \ln u}{1 - 9} = \frac{4 \ln v - 12 \ln u}{-8} = \frac{3 \ln u - \ln v}{2}$$

$$y = \frac{4 \ln u - 3 \cdot 4 \ln v}{1 - 9} = \frac{4 \ln u - 12 \ln v}{-8} = \frac{3 \ln v - \ln u}{2}$$

步骤 4: 计算逆变换的雅可比矩阵

$$\mathbf{J}^{-1} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix}$$

计算偏导数:

$$\frac{\partial x}{\partial u} = \frac{3}{2u}, \quad \frac{\partial x}{\partial v} = -\frac{1}{2v} \quad (89)$$

$$\frac{\partial y}{\partial u} = -\frac{1}{2u}, \quad \frac{\partial y}{\partial v} = \frac{3}{2v} \quad (90)$$

因此:

$$\mathbf{J}^{-1} = \begin{pmatrix} \frac{3}{2u} & -\frac{1}{2v} \\ -\frac{1}{2u} & \frac{3}{2v} \end{pmatrix}$$

步骤 5: 计算逆变换的雅可比行列式

$$|\mathbf{J}^{-1}| = \frac{3}{2u} \cdot \frac{3}{2v} - \left(-\frac{1}{2v}\right) \cdot \left(-\frac{1}{2u}\right)$$

$$= \frac{9}{4uv} - \frac{1}{4uv} = \frac{8}{4uv} = \frac{2}{uv}$$

由于 $u = e^{(x+3y)/4}$, $v = e^{(3x+y)/4}$, 所以:

$$|\mathbf{J}^{-1}| = \frac{2}{e^{(x+3y)/4} \cdot e^{(3x+y)/4}} = \frac{2}{e^{(x+3y+3x+y)/4}} = \frac{2}{e^{x+y}} = 2e^{-(x+y)}$$

验证: $|\mathbf{J}| \cdot |\mathbf{J}^{-1}| = -\frac{1}{2}e^{x+y} \cdot 2e^{-(x+y)} = -1$ (符号差异是因为变换的方向性)

11 特征提取

11.1 特征提取概述 (Background)

11.1.1 什么是特征?

在图像处理和计算机视觉领域，**特征 (Feature)** 是指标志或描述图像内容中我们感兴趣的某一部分信息。它是一种对原始、高维、冗余的图像数据进行降维和信息浓缩的方式，将图像或其子区域转换为一组更有信息量、更易于处理的数值或符号。特征提取是连接图像分割与图像识别、分类、诊断等高级分析任务的关键桥梁。

整个过程可以分为两个阶段：

1. **特征检测 (Feature Detection)**: 在图像中定位我们感兴趣的结构或区域，例如通过分割得到对象的边界和区域。
2. **特征描述 (Feature Description)**: 对检测到的特征赋予定量的属性值。例如，对于一个分割出的肿瘤区域，我们可以计算它的面积、周长、平均灰度等。

一个好的特征应该具备一些理想的性质，例如对某些变换的**不变性 (Invariant)**（如平移、旋转）和对另一些变换的**协变性 (Covariant)**（如缩放）。

11.1.2 特征的分类

根据所利用的信息来源，图像特征可以大致分为三大类：

1. **形状特征 (Shape Feature)**: 只考虑感兴趣区域 (ROI) 的空间几何信息，如大小、轮廓、紧凑度等，与区域内的灰度强度分布无关。
2. **一阶特征 (First Order Feature)**: 只考虑 ROI 内像素（或体素）的强度分布信息，通常通过直方图进行计算，不考虑像素的空间位置关系。
3. **纹理特征 (Texture Feature / Second Order Feature)**: 同时考虑 ROI 内的强度信息和空间信息，描述像素（或体素）灰度值的空间排布规律，如平滑度、粗糙度、规律性等。

11.2 形状特征 (Shape Feature)

形状特征用于描述 ROI 的几何形态，可以进一步分为基于边界的描述符和基于区域的描述符。

11.2.1 基于边界的描述符 (Boundary Descriptors)

这些描述符利用从边缘检测或分水岭算法等方法获得的 ROI 边界信息。

基本边界描述符

- **长度 (Length)**: 边界的长度, 在数字图像中通常用边界上的像素总数来近似。
- **直径 (Diameter)**: 边界上任意两点之间的最大距离。 $diameter(B) = \max_{i,j}[D(p_i, p_j)]$, 其中 p_i, p_j 是边界上的点。
- **主轴与副轴 (Major and Minor Axis)**: 连接构成直径的两个极端点的线段是主轴。垂直于主轴的线段是副轴。
- **偏心率 (Eccentricity)**: 主轴与副轴长度的比值, 用于衡量边界的伸长程度。

基于椭圆的偏心率计算 (重要考点): 偏心率基于椭圆模型进行计算, 假设二维数据可以用椭圆区域近似:

- 椭圆方程: $\frac{c^2}{a^2} = \frac{a^2 - b^2}{a^2}$, 其中 a 为长轴半径, b 为短轴半径
- 偏心率公式: $e = \frac{c}{a} = \frac{\sqrt{a^2 - b^2}}{a} = \sqrt{1 - (b/a)^2}$, 其中 $(a \geq b)$
- 椭圆的主轴是特征向量, 对应协方差矩阵的特征值和特征向量
- 协方差矩阵: $C = \frac{1}{K-1} \sum_{k=1}^K (z_k - \bar{z})(z_k - \bar{z})^T$, 其中 $\bar{z} = \frac{1}{K} \sum_{k=1}^K z_k$
- K 为区域内点的总数

- **曲率 (Curvature)**: 边界斜率的变化率, 描述了边界的弯曲程度。
- **曲折度 (Tortuosity)**: 特别用于描述血管等线状结构的弯曲程度, 是实际路径长度与两端点直线距离的比值。

链码 (Chain Code) (重要考点) 链码 (也称为 Freeman 链码) 是一种紧凑而有效的边界表示方法, 通过记录沿边界移动的方向序列来编码二维边界。

基本原理:

- 从边界上的某个起始像素点开始, 按照某种顺序 (通常是顺时针或逆时针) 遍历整个边界
- 记录从当前像素到下一个边界像素的移动方向
- 用数字代码表示这些方向, 形成一个方向码序列

两种常用的链码方案:

1. 4-连通链码 (4-Connectivity Chain Code):

- 使用 4 个方向: 0 (右)、1 (下)、2 (左)、3 (上)
- 只考虑水平和垂直相邻的像素

- 适用于矩形网格上的简单边界

2. 8-连通链码 (8-Connectivity Chain Code):

- 使用 8 个方向: 0 (右)、1 (右下)、2 (下)、3 (左下)、4 (左)、5 (左上)、6 (上)、7 (右上)
- 考虑 8 邻域内的所有相邻像素
- 更精确地描述边界，特别是对角线方向的边界
- 最常用的方案

链码的特性:

- 旋转敏感性: 原始链码对旋转敏感, 旋转会导致所有方向码发生变化
- 起始点依赖性: 不同的起始点会产生不同的链码序列 (但形状相同)
- 存储效率高: 相比存储所有边界像素坐标, 链码需要更少的存储空间

链码的标准化: 为了使链码具有旋转和起始点不变性, 通常需要进行标准化:

- 差分链码 (Differential Chain Code): 计算相邻方向码之间的差值 (模 8 或模 4), 使其对旋转具有不变性
- 起始点标准化: 选择特定规则确定唯一的起始点, 如选择最左上角的点

链码应用:

- 形状匹配和识别
- 边界压缩和存储
- 图形重建和显示
- 形状分析和分类

边界特征签名 (Signatures) 特征签名是一种将二维边界转换为一维函数表示的方法, 便于分析。一个简单的方法是, 以区域的质心为原点, 绘制边界点到质心的距离 r 作为角度 θ 的函数 $r(\theta)$ 。这种一维曲线的形状 (如峰的数量和位置) 可以有效地描述和区分不同的边界形状。

