

# 1 图像处理

## 1.1 图像表示方法

图像在本质上是连续的图形, 但计算机并不能存储无限的信息, 因此需要对图像进行离散化处理.

**定义 1.1 位图与矢量图** 位图 (Bitmap) 是由像素点组成的矩阵, 每个像素点有一个颜色值, 通过这些像素点的颜色值来表示图像. 矢量图 (Vector Graphics) 是由数学方程和几何图形 (如直线, 曲线, 多边形等) 来表示图像.

两者之间的差异可以由下表总结:

	位图	矢量图
优点	可以表示复杂的图像细节	可以无限放大而不失真
缺点	放大后会失真	难以表示复杂的图像细节

表 1: 位图与矢量图的比较

## 1.2 图像滤波

### 1.2.1 卷积

卷积是图像处理中常用的操作, 用于图像的平滑, 锐化, 边缘检测等.

**定义 1.2 卷积** 二维离散信号 (如图像) 的卷积定义为:

$$(I * K)(x, y) = \sum_{m=-M}^M \sum_{n=-N}^N I(x-m, y-n) K(m, n)$$

其中  $I$  是输入图像,  $K$  是卷积核 (滤波器),  $(x, y)$  是图像中的一个像素位置. 在图形学的实际应用中, 卷积核通常经过预翻转操作, 此时卷积定义为

$$(I * K)(x, y) = \sum_{m=-M}^M \sum_{n=-N}^N I(x+m, y+n) K(m, n)$$

卷积和傅里叶变换密切相关.

**定理 1.3 卷积定理** 函数卷积的傅里叶变换等于各自傅里叶变换的乘积, 即

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$$

这样,可以观察图像和滤波器进行傅里叶变换的频谱来大致得出滤波后的结果,从而针对性地设计滤波器.

### 1.2.2 图像模糊

最简单的模糊滤波器是均值滤波器.

**定义 1.4 均值滤波器** 均值滤波器的卷积核为

$$\mathbf{K}_{\text{mean}} = \frac{1}{(2k+1)^2} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

根据卷积的定义,均值滤波器的作用是取目标像素邻近的  $(2k+1) \times (2k+1)$  个像素求平均值.

均值滤波器地频谱主要集中在低频部分,可以去除高频噪声,但会模糊图像细节.然而,它的频谱有呈十字形向外放射的部分,因此使用均值滤波器可能导致细节错误.

高斯滤波器是一种效果更好的模糊滤波器<sup>1</sup>.它的连续形式为

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

其中  $\sigma$  为方差,可以控制模糊程度.实际使用时,可以截取目标附近的  $(2k+1) \times (2k+1)$  个像素,根据上述函数计算权重并归一化得到卷积核,然后按照类似均值滤波的计算方式完成卷积.

在数学上可以证明,高斯滤波器的频谱仍然是高斯函数,只保留低频部分,不会出现像均值滤波器那样的走样.

### 1.2.3 边缘提取

所谓图像中的边缘,是指图像中一个色块与另一个色块的分界线,通常对应图像中颜色变化较大的部分.我们可以用估计梯度的算子来提取边缘.

**定义 1.5 Sobel 算子** Sobel 算子是一种常用的边缘检测算子,它通过两个卷积核来计算图像在水平和垂直方向上的梯度:

$$\mathbf{K}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

其中  $\mathbf{K}_x$  用于检测水平边缘,  $\mathbf{K}_y$  用于检测垂直边缘.

对图像  $\mathbf{I}$  进行卷积后,可以得到水平和垂直方向的梯度:

$$G_x = \mathbf{I} * \mathbf{K}_x, \quad G_y = \mathbf{I} * \mathbf{K}_y$$

<sup>1</sup>在 Adobe Photoshop 中对应高斯模糊选项.

为了得出各个方向上的边缘, 可以计算指定位置的梯度向量的模长:

$$G = \sqrt{G_x^2 + G_y^2}$$

除了使用梯度提取边缘, 还可以使用二阶导数提取边缘.

**定义 1.6 Laplacian 算子** Laplacian 算子是一种二阶导数算子, 用于检测图像中的边缘. 其卷积核为

$$\mathbf{K}_{\text{Lap}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

对图像  $I$  进行卷积后, 可以得到 Laplacian 响应:

$$L = I * \mathbf{K}_{\text{Lap}}$$

Laplacian 算子对噪声较为敏感, 因此通常在使用前会先对图像进行高斯模糊处理, 这种方法称为 LoG(Laplacian of Gaussian).

### 1.3 图像补全与融合

在图像补全时, 我们需要对图像缺失的部分进行补充, 并且使得结果看起来更为自然, 即满足下面两个条件:

1. **空间局部性**: 同一个物体上相邻的部分应当相似.
2. **奥卡姆剃刀原理**: 如无必要, 勿增实体.

在数学上, 我们可以用以下的办法描述问题: 考虑图像  $I$  的待补全区域  $\Omega$  以及其边界  $\partial\Omega$ . 定义  $f(x, y)$  为  $\Omega \cup \partial\Omega$  上的函数表示待填补的颜色,  $f^*(x, y)$  为  $\mathbb{C}_I\Omega$  上的函数表示已知的颜色. 于是优化目标为

$$\min_f \iint_{\Omega} \|\nabla f\|^2 dS, \quad \text{s.t. } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

在图像融合时, 我们希望补全区域的颜色梯度与源图像的颜色梯度尽可能接近. 定义  $g(x, y)$  为  $I$  上的函数表示原图像的颜色, 那么优化问题为

$$\min_f \iint_{\Omega} \|\nabla f - \nabla g\|^2 dS, \quad \text{s.t. } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

由于图像补全是图像融合的特殊情形 ( $\nabla g = 0$ ), 因此考虑后面的问题即可. 根据 Euler-Lagrange 方程, 上述优化问题的解满足

$$\nabla^2 f = \nabla^2 g, \quad \text{s.t. } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

其中  $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  为 Laplacian 算子. 在离散情形下, 可以用差分法写出  $\nabla^2 f$  的值:

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

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

于是

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

按照一定的方式排列所有  $(x, y) \in \Omega$  即可得到一个线性方程组. 求解这个线性方程组即可得到每一点的像素值.

## 1.4 图像分割

大多数图片都有前景与背景, 但图像只有每个像素点的信息. 为了从颜色信息还原出前景与背景信息,