

第3章 图像分割

图像最重要的特征之一是边缘，是图像分割的主要依据，本章讲述图像分割的基本概念和方法，重点讲述边缘的基本特征、经典边缘检测算子及其模板，包括 Roberts 算子、Sobel 算子、Kirsch 算子、Prewitt 算子、拉普拉斯算子等微分算子。同时，讲述纹理图像分割和二值图像分割的概念及其算法。最后介绍属于特征提取技术的霍夫（Hough）变换。

3.1 引言

在一幅图像中，景物往往由众多的目标组成，反映在图像中是众多的区域。每个目标或区域可以进一步分解成一些具有某些特征的最小成分——基元。因此，在获取图像以后，首先要对复杂的景物作出分解。目标的分解主要根据图像中存在的边缘、纹理、形状、目标表面主方向等图像特征，把图像分解成一系列的目标或区域，直至最终形成基元。这一过程称为图像分割。图像分割的目的是为了识别和理解图像。

图像最基本的特征是边缘。边缘是指其周围像素灰度有阶跃状变化或屋顶状变化的那些像素的集合，它存在于目标与背景、目标与目标、区域与区域、基元与基元之间。因此，边缘是图像分割所依赖的最重要的特征，也是纹理特征中的重要信息源和形状特征的基础。而且，图像的纹理形状特征的提取又常常要依赖于图像分割。

图像的边缘提取和分割是图像处理、图像分析和计算机视觉等研究领域的经典课题之一，有较长的研究历史和一定的深度和难度。

统计模式识别认为，图像可能包含一个或多个物体，并且每个物体属于若干事先定义的类型、范畴或模式之一。虽然模式识别可以用多种方法实现，但在此只关心用数字图像处理技术对它的实现。

在给定一幅含有多个物体的数字图像的条件下，模式识别过程如图 3.1 所示，由 3 个主要阶段组成。

第一阶段称为图像分割或物体分离阶段。在该阶段中检测出各个物体，并且把它们的图像和其余景物分离。这一过程也可以称为图像预处理。

第二阶段是特征提取，一个物体某个可度量性质是度量值，而特征是一个或多个度量值的函数。计算特征是为了对物体的一些重要特征进行定量估计。特征抽取过程产生了一组特征，把它们组合在一起，就形成了特征向量。

第三阶段称为分类。它的输出仅仅是一种决策，确定每个物体应该归属的类别。每个物体被识别为某一特定类型，它是通过一个分类过程加以实现的。分类以特征向量作为依据。

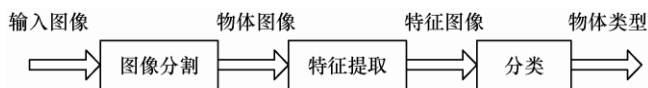


图 3.1 模式识别的三个阶段

本章主要讨论边缘检测和图像分割，为后续的特征提取和模式识别做图像预处理。

3.2 图像分割处理

利用计算机进行数字图像处理的目的有两个：一是产生更适合人类视觉观察和识别的图像，二是希望计算机能够自动识别和理解图像。无论是为了何种目的，图像处理的关键问题是对包含有大量各式各样景物信息的图像进行分解。分解的最终结果就是图像被分成一些具有各种特征的最小成分，这些成分就称为图像的基元。产生这些基元的过程就是图像分割的过程。

可以把图像分割处理定义为将数字图像划分成互不相交的区域的过程。应用数字图像处理技术，设法分离图像中的物体，把图像分裂成像素集合，每个像素集合代表一个物体的图像。数字图像分割的结果可以用于数字图像分析。

图像分割可以采用 3 种不同的原理来实现。在区域方法中，把个别像素划分到各个物体或区域中。在边界方法中，只需确定存在于区域的边界。在边缘方法中，先确定边缘像素，并把它们连接在一起以构成所需的边界。

图像分割作为图像处理领域中极为重要的内容之一，是实现图像分析与理解的基础。从概念上来说，所谓图像分割就是按照一定的原则将一幅图像或景物分为若干个部分或子集的过程。

图像分割也可以理解为将图像中有意义的特征区域或者需要应用的特征区域提取出来，这些特征区域可以是像素的灰度值、物体轮廓曲线、纹理特性等，也可以是空间频谱或直方图特征等。

在图像中用来表示某一物体的区域，其特征都是相近或相同的，但是不同物体的区域之间，特征就会明显变化。目前已经提出的图像分割方法很多，从分割依据的角度来看，图像的分割方法可以分为相似性分割和非连续性分割。

相似性分割就是将具有同一灰度级或相同组织结构的像素聚集在一起，形成图像的不同区域；非连续性分割就是首先检测局部不连续性，然后将它们连接在一起形成边界，这些边界将图像分成不同的区域。由于不同种类的图像，不同的应用场合，需要提取的图像特征是不同的，当然对应的图像特征提取方法也就不同，所以并不存在一种普遍适应的最优方法。