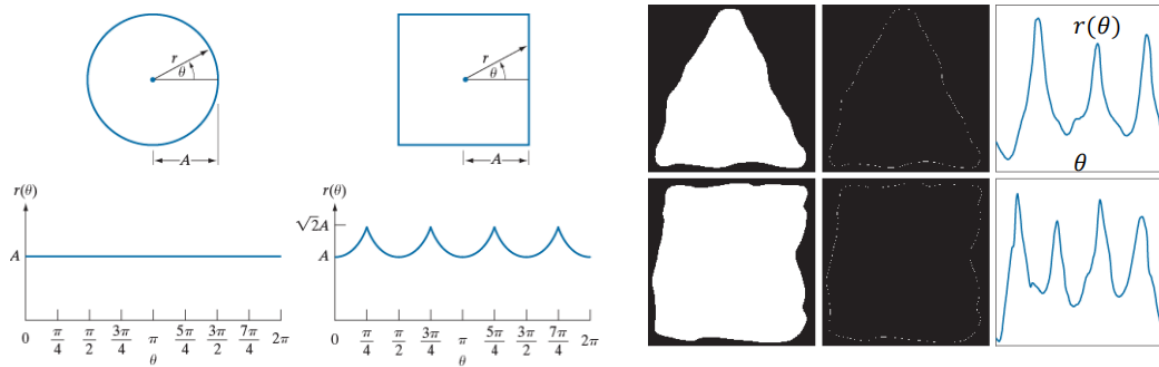


图 21: 边界特征签名示意图

傅里叶描述符 (Fourier Descriptors) (重要考点) 傅里叶描述符是一种功能强大且被广泛应用的边界表示方法。其基本思想是：

1. 将一个由 K 个点组成的数字边界看作复平面上的一个点序列。每个边界点的坐标 (x_k, y_k) 被表示为一个复数 $s(k) = x(k) + jy(k)$ 。
2. 对这个复数序列 $s(k)$ 进行离散傅里叶变换 (DFT)，得到一组傅里叶系数 $a(u)$ ：

$$a(u) = \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K}, \quad u = 0, 1, \dots, K-1$$

3. 这组傅里叶系数 $a(u)$ 就构成了该边界的傅里叶描述符。
4. 可以通过傅里叶逆变换 (IDFT) 从这些系数中完美地重建原始边界。更有用的是，我们可以只使用前 P 个 ($P < K$) 系数来进行逆变换，得到一个近似的边界 $\hat{s}(k)$ ：

$$\hat{s}(k) = \frac{1}{K} \sum_{u=0}^{P-1} a(u) e^{j2\pi uk/K}$$

傅里叶描述符的关键特性在于，低频系数决定了边界的总体形状，而高频系数则描述了边界的精细细节。因此，通过截断高频系数（即只使用前 P 个系数），我们可以得到一个平滑的、概括性的边界表示，这对于形状匹配和分类非常有用。使用的系数越少，重建的边界越粗糙；使用的系数越多，细节越丰富。

11.2.2 基于区域的描述符 (Region Descriptors)

这些描述符利用了 ROI 内部所有像素的信息。

基本区域描述符

- 面积 (Area, A): 区域内的像素总数。

- 周长 (Perimeter, p): 区域边界的长度。

重要的形状特征描述符 (重要考点):

- 紧凑度 (Compactness): 定义为周长的平方与面积的比值

$$\text{compactness} = \frac{p^2}{A}$$

对于一个圆盘, 紧凑度最小。该值对于平移和旋转是不变的。

- 圆度 (Circularity): 与紧凑度密切相关, 定义为

$$\text{circularity} = \frac{4\pi A}{p^2}$$

- 对于一个完美的圆形, 圆度为 1
- 重要特性: 独立于方向和平移变换 (independent of orientation and translation)
- 圆度是紧凑度的倒数乘以常数 4π

- 离心率 (Eccentricity): 通过椭圆近似 2D 数据来计算

$$\text{eccentricity} = \frac{\sqrt{\lambda_2^2 - \lambda_1^2}}{\lambda_2} = \sqrt{1 - \left(\frac{\lambda_1}{\lambda_2}\right)^2}, \quad \lambda_2 \geq \lambda_1$$

- 通过计算区域内所有像素坐标的协方差矩阵来获得特征值 λ_1 和 λ_2
- 特征值对应椭圆的主轴和副轴但是直接 c/a
- 该椭圆的偏心率可以作为区域的偏心率描述符
- 偏心率为 0 表示完美圆形, 接近 1 表示高度拉伸的椭圆

将多个描述符组合成一个**特征向量 (Feature Vector)**, 如 $x = [\text{compactness}, \text{circularity}, \text{eccentricity}]^T$, 可以在一个多维特征空间中表示一个对象, 从而可以根据点在空间中的距离来评估不同形状的相似性。

基本区域描述符实例分析 (重要考点): 下表展示了不同形状的基本区域描述符数值:





Descriptor				
Compactness	10.1701	42.2442	15.9836	13.2308
Circularity	1.2356	0.2975	0.7862	0.9478
Eccentricity	0.0411	0.0636	0	0.8117

图 22: 不同形状的基本区域描述符对比

分析结论:

- 圆形具有高紧凑度和圆度，偏心率很低，是最理想的圆形
- 星形紧凑度最低，圆度也最低，说明它与圆形差异最大
- 水滴形具有最大的偏心率，但很难仅通过紧凑度或圆度来区分
- 正方形具有零偏心率，表明其在主要方向上是对称的

拓扑描述符 (Topological Descriptors) 拓扑描述符描述的是区域在经过拉伸、弯曲等”橡皮膜”形变后仍保持不变的性质，它们不依赖于距离或角度的度量。

- 孔洞数量 (Number of holes): 区域中孔洞的个数是一个重要的拓扑特征。
- 欧拉数 (Euler Number, E): 欧拉数是一个经典的拓扑不变量，定义为区域中连通分量的数量 (C) 减去孔洞的数量 (H) 。

$$E = C - H$$

11.3 一阶特征 (First Order Features)

一阶特征，也称为统计特征或直方图特征，它们只描述 ROI 内像素（或体素）强度的分布情况，完全忽略了像素的空间排布信息。这些特征可以直接从图像的强度直方图计算得到。对于一个 ROI 区域 N ，其强度值为 $f(x, y)$ ，常用的一阶特征包括：

- 能量 (Energy): $Energy = \sum_{(x,y) \in N} f(x, y)^2$ 。
- 均值 (Mean): $\bar{f} = \frac{1}{|N|} \sum_{(x,y) \in N} f(x, y)$ 。
- 中位数 (Median): $Median = \text{median intensity in } N$ 。
- 最大/最小值与范围 (Max/Min, Range): $Max = \max_{(x,y) \in N} f(x, y)$, $Min = \min_{(x,y) \in N} f(x, y)$, $Range = Max - Min$ 。
- 方差与标准差 (Variance, Standard Deviation): $\sigma^2 = \frac{1}{|N|} \sum_{(x,y) \in N} (f(x, y) - \bar{f})^2$ 。方差或标准差反映了图像强度的离散程度或对比度。
- 高阶矩 (Higher Order Moments) (重要考点):

图像的 n 阶矩定义为：

$$\mu_n = \frac{1}{|N|} \sum_{(x,y) \in N} (f(x, y) - \bar{f})^n$$

其中 \bar{f} 是均值。各阶矩的物理含义：

- 一阶矩 (μ_1): 均值 (Mean)，描述图像的平均亮度水平

$$\mu_1 = \bar{f} = \frac{1}{|N|} \sum_{(x,y) \in N} f(x, y)$$

- 二阶矩 (μ_2): 对比度 (Variance), 描述图像灰度值的分散程度

$$\mu_2 = \sigma^2 = \frac{1}{|N|} \sum_{(x,y) \in N} (f(x,y) - \bar{f})^2$$

- 三阶矩 (μ_3): 不对称性 (Skewness), 衡量直方图分布的偏斜程度

$$\text{Skewness} = \frac{\mu_3}{\sigma^3} = \frac{\frac{1}{|N|} \sum_{(x,y) \in N} (f(x,y) - \bar{f})^3}{\sigma^3}$$

