Computer Vision

Contents

1. 数字图像基础

1.1. 数字图像的表示

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

1.2. 图像分辨率

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

1.3. 视觉动态范围

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

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

1.4. 数字图像的基本操作

1.4.1. 点运算

点运算是指对图像中的每个像素独立进行操作. 常见的点运算包括:

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

1.4.2. 代数运算

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

1.4.3. 逻辑运算

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

2. 图像变换

2.1. 空间域变换

2.1.1. 齐次坐标

齐次坐标是对笛卡尔坐标的一种扩展, 允许我们使用矩阵运算来表示各种几何变换. 对于二维空间中的点 (x, y), 其齐次坐标表示为 (x, y, w), 其中 w

是一个非零的缩放因子. 通常情况下, 我们可以将 w 设为 1, 这样点 (x, y) 在齐次坐标中表示为 $(x, y, 1)$

2.1.2. 欧式变换

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

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

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

2.1.3. 相似变换

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

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

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

2.1.4. 仿射变换

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

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

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

2.1.5. 投影变换

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

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

它的自由度为 8, 包括六个线性变换参数和两个平移参数.

2.1.6. 3D 空间中的变换的自由度

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

2.1.7. 灰度变换

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

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

2.2. 频域变换

用于分析图像中的频率成分, 将图像表示为不同频率和振幅的模式之和.

2.2.1. 连续 Fourier 变换

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

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

2.2.2. 离散 Fourier 变换 (DFT)

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

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

3. 图像滤波与数字滤波器

3.1. 图像卷积与相关

3.1.1. 定义

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

- 相关:

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

- 卷积:

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

3.1.2. 相关和卷积的关系

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

3.1.3. 卷积运算的性质

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

3.1.4. 卷积定理

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

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

3.2. 经典数字滤波器

3.2.1. 滤波和滤波器

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

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

3.2.2. 图像噪声

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

3.2.3. 高斯滤波

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

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

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

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

3.2.4. 双边滤波

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

考虑像素 q 的邻域 S, 双边滤波的计算公式如下:

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

其中

1. $f(\|p - q\|)$ 是 **空间权重函数**, 通常采用高斯函数, 用于衡量像素 p 与像素 q 之间的空间距离.
2. $g(|I(p) - I(q)|)$ 是 **像素值权重函数**, 通常也采用高斯函数, 用于衡量像素 p 与像素 q 之间的灰度差异.
3. W 是 **归一化因子**, 用于确保权重和为 1.

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

3.2.5. Wiener 滤波

图像采集过程中造成图像退化, 退化模型为

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

这里的 $h(x, y)$ 是退化函数, $n(x, y)$ 是噪声.

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

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

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

4. 形态学与基于深度学习的图像去噪

4.1. 结构元素

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

4.2. 二值图像的膨胀

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

若原图像为 A, 结构元素为 B, 则膨胀操作定义为:

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

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

用算法流程表示就是

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

膨胀的应用: 填补图像中的小孔洞, 连接断开的前景区域.

4.3. 二值图像的腐蚀

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

若原图像为 A, 结构元素为 B, 则腐蚀操作定义为:

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

用算法流程表示就是

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

4.4. 开运算与闭运算

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

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

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

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

5. 图像增强

5.1. 基于空域的图像增强

5.1.1. 灰度直方图

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

5.1.2. 直方图均衡化

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

直方图均衡化的步骤如下:

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

5.1.3. 局部直方图均衡化

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

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

5.1.4. 直方图匹配 (规定化)

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

直方图匹配的步骤如下:

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

5.1.5. 空间域滤波器

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

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

这里的 $w(s,t)$ 是滤波器的权重矩阵 (卷积核).

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

5.1.6. 空间域平滑

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

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

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

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

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

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

$$g(x,y)=\operatorname{median}\{f(x+s,y+t),-a\le s\le a,-b\le t\le b\}$$

5.1.7. 空间域锐化

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

5.1.8. 梯度锐化法

在离散图像 $f(x,y)$ 中, 其梯度可以通过以下方式近似计算:

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

它们可以分别对应两个卷积核:

$$G_x=[1,-1],G_y=[1;-1]^T$$

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

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

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

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

5.2. 基于频域的图像增强

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

5.2.1. 理想低通滤波器 ILPF

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

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

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

5.2.2. Butterworth 低通滤波器 BLPF

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

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

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

6. 图像退化和复原

6.1. 图像退化

6.1.1. 图像退化类型

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

6.1.2. 图像退化的数学模型

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

在频域中表示为:

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

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

常见的噪声类型有

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

6.1.3. LTI 系统近似

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

6.1.4. 点扩散函数 (PSF)

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

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

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

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

6.1.5. 离散系统的退化模型

在离散图像处理中, 退化模型可以表示为:

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

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

$$[g]=[H][f]+[n]$$

展开来就是

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

其中 H_i 被定义为

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

6.2. 图像复原

图像复原是指通过已知的退化模型和噪声统计特性, 来估计原始图像.

图像复原和图像增强的区别

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

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

6.3. 无约束图像复原

6.3.1. 代数复原方法

假设图像退化模型为

$$n = g - Hf$$

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

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

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

$$f\!\!=\!\left(H^T H\right)^{-1} H^T g = H^{-1} g$$

6.3.2. 逆滤波

逆滤波是一种基于频域的图像复原方法. 假设图像退化模型为

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

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

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

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

6.4. 有约束图像复原

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

6.4.1. 代数有约束复原方法

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

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

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

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

我们也可以设置别的约束条件, 如 $\|g\|^2 = \|f\|^2$.