根据分割过程中处理策略的不同，分割算法又可分为并行算法和串行算法。在并行算法中，所有判断和决定都可独立和同时地进行；而在串行算法中，后续处理过程要用到早期处理的结果。在实际应用中，常常将多种分割算法相结合，以取得更好的分割效果。

图像分割也可以按照如下的标准分类：基于区域的分割方法，包括阈值分割法、区域生长和分裂合并法、聚类分割法等；基于边界的分割方法，包括微分算子法、串行边界技术等；基于区域和边界技术相结合的分割方法。

3.2.1 图像分割的基本方法

图像分割的目的是把一个图像分解成它的子区域或对象，以便对每一目标进行测量。测量结果的质量依赖于图像分割的质量。下面介绍一些常用的图像分割方法。

1. 直方图分割与图像二值化

最简单的分割方法建立在图像灰度直方图分析的基础上。如果一个图像是由明亮目标在一个暗的背景上组成的，那么其灰度直方图将显示两个最大值，一个是由目标点产生的峰值，另一个峰值是由背景点产生的。

如果目标和背景之间反差足够大，则直方图中的两个峰值相距甚远，这样就可以选择一个灰度阈值 T 将两个最大值隔开。图像中所有大于 T 的灰度值可用数值 1 取代，而所有小于或等

于 T 的灰度值可用数值 0 取代。这样就生成一个二值图像 $g(x,y)$ ，其中目标点用 1 表示，背景点用 0 表示。如果图像由两个以上背景成分所组成，则直方图将显示多重峰值，分割可以取多重阈值来完成。二值图像 $g(x,y)$ 可表示为

$$g(x,y) = \begin{cases} 1 & f(x,y) \geq T \\ 0 & f(x,y) < T \end{cases} \quad (3.1)$$

二值图像是指图像上的所有像素点的灰度值只有两种可能，不是“0”就是“255”，也就是说，整个图像呈现出明显的黑白效果。通常，采用归一化方法，用“0”、“1”表示“0”和“255”。

为了得到理想的二值图像，一般采用阈值分割技术。它对物体与背景有较强对比的图像的分割特别有效，计算简单而且总能用封闭、连通的边界定义不交叠的区域。所有灰度大于或等于阈值的像素被判决为属于物体，灰度值用“255”表示，否则这些像素点被排除在物体区域以外，灰度值为“0”，表示背景。这样，物体的边界就成为这样一些内部的点的集合，这些点都至少有一个邻域点不属于该物体。如果感兴趣的物体在内部有均匀一致的灰度值，并且其处在一个具有另外一个灰度值的均匀背景下，那么使用阈值法可以得到较好的分割效果。如果物体与背景的差别不在灰度值上，如纹理不同，可以将这个性质转换为灰度的差别，然后利用阈值化技术来分割该图像。

为了使分割适用性更强，设计的图像分割算法应该可以自动选择阈值。在数字图像处理的实际应用中，基于物体、环境和应用域等知识的图像分割算法比基于固定阈值的算法更具有普遍性和适应性。这些知识包括：对应于物体的图像灰度特性、物体的尺寸、物体在图像中所占的比例、图像中不同类型物体的数量等。图像直方图具有灰度统计特性，通常被用来作为分割图像的工具。

2. 区域生长

另一种同样利用区域均匀性进行分割的方法称为区域生长（Region Growing）法。首先，在图像中挑选一个或一个以上的种子点，种子点的数目等于被检测区域的数目。然后，规定一个像素之间的相似性准则，最简单的是基于像素灰度值的准则。例如，两个像素的灰度值之间相差的绝对值小于给定的阈值，则称它们是相似的。区域生长法开始时，先检查种子点周围的全部点。如果一个点足够类似于它的种子点，则它被划归为该种子点区域。对每一个区域都重复这一过程，即检查围绕相应区域种子点的各像素，直到在这个图像中所有的点都被分类划分为止。

区域生长是把图像分割成若干小区域，比较相邻小区域的特征相似性，将相似的区域逐一合并，最后形成特征不同的区域。区域生长法要注意 3 个问题：取定区域的数目、选择特征、确定相似性准则。特征相似性是构成合并区域的基本准则，相邻性是指所取的邻域方式。区域生长根据所用的邻域方式和相似性准则的不同，可以简单分为：像素与像素、像素与区域、区域与区域。

(1) 简单区域扩张法

以图像的某个像素为生长点，比较相邻像素的特征，将特征相似的相邻像素合并为同一区域；以合并的像素为生长点，重复整个过程，最终形成具有相似特征的像素的集合。

(2) 像素与区域生长

比较单个像素的特征与其相邻区域的特征，如果相似则将该像素合并到相邻区域中。比较已经存在区域像素灰度平均值与该区域邻接的像素灰度值，如果差值小于阈值则合并，否则不合并。这种方法与起始像素有关，起始位置不同分割结果有差异。