- * 正偏斜 (Positively Skewed): 分布尾部偏向右侧 (高灰度值)
- * 负偏斜 (Negatively Skewed): 分布尾部偏向左侧 (低灰度值)
- * 零偏斜: 分布对称

- 四阶矩 (μ_4): 尖峭度 (Kurtosis), 衡量分布”尾部厚度”的程度

$$\text{Kurtosis} = \frac{\mu_4}{\sigma^4} = \frac{\frac{1}{|N|} \sum_{(x,y) \in N} (f(x,y) - \bar{f})^4}{\sigma^4}$$

- * 正峰度 (Positive): 分布比正态分布更尖峭, 尾部更厚
- * 负峰度 (Negative): 分布比正态分布更平坦, 尾部更薄
- * 正态分布的峰度为 3, 常用超峰度 (Excess Kurtosis = Kurtosis - 3)

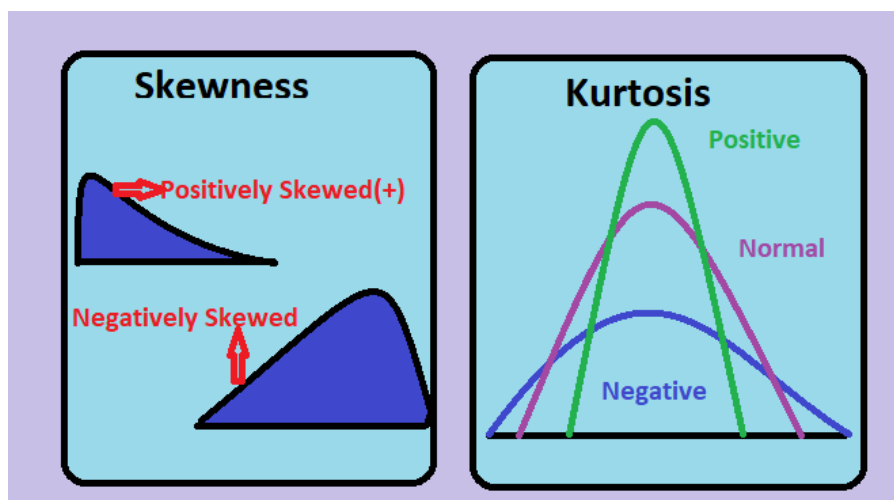


图 23: 偏度和峰度示意图

• 基于直方图的特征:

- 均匀度 (Uniformity): $U = \sum_{i=0}^{L-1} p(z_i)^2$, 其中 $p(z_i)$ 是归一化直方图。如果图像强度完全恒定 (只有一个灰度值), 均匀度为 1 (最大); 如果图像强度完全均匀分布在 L 个灰度级上, 均匀度为 $1/L$ (最小)。强度分布越集中, 均匀度越高。
- 相对平滑度 (Relative Smoothness): $R_z = 1 - \frac{1}{1 + \sigma^2(z)}$, 其中 $\sigma^2(z)$ 是方差。当方差为 0 时 $R_z = 0$ (完全平滑), 方差越大 R_z 越接近 1。

- **熵 (Entropy):** $e = - \sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$ 。熵衡量了图像强度的随机性或复杂性。如果图像强度恒定，熵为 0（最小）。

不同纹理类型的一阶特征对比 (重要考点):

纹理	均值	标准差	R (归一化)	三阶矩	均匀度	熵
平滑 (Smooth)	82.64	11.79	0.002	-0.105	0.026	5.434
粗糙 (Coarse)	143.56	74.63	0.079	-0.151	0.005	7.783
规律 (Regular)	99.72	33.73	0.017	0.750	0.013	6.674

表 12: 不同纹理类型的一阶特征数值对比

特征分析:

- **平滑纹理:** 标准差最小，相对平滑度最低，均匀度最高，熵最小
- **粗糙纹理:** 标准差最大，相对平滑度最高，均匀度最低，熵最大
- **规律纹理:** 三阶矩为正值 (0.750)，表明具有特定的分布规律性

例题 1: 一阶特征计算 (重要考点)

给定图像矩阵:

$$\begin{bmatrix} 3 & 1 & 0 & 0 & 2 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 3 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 3 & 1 \end{bmatrix}$$

题目要求:

1. 计算图像的能量 (Energy)、均值 (Mean)、范围 (Range)
2. 计算二阶矩、三阶矩 (2nd, 3rd moment of mean)
3. 绘制图像直方图，计算均匀度 (Uniformity) 和熵 (Entropy)

解答:

步骤 1: 基本统计量计算

- 像素总数: $N = 25$
- 像素值: 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3
- 均值: $\bar{f} = \frac{1}{25}(6 \times 0 + 6 \times 1 + 5 \times 2 + 4 \times 3) = \frac{28}{25} = 1.12$
- 最大值: $Max = 3$, 最小值: $Min = 0$

- 范围: $Range = Max - Min = 3$
- 能量: $Energy = \sum f(x, y)^2 = 6 \times 0^2 + 6 \times 1^2 + 5 \times 2^2 + 4 \times 3^2 = 58$

步骤 2: 高阶矩计算

- 二阶矩 (方差): $\mu_2 = \frac{1}{25} \sum (f(x, y) - 1.12)^2 = 1.4256$
- 三阶矩: $\mu_3 = \frac{1}{25} \sum (f(x, y) - 1.12)^3 = 0.512$

步骤 3: 直方图特征

- 直方图: $p(0) = \frac{6}{25} = 0.24$, $p(1) = \frac{6}{25} = 0.24$, $p(2) = \frac{5}{25} = 0.2$, $p(3) = \frac{4}{25} = 0.16$
- 均匀度: $U = \sum p(z_i)^2 = 0.24^2 + 0.24^2 + 0.2^2 + 0.16^2 = 0.1712$
- 熵: $e = -\sum p(z_i) \log_2 p(z_i) = -(0.24 \log_2(0.24) \times 2 + 0.2 \log_2(0.2) + 0.16 \log_2(0.16)) = 1.946 \text{ bits}$

11.4 纹理特征 (Texture Features)

纹理是图像中一种重要的视觉特征，直观上可以理解为图像的平滑度、粗糙度、颗粒感或规律性。与一阶特征不同，纹理特征描述的是像素灰度值的空间分布关系，属于二阶或更高阶的统计特征。

11.4.1 灰度共生矩阵 (GLCM) (重要考点)

灰度共生矩阵 (Gray-Level Co-occurrence Matrix, GLCM) 是描述纹理最经典和最强大的方法之一。它统计了图像中具有特定灰度值的像素对，在预设的空间关系（距离和方向）下共同出现的频率。

GLCM 的构建步骤:

1. 灰度量化的: 为了减小计算量，首先将原始图像的灰度级量化到较少的级别 K 。
2. 定义空间关系: 定义一个位移向量（或位置算子 Q ），例如”右侧相邻 1 个像素”($\Delta x = 1, \Delta y = 0$)，或”右下方相邻 1 个像素”($\Delta x = 1, \Delta y = 1$)。
3. 构建共生矩阵: 创建一个 $K \times K$ 大小的矩阵 G 。遍历图像中的每一个像素，查找其在定义好的空间关系下的邻居像素。如果当前像素的灰度级为 i ，其邻居像素的灰度级为 j ，则将矩阵 G 的第 (i, j) 个元素的值加 1。
4. 归一化: 将矩阵 G 中的每个元素除以像素对的总数 n ，得到归一化的共生矩阵 P ，其元素 p_{ij} 表示灰度对 (i, j) 出现的概率。