(3) 区域与区域生长

这种方法的基本原则是将图像分割成若干子块，比较相邻子块的相似性，如果相似则合并，否则不合并。

下面介绍不依赖于起始点的方法：

- ① 设灰度差的阈值为 0，用像素与区域生长法将具有相同灰度的像素合并到同一区域，得到图像的初始分割图像；
- ② 从被分割图像的一个小区域开始，求出相邻区域间的灰度差，将差值最小的相邻区域合并；
- ③ 重复上述操作，将区域逐一合并，终止规则为没有像素满足加入某个区域的条件时区域生长停止。

3. 使用阈值进行图像分割

使用阈值进行图像分割是一种区域分割技术，对物体与背景有较强对比的图像的分割特别有效。该方法计算简单，而且总能用封闭且连通的边界定义不交叠的区域。

(1) 局部阈值分割

阈值分割法分为全局阈值分割法和局部阈值分割法。局部阈值分割法是将原图像划分成较小的图像，并对每个子图像选取相应的阈值。在阈值分割后，相邻子图像之间的边界处可能产生灰度级的不连续性，因此需用平滑技术进行排除。

局部阈值分割法虽然能改善分割效果，但存在以下几个缺点：

- ① 每幅子图像的尺寸不能太小，否则统计出的结果无意义；
- ② 每幅图像的分割是任意的，如果有一幅子图像正好落在目标区域或背景区域，而根据统计结果对其进行分割，也许会产生更差的结果；
- ③ 由于局部阈值法对每一幅子图像都要进行统计，所以速度很慢，难以适应实时性的要求。

(2) 全局阈值分割

采用阈值确定边界的最简单做法是在整幅图像中，将灰度阈值设置为常数。如果背景的灰度值在整个图像中可合理地认为恒定，而且所有物体与背景都具有几乎相同的对比度，那么只要选择了正确的阈值，使用一个固定的全局阈值分割图像一般会有较好的效果。

全局阈值分割方法在图像处理中应用比较多，它在整幅图像内采用固定的阈值分割图像。对于比较简单的图像，可以假定物体和背景分别处于不同的灰度级，图像被零均值高斯噪声污

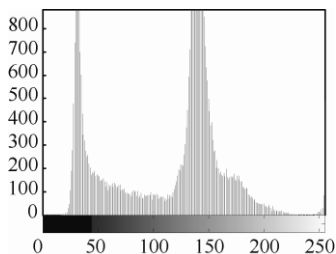


图 3.2 原图像直方图

染，所以图像的灰度分布曲线近似认为是由两个正态分布函数 (μ_1, σ_1^2) 和 (μ_2, σ_2^2) 叠加而成的，图像的直方图将会出现两个分离的峰值，如图 3.2 所示。对于这样的图像，分割阈值可以选择直方图的两个波峰间的波谷所对应的灰度值作为分割的阈值。这种分割方法不可避免地会出现误分割，使一部分本属于背景的像素被判决为物体，属于物体的一部分像素同样会被误认为是背景。可以证明，当物体的尺寸和背景相等时，这样选择阈值可以使错误分割的概率达到最小。在大多数情况下，由于图像的直方图在波谷附近的像素稀疏，因此这种方法对图

像的分割影响不大。可以推广到具有不同灰度均值的多物体图像的分割。

迭代式阈值选择算法是对上一种方法的改进，它首先选择一个近似阈值 T ，将图像分割成

两部分 R_1 和 R_2 ，计算区域 R_1 和 R_2 的均值 μ_1 和 μ_2 ，选择新的分割阈值 $T=(\mu_1+\mu_2)/2$ ，重复上述步骤直到 μ_1 和 μ_2 不再变化为止。

根据如图 3.2 所示的灰度直方图选择不同的阈值，产生的图像分割效果也不同，不同阈值 T 对分割结果的影响如图 3.3 所示。因此，对于不同的图像以及对于图像处理结果的不同要求，必须正确地选择阈值，以便得到更好的分割效果。

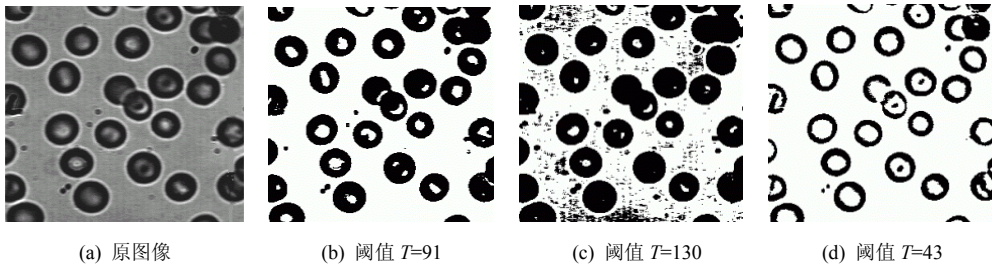


图 3.3 不同阈值对分割结果的影响

综上所述，当使用阈值规则进行图像分割时，所有灰度值大于或等于阈值的像素都被判属于物体。所有灰度值小于该阈值的像素被排除在物体之外。

(3) 最佳自适应阈值

在许多情况下，背景的灰度值并不是常数，物体与背景的对比度在图像中也有变化。这时，一个在图像中某一区域分割效果良好的阈值，在其他区域可能效果很差。在这种情况下，把灰度阈值取成一个随图像中位置缓慢变化的函数值是适宜的。或者，对图像采用划分区域处理的方法，这种算法就称为自适应阈值。

除非图像中的物体有陡峭的边沿，否则灰度阈值的抽取对物体的边界定位和整体的尺寸有很大的影响。这就意味着后续的尺寸，特别是面积的测量对于灰度阈值的选择很敏感。由于这个原因，需要一个最佳的、或至少是具有一致性的方法确定阈值。在此，可以采用直方图技术。

一幅含有一个与背景明显对比的物体的图像具有包含双峰的灰度直方图，如图 3.4 所示。两个尖峰对应于物体内部和外部数目较多的点。两峰间的谷对应于物体边缘附近数目相对较少的点。在类似这样的情况下，通常使用直方图来确定灰度阈值。

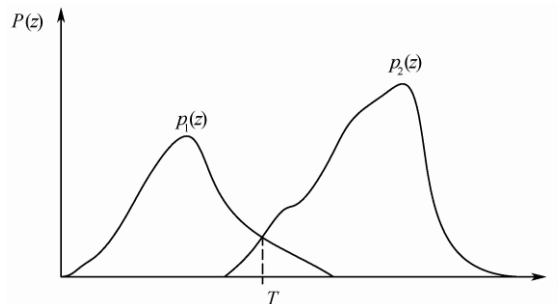


图 3.4 一幅图像中两个区域的灰度级概率密度函数

利用阈值 T 对物体面积进行计算的定义为

$$A = \int_T^{\infty} H(D) dD \quad (3.2)$$

式中, D 为目标中的灰度级, $H(D)$ 为灰度直方图。

显然, 如果阈值对应于直方图的谷, 阈值从 T 增加到 $T + \Delta T$ 只会引起面积略微减小。因此, 如果把阈值设在直方图的谷上, 那么可以把阈值选择中的小误差对面积的影响降到最低。一种更可靠的方法是把阈值设在相对于两峰的某个固定位置, 如中间位置上, 这两个峰分别代表物体内部和外部典型的灰度值。

假设一幅图像仅包含两个主要灰度级区域。令 z 表示灰度级的值, 而且将这些值看成是随机量, 并且将图像的直方图看成是概率密度函数 (PDF) 的估计 $P(z)$ 。在两个灰度级区域中, 一个是图像中亮区域的密度, 另一个是暗区域的密度。如果密度的表达式已知或者进行了假设, 则它能够确定一个最佳阈值 (具有最低分割误差), 将图像分割为两个可以区分的区域。

如图 3.4 所示, 假设两个 PDF 中, $p_1(z)$ 表示图像中目标的灰度级, $p_2(z)$ 对应背景的灰度级。描述图像中整体灰度级变化的混合概率密度函数为

$$P(z) = P_1 p_1(z) + P_2 p_2(z) \quad (3.3)$$

式中, P_1 为像素属于目标像素的概率, P_2 为像素属于背景像素的概率。假设任何像素不是属于目标就是属于背景, 那么有

$$P_1 + P_2 = 1 \quad (3.4)$$

选择一个阈值 T , 使得在判别一个像素是属于目标或者背景时的平均出错率降至最小。将一个背景点当成目标点进行分类时, 错误发生的概率为

$$E_1 = \int_{-\infty}^T p_2(z) dz \quad (3.5)$$

式(3.5)表示曲线 $p_2(z)$ 下方位于阈值 T 左边区域的面积。同样, 将一个目标点当成背景点进行分类, 错误发生的概率为

$$E_2 = \int_T^{\infty} p_1(z) dz \quad (3.6)$$

式(3.6)表示曲线 $p_1(z)$ 下方位于 T 右边区域的面积。判别出错率的整体概率为

$$E(T) = P_1 E_1(T) + P_2 E_2(T) \quad (3.7)$$

要选择出错最少的阈值, 需要对 $E(T)$ 进行微分运算, 表示为

$$\frac{dE(T)}{dT} = 0 \quad (3.8)$$

得到结果为

$$P_1 p_1(T) = P_2 p_2(T) \quad (3.9)$$

根据式(3.9)解出 T , 即为最佳阈值。如果 $P_1 = P_2$, 则最佳阈值 T 位于图 3.4 中曲线 $p_1(z)$ 和 $p_2(z)$ 的交点处。

另一种确定 T 的方法是由高斯密度来表示, 高斯密度用均值和方差表示为

$$P(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(z-\mu_1)^2}{2\sigma_1^2}\right] + \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(z-\mu_2)^2}{2\sigma_2^2}\right] \quad (3.10)$$