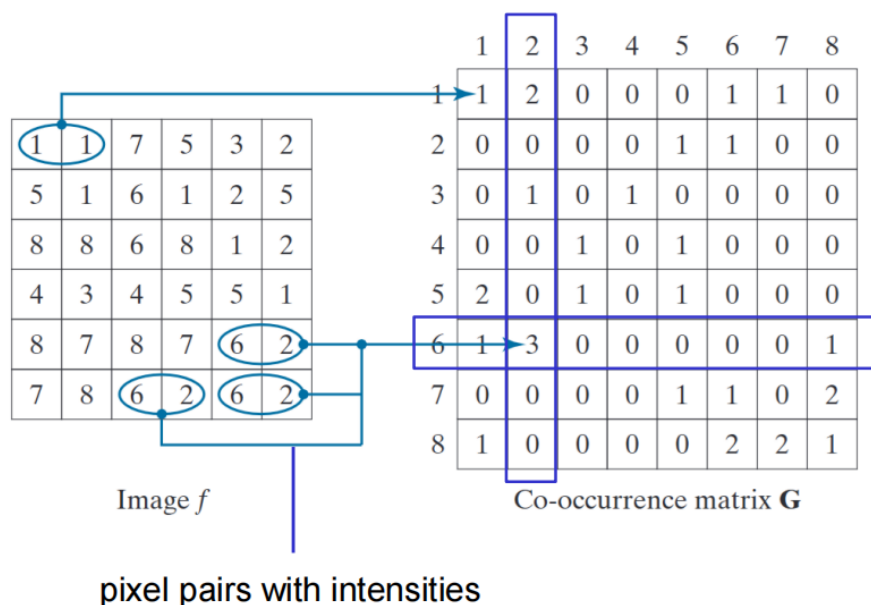


图 24: GLCM 示意图

基于 GLCM 的纹理描述符: 从归一化的 GLCM 矩阵 P 中, 可以计算出多种描述纹理的量化指标:

- **能量/均匀度 (Energy/Uniformity):** $\sum \sum p_{ij}^2$ 。如果图像纹理均匀 (如大片平坦区域), GLCM 中的值会集中在少数几个位置, 能量值就高。
- **对比度 (Contrast):** $\sum \sum (i - j)^2 p_{ij}$ 。衡量了图像的局部强度变化, 反映了纹理的清晰度和沟纹的深浅。对比度越大, 纹理越清晰。
- **相关性 (Correlation):** $\sum \sum \frac{(i - m_r)(j - m_c)p_{ij}}{\sigma_r \sigma_c}$ 。衡量了图像灰度在空间上的线性依赖关系。如果图像具有明显的方向性纹理, 相关性会很高。
- **同质性 (Homogeneity):** $\sum \sum \frac{p_{ij}}{1 + |i - j|}$ 。衡量 GLCM 中元素在对角线附近的聚集程度。图像局部变化越小, 同质性越高。
- **熵 (Entropy):** $-\sum \sum p_{ij} \log_2 p_{ij}$ 。衡量图像纹理的随机性或复杂性。纹理越复杂、越无序, 熵值越高。

例题 2: 灰度共生矩阵计算 (重要考点)

给定图像矩阵模式:

$$\begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$

题目要求:

1. 位置算子 Q 定义为“右侧相邻 1 个像素” (one pixel to the right)
2. 位置算子 Q 定义为“右侧相邻 2 个像素” (two pixels to the right)
3. 对于情况 1 和 2, 分别计算 GLCM 的对比度 (Contrast) 和同质性 (Homogeneity)

分析结论:

- 距离越远的位置算子会产生更高的对比度值
- 规律性图像模式在不同距离下的同质性相对稳定
- GLCM 特征能有效捕获纹理的空间关系特征

11.4.2 谱方法 (Spectral Approaches)

另一种分析纹理的方法是利用傅里叶变换等谱分析工具。其基本思想是:

- 周期性或方向性强的纹理, 会在其傅里叶频谱图上表现为集中的、明亮的峰值。
- 频谱中峰值的位置可以揭示纹理的**基本空间周期**。
- 频谱中峰值的方向可以揭示纹理的**主方向**。
- 通过在频域进行滤波, 可以分离出周期性成分和非周期性成分, 然后对非周期性部分再用统计方法进行分析。

11.5 影像组学 (Radiomics)

11.5.1 定义与工作流程

影像组学 (Radiomics) 是一个新兴的交叉领域, 其核心思想是: 通过高通量的方式, 从医学图像 (如 CT、MRI、PET) 的感兴趣区域 (ROI) 中自动提取大量的、定量的图像特征, 并将这些特征转化为可供挖掘的高维数据, 以期发现与临床终点 (如诊断、预后、疗效预测) 相关的影像生物标志物。

一个典型的影像组学工作流程包括:

1. **图像采集与分割:** 获取标准化的医学影像, 并对病灶等 ROI 进行精确分割。
2. **特征提取:** 从 ROI 中大规模提取多类别特征, 通常包括我们前面讨论的所有特征类别:
 - 形状特征。
 - 一阶统计特征 (直方图特征)。
 - 纹理特征 (如 GLCM、灰度游程长度矩阵等)。

- **滤波特征:** 这是影像组学的一大特色。在提取上述特征之前, 先对原始图像应用一系列滤波器 (如小波变换、高斯-拉普拉斯滤波等), 然后在滤波后的图像上再提取特征, 从而极大地扩展了特征空间, 能够捕获不同尺度和频率下的图像信息。

3. 模型构建与验证:

- **特征选择:** 由于提取的特征数量庞大 (成百上千), 存在高度冗余和过拟合风险, 必须进行特征选择, 筛选出最具有预测能力的特征子集。
- **模型训练:** 使用筛选出的特征作为输入, 训练机器学习模型 (如支持向量机 SVM、随机森林、LASSO 回归等) 来执行分类、回归或生存分析任务。
- **模型验证:** 在独立的验证集或测试集上评估模型的性能和泛化能力。

11.5.2 应用与挑战

应用: 影像组学已在肿瘤学等多个领域展现出巨大潜力, 例如:

- **诊断与分级:** 预测肿瘤的良恶性、病理分型等。
- **预后预测:** 构建影像组学标签 (rad-signature) 来预测患者的生存期 (如无进展生存期 PFS)。
- **疗效预测:** 预测患者对特定治疗 (如放疗、化疗、靶向治疗) 的反应。
- **关联基因组学:** 探索影像组学特征与基因表达、突变状态等分子生物学信息之间的关联。

挑战与局限性:

- **可重复性与标准化:** 影像组学特征的计算对图像采集参数、分割方法、软件实现等多个环节都非常敏感, 导致不同研究之间的结果可重复性差。图像生物标志物标准化倡议 (IBSI) 等工作正在致力于解决此问题。
- **过拟合风险:** 在特征数量远大于样本数量的情况下, 极易产生过拟合, 即模型在训练集上表现很好, 但在新数据上表现很差。
- **可解释性:** 许多复杂的影像组学特征缺乏直观的生物学解释, 这限制了其在临床上的接受度。

11.5.3 图像增强技术

图像增强是医学图像处理的基础步骤, 旨在改善图像质量, 突出感兴趣的特征, 为后续分析提供更好的输入。

空域增强方法 空域增强直接在图像的像素域进行操作, 是最直观的增强方式。

点运算增强

- **灰度变换:** 通过改变像素值的映射关系来调整图像的对比度和亮度
 - 线性变换: $g(x, y) = af(x, y) + b$, 其中 a 控制对比度, b 控制亮度
 - 对数变换: $g(x, y) = c \log(1 + f(x, y))$, 用于压缩动态范围
 - 幂律变换: $g(x, y) = cf(x, y)^\gamma$, 伽马校正的基础
- **直方图处理:**
 - 直方图均衡化: 通过累积分布函数变换实现灰度级的均匀分布
 - 直方图规定化: 将图像的直方图变换为指定的形状
 - 局部直方图均衡化: 在图像的局部区域内进行直方图均衡化