式中, μ_1 和 σ_1^2 分别是目标像素高斯密度的均值和方差, μ_2 和 σ_2^2 分别是背景像素的均值和方差。利用式(3.3), 并将式(3.10)代入式(3.9), 求出一般解, 则可以得到阈值 T , 表示为

$$AT^2 + BT + C = 0 \quad (3.11)$$

式中

$$A = \sigma_1^2 - \sigma_2^2$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2) \quad (3.12)$$

$$C = \sigma_1^2 \mu_2^2 - \sigma_2^2 \mu_1^2 + 2\sigma_1^2 \sigma_2^2 \ln \left(\frac{\sigma_2 P_1}{\sigma_1 P_2} \right)$$

由于二次方程有两个可能的解，所以要得到最佳的解，需要两个阈值。如果方差都相等，即 $\sigma^2 = \sigma_1^2 = \sigma_2^2$ ，则阈值可以表示为

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln \left(\frac{P_2}{P_1} \right) \quad (3.13)$$

如果 $P_1 = P_2$ ，则最佳阈值是均值的平均数。

3.2.2 边缘图像及分类

确定图像中的物体边界的另一种方法是，先检测每个像素及其直接邻域的状态，以决定该像素是否确实处于一个物体的边缘上。具有所需特性的像素称为边缘点。当图像中各个像素的灰度级用来反映各像素符合边缘像素要求的程度时，那么这种图像就称为边缘图像或边缘图。

如图 3.5 所示，当目标和背景的边缘清晰时，称为阶跃状边缘。当目标和背景的边缘是渐变时，称为屋顶状边缘。根据不同的边缘选择不同的边缘检测算子才能对图像进行边缘检测和有效的分割。

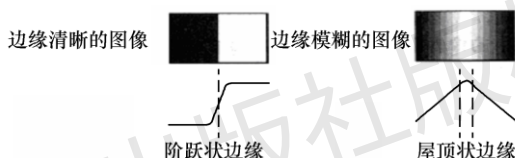


图 3.5 图像边缘示意图

3.2.3 边缘检测算子

边缘检测算子检查每个像素的邻域，并对灰度变化率进行量化，通常也包括方向的确定。有很多方法可以使用，其中大多数是基于方向导数掩模求卷积的方法。经典的、最简单的边缘检测方法是按原图像按像素的某邻域构造边缘检测算子。如梯度算子、Sobel 算子、拉普拉斯算子、Kirsch 算子和 Prewitt 算子等。

对于图像曲面 $z = f(x, y)$ ，定义它的 α 方向的方向导数在 (x, y) 点的值为

$$f'_\alpha(x, y) = \frac{\partial f(x, y)}{\partial x} \sin \alpha + \frac{\partial f(x, y)}{\partial y} \cos \alpha \quad (3.14)$$

$$f'_\alpha(x, y) = \lim_{h \rightarrow 0} \frac{f(x + h \sin \alpha, y + h \cos \alpha) - f(x, y)}{h} \quad (3.15)$$

式中， α 取顺时针方向。由定义容易验证二阶方向导数为

$$f''_\alpha(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} \sin^2 \alpha + 2 \frac{\partial^2 f(x, y)}{\partial x \partial y} \sin \alpha \cos \alpha + \frac{\partial^2 f(x, y)}{\partial y^2} \cos^2 \alpha \quad (3.16)$$

对于经过量化和离散化而获得的数字图像，这些微分符号均用差分符号来代替。经过量化和离散化而得到的数字图像差分形式为

$$\Delta_x f(i, j) = f(i, j) - f(i-1, j) \quad (3.17)$$

$$\Delta_y f(i, j) = f(i, j) - f(i, j-1) \quad (3.18)$$

x 方向、 y 方向的二阶差分分别为

$$\Delta_x^2 f(i, j) = \Delta_x f(i+1, j) - \Delta_x f(i, j) \quad (3.19)$$

$$\Delta_y^2 f(i, j) = \Delta_y f(i, j+1) - \Delta_y f(i, j) \quad (3.20)$$

$$\Delta_{xy}^2 f(i, j) = \Delta_x f(i, j+1) - \Delta_x f(i, j) \quad (3.21)$$

$$\Delta_{yx}^2 f(i, j) = \Delta_y f(i+1, j) - \Delta_y f(i, j) \quad (3.22)$$

下面介绍几种常用的边缘检测算子，如图 3.6 所示为一个像素的 8 邻域示意图，可以用坐标进行标注。例如，位于坐标 (i, j) 的一个像素 p 有 4 个水平和垂直的相邻像素，其坐标表示为

$$(i+1, j), (i-1, j), (i, j+1), (i, j-1)$$

这个像素集称为 p 的 4 邻域，用 $N_4(p)$ 表示，每个像素距离 (i, j) 一个单位距离。

p 的 4 个对角的相邻像素坐标表示为

$$(i-1, j-1), (i-1, j+1), (i+1, j+1), (i+1, j-1)$$

它们与 4 邻域点共同构成 p 点的 8 邻域，用 $N_8(p)$ 表示。下面讲述的边缘检测算子表达式都是基于图 3.6 所示的 8 邻域表示。

1. 梯度边缘算子

对于阶跃状边缘图像，在边缘点其一阶导数取极值。由此，对数字图像 $\{f(i, j)\}$ 的每个像素取它的梯度值，表示为

$$G(i, j) = \sqrt{[\Delta_x f(i, j)]^2 + [\Delta_y f(i, j)]^2} \quad (3.23)$$

适当取门限 T_g 做如下判断：若 $G(i, j) = T_g$ ，则 (i, j) 点为阶跃状边缘点， $\{G(i, j)\}$ 称为梯度算子的边缘图像。在有些问题中，只对边缘位置感兴趣，把边缘点标为“1”，非边缘点标为“0”，形成边缘二值图像。

2. Roberts 边缘算子

Roberts 边缘算子是一种利用局部差分算子寻找边缘的算子。表示为

$$R(i, j) = \{[\sqrt{f(i, j)} - \sqrt{f(i+1, j+1)}]^2 + [\sqrt{f(i+1, j)} - \sqrt{f(i, j+1)}]^2\}^{\frac{1}{2}} \quad (3.24)$$

式中， $f(i, j)$ 是具有整数像素坐标的输入图像。其中的平方根运算使该处理类似于人类视觉系统中发生的过程。如图 3.7 所示为 Roberts 的边缘算子模板。根据模板在图像处理中的运算特点，有时也将模板称为卷积核，运算过程为：将被模板覆盖的像素与模板相应位置处的数据先相乘再求和。

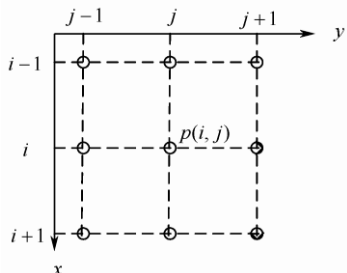


图 3.6 像素 8 邻域坐标示意图

1	0	0	1
0	-1	-1	0

图 3.7 Roberts 边缘算子模板

3. Sobel 边缘算子

如图 3.8 所示，两个卷积核形成了 Sobel 边缘算子。图像中的每个点都用这两个核做卷积。通常，一个核对垂直边缘响应最大，而另一个核则对水平边缘响应最大。两个卷积的最大值作

为该点的输出值。运算结果是一幅边缘幅度图像。

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

图 3.8 Sobel 边缘算子模板

对于阶跃状边缘图像，Sobel 提出一种检测边缘点的算子。对数字图像 $\{f(i, j)\}$ 的每个像素考察其上、下、左、右相邻点灰度的加权差，与之接近的邻点权重重大。据此，定义的 Sobel 算子表示为

$$S(i, j) = \left| [f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1)] - [f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1)] \right| \\ + \left| [f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1)] - [f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1)] \right| \quad (3.25)$$

适当取阈值 T_s ，做如下判断：若 $S(i, j) \geq T_s$ ，则 (i, j) 点为阶跃状边缘点，所有边缘点的集合构成 $\{S(i, j)\}$ ，为边缘图像。

4. 拉普拉斯 (Laplacian) 边缘算子

对于阶跃状边缘，二阶导数在边缘点出现零交叉，即边缘点两旁二阶导数取异号，据此，对数字图像 $\{f(i, j)\}$ 的每个像素，取它关于 x 轴方向和 y 轴方向二阶差分之和，表示为

$$\nabla^2 f(i, j) = \Delta_x^2 f(i, j) + \Delta_y^2 f(i, j) \\ = f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1) - 4f(i, j) \quad (3.26)$$

式(3.26)就是拉普拉斯边缘算子，这是一个与边缘方向无关的边缘检测算子。通常可以采用数字化方式，用如图 3.9 所示的模板中的一个来实现。

对于屋顶状边缘，在边缘点的二阶导数取极小值。对数字图像 $\{f(i, j)\}$ 的每个像素取它关于 x 轴方向和 y 轴方向二阶差分之和的相反数，即拉普拉斯边缘算子的相反数。则表示为

$$L(i, j) = -\nabla^2 f(i, j) \\ = -f(i+1, j) - f(i-1, j) - f(i, j+1) - f(i, j-1) + 4f(i, j) \quad (3.27)$$

对角线上的像素加入后得到拉普拉斯边缘算子的另外一种表达方式，表示为

$$\nabla^2 f(i, j) = [f(i+1, j-1) + f(i+1, j+1) + f(i-1, j+1) + f(i-1, j-1) \\ + f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1)] - 8f(i, j) \quad (3.28)$$