邻域运算增强

- **空域滤波:**
 - 平滑滤波: 高斯滤波、均值滤波、中值滤波等, 用于去噪
 - 锐化滤波: 拉普拉斯算子、Sobel 算子等, 用于边缘增强
 - 形态学滤波: 开运算、闭运算、顶帽变换等
- **自适应滤波:**
 - 维纳滤波: 基于信号和噪声的统计特性进行滤波
 - 双边滤波: 在保持边缘的同时进行平滑
 - 非局部均值滤波: 利用图像的自相似性进行去噪

频域增强方法 频域增强通过傅里叶变换将图像转换到频域, 在频域进行处理后再变换回空域。

频域滤波基础

- **低通滤波:** 去除高频噪声, 保留图像的主要结构
 - 理想低通滤波器: 具有明确的截止频率
 - 巴特沃斯低通滤波器: 平滑的频率响应
 - 高斯低通滤波器: 最优的空频域特性
- **高通滤波:** 增强边缘和细节信息

- 理想高通滤波器：完全阻止低频成分
- 巴特沃斯高通滤波器：渐变的频率响应
- 高斯高通滤波器：平滑的高频增强

- **带通和带阻滤波**：选择性地处理特定频率范围

同态滤波 同态滤波是一种特殊的频域增强技术，特别适用于处理照明不均匀的图像：

- 基于图像形成模型： $f(x, y) = i(x, y) \cdot r(x, y)$
- 通过对数变换将乘性关系转换为加性关系
- 在频域分别处理照明分量和反射分量
- 能够同时实现动态范围压缩和对比度增强

例题 3：2023-2024 春夏数字图像处理回忆卷 GLCM 题目 (重要考点)

给定图像矩阵：

$$f = \begin{bmatrix} 7 & 5 & 3 & 2 \\ 5 & 1 & 6 & 1 \\ 8 & 8 & 6 & 8 \\ 4 & 3 & 4 & 5 \\ 8 & 7 & 8 & 7 \\ 7 & 8 & 6 & 2 \end{bmatrix}$$

位置算子 Q 定义为”右侧相邻一个像素”（考虑周有 L 个灰度）

题目要求：

1. 写出一个矩阵的灰度共生矩阵
2. 熵的公式，计算灰度共生矩阵的熵
3. 灰度共生矩阵的意义

解答：

步骤 1：确定灰度级别首先确定图像中的灰度级别：1, 2, 3, 4, 5, 6, 7, 8（共 8 个灰度级）
所以 $L = 8$ ，需要构建 8×8 的共生矩阵 G 。

步骤 2：统计像素对使用位置算子”右侧相邻一个像素”，逐行扫描统计灰度对 (i, j) 的出现次数：

第 1 行：(7,5), (5,3), (3,2) 第 2 行：(5,1), (1,6), (6,1) 第 3 行：(8,8), (8,6), (6,8) 第 4 行：
(4,3), (3,4), (4,5) 第 5 行：(8,7), (7,8), (8,7) 第 6 行：(7,8), (8,6), (6,2)

步骤 3：构建共生矩阵 G 统计各灰度对出现的次数：

$$G = \begin{bmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 5 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 6 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 8 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 1 \end{bmatrix}$$

总像素对数: $n = 18$

步骤 4: 归一化共生矩阵 P 将每个元素除以总像素对数 $n = 18$:

$$P = \begin{bmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & \frac{1}{18} & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & \frac{1}{18} & 0 & \frac{1}{18} & 0 & 0 & 0 \\ 5 & \frac{1}{18} & 0 & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 \\ 6 & \frac{1}{18} & \frac{1}{18} & 0 & 0 & 0 & 0 & 0 & \frac{1}{18} \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{18} \\ 8 & \frac{1}{18} & 0 & 0 & 0 & 0 & \frac{2}{18} & \frac{2}{18} & \frac{1}{18} \end{bmatrix}$$

步骤 5: 熵的公式与计算

熵的公式:

$$H = - \sum_{i=1}^L \sum_{j=1}^L p_{ij} \log_2 p_{ij}$$

其中 p_{ij} 为归一化共生矩阵中的元素, 当 $p_{ij} = 0$ 时, $p_{ij} \log_2 p_{ij} = 0$ 。

熵的计算: 只考虑非零元素:

$$H = - \left[8 \times \frac{1}{18} \log_2 \left(\frac{1}{18} \right) + 2 \times \frac{2}{18} \log_2 \left(\frac{2}{18} \right) \right] \quad (91)$$

$$= - \left[\frac{8}{18} \log_2 \left(\frac{1}{18} \right) + \frac{4}{18} \log_2 \left(\frac{2}{18} \right) \right] \quad (92)$$

$$= - \left[\frac{8}{18} \times (-4.17) + \frac{4}{18} \times (-3.17) \right] \quad (93)$$

$$= -(-1.85 - 0.70) = 2.55 \text{ bits} \quad (94)$$

步骤 6: 灰度共生矩阵的意义

灰度共生矩阵的意义:

- **纹理特征描述:** GLCM 能够有效描述图像的纹理特征, 通过统计不同灰度值像素对在特定空间关系下的共现频率
- **空间关系量化:** 不仅考虑像素的灰度值分布, 还考虑了像素间的空间位置关系
- **方向敏感性:** 通过改变位置算子, 可以分析不同方向上的纹理特征
- **纹理均匀性判断:**
 - 如果图像纹理均匀, GLCM 中的值会集中在对角线附近
 - 如果图像纹理粗糙或不规则, GLCM 中的值会分散分布
- **纹理复杂度度量:** 通过熵值可以量化纹理的复杂程度, 熵值越高, 纹理越复杂随机

12 机器学习与深度学习

12.1 引言：医学中的人工智能 (Introduction: AI in Medicine)

12.1.1 发展历程 (Historical Development)

人工智能在医疗领域的应用可以大致分为四个阶段：

- **1980-2000 年：专家系统时代。**这个时期的代表是基于规则的系统，它们试图模拟人类专家的决策过程。例如，美国的 MYCIN 系统用于血液感染疾病的诊断和治疗。在中国，1978 年的“关幼波肝病诊疗程序”是将专家系统应用于中医领域的早期尝试。
- **2000-2015 年：机器学习时代。**随着计算能力和数据量的增加，机器学习技术开始被广泛探索，医疗领域进入了数据驱动方法的时代。
- **2015-2017 年：深度学习时代。**深度学习的突破，特别是卷积神经网络（CNN）的成功，极大地推动了医疗 AI 在多个细分领域的商业化落地，尤其是在医学影像分析方面。
- **2018 年至今：大模型时代。**以 Transformer 架构为基础的大语言模型（LLMs）和多模态模型开始涌现，如 GPT 系列、BERT 等。这些模型在海量数据上进行预训练，展现出强大的通用能力，引发了医学人工智能的新一轮变革。

12.1.2 大模型时代 (The Era of Large Models)

大模型（Large Models）是指在海量数据上进行大规模预训练，然后通过指令微调等方式适应一系列下游任务的人工智能模型。这被认为是迈向通用人工智能（AGI）的里程碑式进展。

- **通用大模型 (General Large Models)：**跨领域、跨任务的深度学习模型，具有广泛的适用性。
- **领域大模型 (Domain-specific Large Models)：**针对特定领域（如医疗）的模型，通常由“通用大模型 + 领域知识 + 领域数据”构成，具有更强的专业能力。

12.1.3 典型的医疗大模型 (Typical Medical Large Models)

- **LLaVA-Med (微软)：**这是一个开创性的生物医学多模态对话模型。
 - **架构：**由视觉编码器（Vision Encoder）、投影层（Projection）和语言模型（Language Model）组成，能够同时理解图像和文本指令。
 - **数据：**使用了从 PubMed Central 获取的包含 1500 万生物医学图像-文本对的 PMC-15M 数据集进行训练。