$\{L(i, j)\}$ 称作边缘图像，拉普拉斯边缘算子模板及扩展模板如图 3.9 所示。

0	-1	0
-1	4	-1
0	-1	0

1	1	1
1	-8	1
1	1	1

图 3.9 拉普拉斯边缘算子模板及扩展模板

5. Kirsch 边缘算子

如图 3.10 所示，8 个卷积核组成了 Kirsch 边缘算子。图像中的每个点都用 8 个模板进行

卷积，每个模板对某个特定边缘方向作出最大响应。所有 8 个方向中的最大值作为边缘幅度图像的输出。最大响应模板的序号构成了对边缘方向的编码。

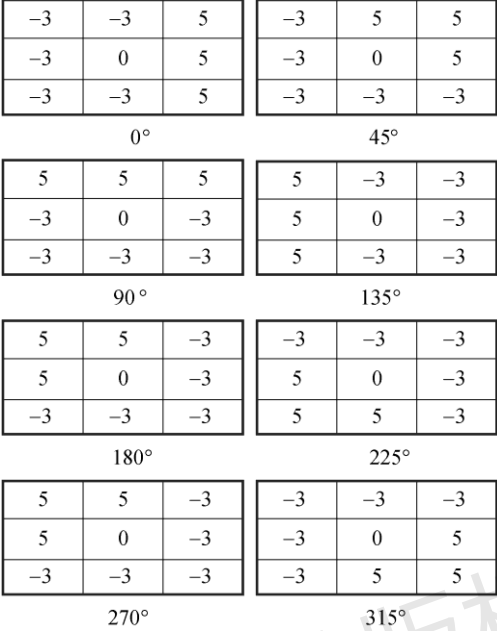


图 3.10 Kirsch 边缘算子模板

6. Marr-Hildreth 边缘算子

Marr-Hildreth边缘检测算子是将高斯算子和拉普拉斯算子结合在一起而形成的一种新的边缘检测算子，先用高斯算子对图像进行平滑处理，然后采用拉普拉斯算子根据二阶微分过零点来检测图像边缘，因此该算子也称为LOG（Laplacian of Gaussian）算子。

在数字图像中实现图像与模板卷积运算时，运算速度与选取的模板大小有直接关系。模板越大，检测效果越明显，速度越慢，反之则效果差一点，但速度提高很多。因此，在不同的条件下应选取不同大小的模板。在实际计算过程中，还可以通过分解的方法提高运算速度，即把二维滤波器分解为独立的行、列滤波器。常用的 5×5 模板的Marr-Hildreth边缘算子如图3.11所示。这是一个 5×5 的矩阵表示，在数字图像处理中模板有两种表示方法，一种是用填有数字的方格表示，一种是用矩阵表示，它们在算法上是一致的。

图3.12所示为LOG算子中心点的像素与其邻域像素的距离和位置加权系数的关系。将图3.12绕 y 轴旋转一周后，LOG算子很像一顶墨西哥草帽，所以，LOG又叫墨西哥草帽滤波器。

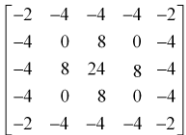


图3.11 LOG算子的 5×5 模板

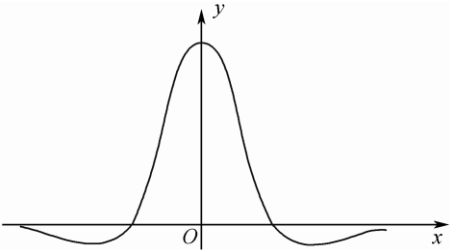


图3.12 LOG算子中心点的距离与位置加权系数的关系

7. Canny 边缘算子

Canny边缘算子是近年来在数字图像处理中广泛应用的边缘算子，它是应用变分原理推导

出的一种用高斯模板导数逼近的最优算子。通过Canny算子的应用，可以计算出数字图像的边缘强度和边缘梯度方向，为后续边缘点的判断提供依据。

Canny算子用范函求导方法推导出高斯函数的一阶导数，即为最优边缘检测算子的最佳近似。由于卷积运算可交换、可结合，故Canny算法首先采用二维高斯函数对图像进行平滑，二维高斯函数表示为

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (3.29)$$

式中， σ 为高斯滤波器参数，它控制着平滑的程度， σ 较小的滤波器定位精度高，但信噪比低； σ 较大的滤波器情况正好相反。因此，要根据需要选取高斯滤波器参数。

传统Canny算法利用一阶微分算子来计算平滑后图像各点处的梯度幅值和梯度方向，获得相应的梯度幅值图像 C 和梯度方向图像 θ 。点 (i, j) 处两个方向的偏导数 $C_x(i, j)$ 和 $C_y(i, j)$ 分别为