- **训练**：分阶段进行，首先对齐图像和文本的表示，然后学习开放式的对话能力，最后在下游任务上微调。
- **能力**：能就医学图像内容进行问答。例如，对于一张胸部 CT 图像，它可以定位病灶（“右肺上叶后部”），判断其是否侵犯胸壁，并解释“空洞性病变”（cavitating lesion）的含义和可能原因。
- **Med-PaLM M (谷歌)**：首个全科通才（generalist）医疗大模型。
 - **多模态与多任务**：该模型能够处理包括医学成像、临床文本和基因组学在内的多种数据模态，并执行 14 种不同的任务，如图像分类、视觉问答、放射学报告生成和总结等。
 - **性能**：Med-PaLM M 使用一组单一的模型权重，无需针对特定任务进行定制，就能在多个任务上达到甚至超过当时最先进的（SOTA）专业模型的性能。在一项盲选中，40.5% 的病例中，临床医生更偏好模型生成的报告而非放射科医生的报告。
 - **基准**：谷歌为此构建了 MultiMedBench，一个包含超过 100 万样本的多模态生物医学基准测试集。
- **Med-Gemini (谷歌)**：基于更强大的 Gemini 模型构建，是谷歌在医疗 AI 领域的进一步探索。
 - **核心能力**：继承并增强了 Gemini 的高级推理、多模态理解和长文本处理能力。
 - **训练方法**：引入了高质量的真实世界医疗数据，并通过“人在回路的思维链（chain-of-reasoning）”方法进行增强训练。
 - **特色功能**：具备“不确定性引导的搜索”能力。当模型对答案不自信时，可以主动生成搜索查询来查找额外信息以解决冲突和不确定性。

12.2 机器学习基础 (Machine Learning Fundamentals)

12.2.1 传统方法：特征工程 (Traditional Method: Feature Engineering)

在深度学习普及之前，机器学习在图像处理中的应用严重依赖于特征工程。

- **定义**：特征工程是指由领域专家手动设计和提取能够描述图像关键信息的特征的过程。对于医学图像，一个核心问题是“什么是高质量的图像？”，答案是“可诊断的（diagnoseable）”。
- **特征示例**：为了量化图像质量，专家可能会定义诸如“边缘清晰度（Clear edge）”、“高对比度（High contrast）”、“高信噪比（High SNR）”等特征。反之，运动伪影（Motion artifact）则代表低质量。
- **局限性 (Problems)**：

- **受限于人类知识**: 设计的特征完全依赖于专家的先验知识, 可能无法捕捉到所有相关信息。
- **无法获得高维特征**: 手动设计的特征通常是低维的, 而图像中蕴含的复杂模式往往需要高维特征来描述。
- **主观偏见**: 特征的设计和选择可能带有主观性。
- **泛化能力差**: 对于新数据或小样本情况, 效果可能不佳。

12.2.2 从特征工程到机器学习

深度学习的核心优势在于实现了”端到端”的学习, 即从原始数据(如图像像素)直接学习到最终的输出(如诊断结果), 而中间的特征提取过程是自动完成的, 这些特征被称为”可学习的特征(Learned feature)”。这克服了传统特征工程的种种弊端。

12.2.3 线性不可分问题与核技巧思想

许多现实世界的问题, 包括图像分类, 都不是线性可分的。

- **线性不可分**: 在原始特征空间中, 无法用一个线性分类器(如一条直线或一个平面)将不同类别的样本完全分开。
- **解决方案: 升维**: 一个关键思想是将数据映射到一个更高维度的特征空间, 使其在这个高维空间中变得线性可分。例如, 通过引入一个新的维度”到原点的距离(The Distance)”, 可以将二维空间中的同心圆分布转换为三维空间中可用一个平面分开的点集。神经网络的隐藏层本质上就是在进行类似的非线性变换和升维操作。

12.3 神经网络基础 (Fundamentals of Neural Networks)

12.3.1 感知机 (The Perceptron)

- **历史**: 感知机是第一个学习机器, 于 1960 年在康奈尔大学被发明。它是一个位于特征提取器之上的线性分类器。
- **模型结构**: 输入特征向量 X , 经过一个特征提取器得到 $F_i(X)$, 然后通过加权求和与偏置项 b 结合, 最后由一个符号函数 (sign function) 输出类别。其决策函数可以写为:

$$d(x) = \sum_{i=1}^n w_i x_i + w_{n+1}$$

其中 x_i 是输入, w_i 是权重, w_{n+1} 是偏置项。输出 O 由一个激活元件 (activation element) 决定:

$$O = \begin{cases} +1 & \text{if } \sum_{i=1}^n w_i x_i > -w_{n+1} \\ -1 & \text{if } \sum_{i=1}^n w_i x_i < -w_{n+1} \end{cases}$$

- **训练算法：**这是一个迭代的过程。当一个属于类别 C_1 的样本 y 被错误地分类时（即 $w^T y \leq 0$ ），权重会按照 $w(k+1) = w(k) + \alpha y$ 进行更新，其中 α 是学习率。反之亦然。当所有训练样本都能被正确分类时，算法收敛。
- **局限性：**单层感知机只能解决线性可分的问题，对于像 XOR 这样的非线性问题则无能为力。

12.3.2 激活函数 (Activation Functions)

激活函数为神经网络引入了非线性，这是网络能够学习复杂模式的关键。激活函数必须是可微的，以便于使用基于梯度的优化方法（如反向传播）。

- **Sigmoid:** $\sigma(x) = \frac{1}{1+e^{-x}}$ 。将输入压缩到 $(0, 1)$ 之间，常用于二分类问题的输出层。
- **Tanh (双曲正切):** $\tanh(x)$ 。将输入压缩到 $(-1, 1)$ 之间，通常比 Sigmoid 收敛更快。
- **ReLU (Rectified Linear Unit):** $f(x) = \max(0, x)$ 。计算简单，能有效缓解梯度消失问题，是目前最常用的激活函数之一。
- **Leaky ReLU:** $f(x) = \max(0.1x, x)$ 。ReLU 的变体，解决了 ReLU 在负区间的“死亡”问题。
- **ELU (Exponential Linear Unit):** 是 ReLU 的另一变体，旨在结合 Sigmoid 和 ReLU 的优点。

12.3.3 XOR 问题：神经网络非线性能力的核心示例

XOR（异或）问题是证明单层感知机局限性和多层网络必要性的经典例子。

- **问题：**XOR 函数在输入不同时输出 1，输入相同时输出 0。这些数据点在二维平面上是线性不可分的。
- **线性模型失败：**若尝试用线性模型 $f(x; w, b) = x^T w + b$ 拟合 XOR 数据，通过最小化 L2 损失函数求解，会得到 $w_1 = 0, w_2 = 0, b = 1/2$ 的结果，导致模型对所有输入都输出 0.5，完全无法工作。
- **解决方案：**引入隐藏层和非线性。通过构建一个包含隐藏层和 ReLU 激活函数的两层神经网络，可以完美解决 XOR 问题。模型结构为 $y = m^T \cdot \text{ReLU}(W^T x + b) + c$ 。一个给定的解决方案是：

$$W = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, b = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, m = \begin{pmatrix} 1 \\ -2 \end{pmatrix}, c = 0$$

这个例子具体展示了网络如何通过组合简单的非线性单元来构建复杂的决策边界。

12.3.4 神经网络的训练：反向传播 (Training NNs: Backpropagation)

反向传播 (BP) 算法是训练神经网络的核心。它利用链式法则来高效地计算损失函数相对于网络中所有参数 (权重和偏置) 的梯度。

- **代价函数 (Cost Function):** 通常是均方误差 (Mean Square Error) 项和正则化 (Regularization) 项的和。

$$J(W, b) = \left[\frac{1}{m} \sum_{i=1}^m \frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2$$

正则化项 (第二项) 通过惩罚大的权重值来防止过拟合。

- **BP 算法核心步骤:**

1. **前向传播:** 输入样本, 逐层计算直到输出层, 得到预测值 $a^{(n_l)}$ 。
2. **计算输出层误差 $\delta^{(n_l)}$:** 这是代价函数对该层加权输入 $z^{(n_l)}$ 的偏导数。

$$\delta_i^{(n_l)} = -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)})$$

其中 f' 是激活函数的导数。

3. **反向传播误差:** 从后向前逐层计算每一层的误差 $\delta^{(l)}$ 。第 l 层的误差是第 $l+1$ 层误差的加权和, 再乘以当前层激活函数的导数。

$$\delta^{(l)} = ((W^{(l)})^T \delta^{(l+1)}) \odot f'(z^{(l)})$$

其中 \odot 表示元素级乘积。

4. **计算梯度:** 根据计算出的误差项 δ , 可以得到代价函数对每一层权重 $W^{(l)}$ 和偏置 $b^{(l)}$ 的梯度。

$$\frac{\partial J}{\partial W_{ij}^{(l)}} = a_j^{(l)} \delta_i^{(l+1)} \quad , \quad \frac{\partial J}{\partial b_i^{(l)}} = \delta_i^{(l+1)}$$

5. **参数更新:** 使用梯度下降法更新参数。

$$W_{ij}^{(l)} := W_{ij}^{(l)} - \alpha \frac{\partial J}{\partial W_{ij}^{(l)}}$$

- **梯度下降法变体:**

- **批量梯度下降 (Batch GD):** 每次更新使用全部训练数据, 计算成本高。
- **随机梯度下降 (SGD):** 每次更新只用一个样本, 速度快但波动大。
- **小批量梯度下降 (Mini-batch GD):** 介于两者之间, 是深度学习中最常用的方法。
- **动量 (Momentum):** 引入一个速度向量 v_t , 可以加速收敛并减少震荡。 $v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta)$ 。

12.4 卷积神经网络 (Convolutional Neural Networks - CNN)

CNN 是专门为处理网格状数据（如图像）而设计的神经网络。

12.4.1 核心思想与动机

全连接网络处理高维图像（如 200×200 ）时参数量巨大，计算上不可行。CNN 利用了图像的三个重要属性：

1. **局部性 (Locality)**: 一些模式（如边缘、角点）远小于整个图像。这启发了使用小的卷积核进行局部连接。
2. **平移不变性 (Translation Invariance)**: 一个模式（如一只眼睛）可能出现在图像的任何位置。这启发了“权值共享”，即用同一个卷积核扫描整个图像。
3. **尺度不变性 (Scale Invariance)**: 对像素进行下采样不会改变图像中的物体。这启发了“池化”操作。

12.4.2 关键组件

- **卷积层 (Convolutional Layer)**

- **操作**: 一个称为“滤波器”或“卷积核”的小矩阵在输入图像上滑动，计算点积，生成一个“特征图 (Feature Map)”。
- **权值共享 (Shared Weights)**: 卷积核内的参数在整个卷积过程中保持不变，极大地减少了模型的参数数量。
- **多滤波器**: 一个卷积层通常包含多个滤波器，每个滤波器学习提取一种特定的特征（如水平边缘、垂直边缘、某种颜色等）。
- **感受野 (Receptive Field)**: 输出特征图上的一个像素对应于输入图像中的一个区域，这个区域就是该像素的感受野。随着网络层数的加深，感受野会逐渐变大。
- **填充 (Padding)**: 在输入图像周围添加像素（通常是 0），以控制输出特征图的空间尺寸，避免图像尺寸过快缩小。

- **池化层 (Pooling Layer)**

- **操作**: 对特征图进行下采样 (down sampling)，以减小其尺寸。
- **类型**: 最常用的是最大池化 (Max Pooling)，即在邻域内取最大值。其他还有平均池化 (Average Pooling) 等。
- **优点**: 提供特征不变性（对小的平移不敏感）、降低特征维度、减少计算量、在一定程度上防止过拟合。

- **全连接层 (Fully Connected Layer)**: 在经过多层卷积和池化后，将最终的特征图“展平 (Flatten)” 成一个一维向量，然后送入一个或多个全连接层进行分类或回归。

12.4.3 CNN 基本运算（手动计算）

重要考点：这是检验对 CNN 工作原理理解的核心。

基本计算步骤

- **卷积：**将卷积核与输入矩阵的对应区域进行元素级乘法后求和。注意步长（Stride）和填充（Padding）的影响。
- **ReLU 激活：**对卷积结果的每个元素应用 $f(x) = \max(0, x)$ 。
- **最大池化：**在池化窗口内取最大值作为输出。注意窗口大小和步长。

考试例题：CNN 前向传播计算 题目：给定以下输入矩阵 $I (4 \times 4)$ 和卷积核 $K (3 \times 3)$ ，计算卷积层、ReLU 激活层和最大池化层的输出结果。显示每一步的主要计算过程。

$$\begin{array}{cc} \text{输入矩阵 } I (4 \times 4) & \text{卷积核 } K (3 \times 3) \\ \begin{pmatrix} 3 & 1 & 2 & 0 \\ 1 & 2 & 1 & 3 \\ 0 & 1 & 3 & 2 \\ 2 & 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \end{array}$$

网络参数：

- 卷积层参数：步长 (Stride) = 1，填充 (Padding) = 0
- 激活函数：ReLU，即 $f(x) = \max(0, x)$
- 最大池化层参数：窗口大小 = 2×2 ，步长 = 1

要求计算：

1. 卷积输出 C
2. ReLU 激活输出 R
3. 最大池化输出 P

解答：

步骤 1：卷积操作计算

输出尺寸计算： $(4 - 3 + 2 \times 0) / 1 + 1 = 2$ ，所以卷积输出 C 为 2×2 矩阵。

各位置的卷积计算：

位置 (0,0):

$$C_{00} = \begin{pmatrix} 3 & 1 & 2 \\ 1 & 2 & 1 \\ 0 & 1 & 3 \end{pmatrix} \odot \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} = (3 \times 1 + 1 \times 0 + 2 \times (-1)) + (1 \times 1 + 2 \times 0 + 1 \times (-1)) + (0 \times 1 + 1 \times 0 + 3 \times (-1)) =$$

位置 (0,1):

$$C_{01} = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 3 \\ 1 & 3 & 2 \end{pmatrix} \odot \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} = (1 \times 1 + 2 \times 0 + 0 \times (-1)) + (2 \times 1 + 1 \times 0 + 3 \times (-1)) + (1 \times 1 + 3 \times 0 + 2 \times (-1)) =$$

位置 (1,0):

$$C_{10} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 3 \\ 2 & 0 & 1 \end{pmatrix} \odot \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} = (1 \times 1 + 2 \times 0 + 1 \times (-1)) + (0 \times 1 + 1 \times 0 + 3 \times (-1)) + (2 \times 1 + 0 \times 0 + 1 \times (-1)) =$$

位置 (1,1):

$$C_{11} = \begin{pmatrix} 2 & 1 & 3 \\ 1 & 3 & 2 \\ 0 & 1 & 1 \end{pmatrix} \odot \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} = (2 \times 1 + 1 \times 0 + 3 \times (-1)) + (1 \times 1 + 3 \times 0 + 2 \times (-1)) + (0 \times 1 + 1 \times 0 + 1 \times (-1)) =$$

因此, 卷积输出: $C = \begin{pmatrix} -2 & -1 \\ -2 & -3 \end{pmatrix}$

步骤 2: ReLU 激活

对卷积输出 C 的每个元素应用 ReLU 函数: $f(x) = \max(0, x)$

$$R = \max(0, C) = \begin{pmatrix} \max(0, -2) & \max(0, -1) \\ \max(0, -2) & \max(0, -3) \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

步骤 3: 最大池化

使用 2×2 窗口, 步长为 1, 对 ReLU 输出 R 进行最大池化。

由于 R 是 2×2 矩阵, 使用 2×2 窗口且步长为 1, 输出尺寸为: $(2 - 2) / 1 + 1 = 1$, 所以输出 P 为 1×1 矩阵。

$$P = \max\{R_{00}, R_{01}, R_{10}, R_{11}\} = \max\{0, 0, 0, 0\} = 0$$

因此, 最大池化输出: $P = (0)$

最终答案:

- 卷积输出 C: $\begin{pmatrix} -2 & -1 \\ -2 & -3 \end{pmatrix}$

- ReLU 激活输出 R: $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
- 最大池化输出 P: (0)

12.4.4 经典 CNN 架构

- **AlexNet (2012):** 在 ILSVRC-2012 竞赛中取得巨大成功, 标志着深度学习时代的到来。其关键贡献包括:
 - 使用 ReLU 作为激活函数, 解决了梯度消失问题。
 - 使用 Dropout 和数据增强来减轻过拟合。
 - 使用重叠池化 (Overlapping pooling) 来避免平均池化的模糊效应。
 - 使用双 GPU 进行训练, 加快了速度。
- **VGG (2015):** 探索了网络深度对性能的影响。其特点是:
 - 网络非常深 (16 或 19 层)。
 - 使用了非常小的 3×3 卷积核。通过堆叠两个 3×3 卷积核可以获得与一个 5×5 卷积核相同的感受野, 但参数更少, 非线性更强。
- **ResNet (2014):** 解决了”网络退化”问题 (即更深的网络性能反而下降)。
 - **核心思想:** 引入”残差块 (Residual Block)”和”跳跃连接 (Skip Connection)”。网络不再直接学习目标映射 $H(x)$, 而是学习残差映射 $F(x) = H(x) - x$ 。原始的输入 x 通过跳跃连接直接加到输出上, 即 $H(x) = F(x) + x$ 。
 - **优点:** 如果恒等映射是最优的, 网络只需将 $F(x)$ 学为 0 即可, 这比学习一个恒等变换容易得多。这使得训练非常深的网络 (如超过 100 层) 成为可能。
- **U-Net (2015):** 为生物医学图像分割而设计的经典架构。
 - **结构:** 一个对称的编码器-解码器结构 (U 形)。编码器部分 (下采样路径) 用于提取上下文特征, 解码器部分 (上采样路径) 用于精确定位。
 - **关键创新:** 编码器和解码器之间的长跳跃连接 (skip connection)。它将编码器中对应层级的特征图直接拼接到解码器的特征图上, 使得解码器能够利用底层的、高分辨率的特征, 从而实现更精确的分割。
 - **影响:** U-Net 及其变体 (如 3D U-Net, V-Net) 已成为医学图像分割领域最主流和最成功的模型之一。

12.5 注意力机制 (Attention Mechanisms)

注意力机制模仿人类的视觉系统，让模型能够动态地关注输入中更重要的部分。

12.5.1 通道与空间注意力

- **SE Block (Squeeze-and-Excitation)**: 一种通道注意力机制。
 1. **Squeeze**: 通过全局平均池化将每个通道的特征图压缩成一个单一数值。
 2. **Excitation**: 使用一个简单的全连接网络（瓶颈结构）来学习通道之间的相关性，并为每个通道生成一个权重（注意力分数）。
 3. **Scale**: 将学习到的权重乘以原始特征图的对应通道，从而增强重要特征通道，抑制不重要通道。
- **CBAM (Convolutional Block Attention Module)**: 一个结合了通道注意力和空间注意力的模块。它首先应用一个通道注意力模块，然后串联一个空间注意力模块，依次对特征图在通道和空间维度上进行提炼。

12.5.2 自注意力与 Transformer

Transformer 模型完全基于自注意力机制，最初用于自然语言处理，后被成功应用于计算机视觉。

- **核心组件：缩放点积注意力 (Scaled Dot-Product Attention)**。
 - **输入**: 查询 (Query, Q)，键 (Key, K)，值 (Value, V)。
 - **过程**:
 1. 计算 Q 和所有 K 的点积，得到相似度分数。
 2. 对分数进行缩放（除以 $\sqrt{d_k}$ ），以防止梯度过小。
 3. 应用 Softmax 函数将分数转换为权重。
 4. 将权重与对应的 V 进行加权求和。
 - **公式**: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$
- **多头注意力 (Multi-Head Attention)**: 并行地运行多次缩放点积注意力，每次使用 Q, K, V 的不同线性投影。这允许模型同时关注来自不同表示子空间的信息。最后将所有“头”的输出拼接并再次进行线性投影。
- **视觉 Transformer (ViT)**: 将 Transformer 应用于图像分类。
 1. 将图像分割成一系列固定大小的图像块 (patches)。
 2. 将每个图像块展平并通过线性投影转换为一个向量 (token)。

3. 添加位置编码（Positional Encoding）来保留空间信息。
 4. 将这些 tokens 序列送入一个标准的 Transformer 编码器进行处理。
 5. 通常会添加一个特殊的 '[class]' token，其在 Transformer 输出端的对应向量被用于最终的分类任务。
- **Transformer 在医学影像中的应用：**近年来发展迅速，涌现出许多优秀模型。
 - **TransUNet：**结合了 Transformer 和 U-Net，利用 Transformer 强大的全局上下文建模能力作为编码器，再通过 U-Net 的解码器和跳跃连接恢复空间分辨率，用于图像分割。
 - **UNETR：**一个用于 3D 医学图像分割的纯 Transformer 模型。它将 3D 图像块作为输入，使用 ViT 作为编码器，并设计了一个新颖的解码器结构将特征上采样并与编码器特征结合。

12.6 生成模型 (Generative Models)

生成模型的目标是学习训练数据的潜在分布，并能够生成新的、与真实数据相似的样本。

12.6.1 生成对抗网络 (Generative Adversarial Networks - GAN)

- **基本原理：**由两个网络组成——生成器（Generator）和判别器（Discriminator），它们进行一场“猫捉老鼠”的对抗游戏。
 - **生成器 G：**接收一个随机噪声向量 z ，并试图生成一个以假乱真的图像 $G(z)$ 。
 - **判别器 D：**接收一个图像（真实的或生成的），并试图判断其真伪。
- **训练过程：**两者交替训练。D 的目标是最大化正确分类的概率，而 G 的目标是最小化 D 的分类准确率（即最大化 D 犯错的概率）。其目标函数（minimax game）为：

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

- **重要变体：**
 - **DCGAN (Deep Convolutional GAN)：**将卷积层引入 GAN 的生成器和判别器中，显著提升了生成图像的质量和训练的稳定性。
 - **CycleGAN：**用于非成对的图像到图像转换（unpaired image-to-image translation）。例如，将普通马的照片转换为斑马的照片，而不需要成对的马-斑马图像。其核心是引入了“循环一致性损失（cycle consistency loss）”，确保图像转换再转回后能恢复原样。

12.6.2 扩散模型 (Diffusion Models)

扩散模型是近年来兴起的一类强大的生成模型，以其高质量的生成结果和稳定的训练过程而闻名，解决了 GAN 训练不稳定的问题。

- **基本原理：**包含两个过程。
 1. **前向过程（加噪）：**这是一个固定的马尔可夫链，它在多个时间步（ T ）中，逐渐向真实图像 x_0 添加高斯噪声，直到图像最终变成一个纯粹的各向同性高斯噪声 x_T 。
 2. **反向过程（去噪）：**这是学习的核心。训练一个神经网络（通常是 U-Net 结构），从纯噪声 x_T 开始，逐步地、迭代地去除噪声，最终恢复出原始图像 x_0 。该网络学习预测在每个时间步被添加的噪声。
- **训练：**目标是最大化训练数据的似然，通过优化一个变分上界来实现。这在实践中简化为让模型预测的噪声与前向过程中实际添加的噪声之间的均方误差最小。
- **采样：**从一个标准正态分布中采样一个噪声图像，然后反复应用训练好的去噪网络 T 次，即可生成一张新图像。
- **应用：**在医学影像中，扩散模型已被用于图像重建、图像合成（如从一种模态生成另一种模态）、以及生成高分辨率的合成数据以增强数据集。