$$C_x(x, y) = (f(i, j+1) - f(i, j) + f(i+1, j+1) - f(i+1, j))/2 \quad (3.30)$$

$$C_y(i, j) = (f(i, j) - f(i+1, j) + f(i+1, j+1))/2 \quad (3.31)$$

则此时点 (i, j) 处的梯度幅值和梯度方向分别表示为

$$C(i, j) = \sqrt{C_x^2(i, j) + C_y^2(i, j)} \quad (3.32)$$

$$\theta(i, j) = \arctan \frac{C_x(i, j)}{C_y(i, j)} \quad (3.33)$$

为了精确定位边缘，必须细化梯度幅值图像 C 中的屋脊带，只保留幅值的局部极大值，即非极大值抑制（NMS）。Canny算法在梯度幅值图像 C 中以点 (i, j) 为中心 3×3 的邻域内沿梯度方向 $\theta(i, j)$ 进行插值，若点 (i, j) 处的梯度幅值 $C(i, j)$ 大于 $\theta(i, j)$ 方向上与其相邻的两个插值，则将点 (i, j) 标记为候选边缘点，反之则标记为非边缘点。这样，就得到了候选的边缘图像 E 。

传统Canny算法采用双阈值法从候选边缘点中检测和连接出最终的边缘。双阈值法首先选取高阈值 T_H 和低阈值 T_L ，然后开始扫描图像。对候选边缘图像 E 中标记为候选边缘点的任一像素点 (i, j) 进行检测，若点 (i, j) 梯度幅值 $C(i, j)$ 高于高阈值 T_H ，则认为该点一定是边缘点，若点 (i, j) 梯度幅值 $C(i, j)$ 低于低阈值 T_L ，则认为该点一定不是边缘点。而对于梯度幅值处于两个阈值之间的像素点，则将其看作疑似边缘点，再进一步依据边缘的连通性对其进行判断，若该像素点的邻接像素中有边缘点，则认为该点也为边缘点，否则，认为该点为非边缘点。

在下面描述的3个标准意义下，Canny边缘检测算子对受白噪声影响的阶跃状边缘检测是最优的。

- ① 检测标准——不丢失重要的边缘，不应有虚假的边缘；
- ② 定位标准——实际边缘与检测到的边缘位置之间的偏差最小；
- ③ 单响应标准——将多个响应降低为单个边缘响应。

Canny边缘检测算子是基于如下的几个概念提出的。

- ① 边缘检测算子是针对一维信号表达的，对检测标准和定位标准最优。
- ② 如果考虑第3个标准（多个响应），则需要通过数值优化的办法得到最优解。该最优Canny算子可以有效地近似为标准差为 σ 的高斯平滑滤波器的一阶微分，因此可以实现边缘检测误差小于20%，这与LOG边缘检测算子很相似。

- ③ 将边缘检测算子推广到二维情况，阶跃状边缘由位置、方向和可能的幅度来确定。

8. Prewitt 边缘算子

Prewitt 提出了类似的计算偏微分估计值的方法，梯度计算算法表示为

$$p_x = \{f(i+1, j-1) + f(i+1, j) + f(i+1, j+1)\} - \{f(i-1, j-1) + f(i-1, j) + f(i-1, j+1)\} \quad (3.34)$$

$$p_y = \{f(i-1, j+1) + f(i, j+1) + f(i+1, j+1)\} - \{f(i-1, j-1) + f(i, j-1) + f(i+1, j-1)\} \quad (3.35)$$

它的卷积算子模板如图 3.13 所示。

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

图 3.13 Prewitt 边缘检测算子方向的模板

当用两个模板（卷积核）组成边缘检测器时，通常取较大的幅度作为输出值，这使得它们对边缘的走向有些敏感。取它们的平方和之后的开平方，即可以获得性能更一致的全方位的响应，这与真实的梯度值更接近。另一种方法是将被 Prewitt 算子扩展到 8 个方向，即边缘样板算子。这些算子样板由理想的边缘子图像构成。依次用边缘样板去检测图像，与被检测区域最为相似的样板给出最大值。用这个最大值作为算子的输出值 $p(i, j)$ ，这样可将边缘像素检测出来。采用 Prewitt 算子不仅能检测边缘点，而且能抑制噪声的影响。定义 Prewitt 边缘检测算子模板如图 3.14 所示。8 个算子样板对应的边缘方向如图 3.15 所示。

$$\begin{array}{cccc} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix} \\ \text{1 方向} & \text{2 方向} & \text{3 方向} & \text{4 方向} \\ \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \\ \text{5 方向} & \text{6 方向} & \text{7 方向} & \text{8 方向} \end{array}$$

图 3.14 Prewitt 边缘检测算子模板

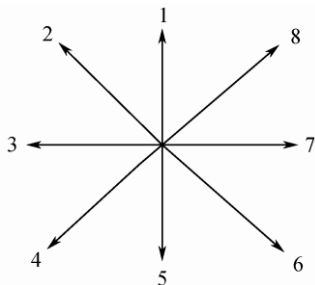


图 3.15 8 个算子样板对应的边缘方向

3.2.4 边缘检测算子的对比

在数字图像处理中，对边缘检测算法的主要要求就是运算速度快，边缘定位准确，噪声抑

制能力强，因此就这几方面对以上介绍的几种算子进行分析比较。首先，在运算速度方面，对于一个 $N \times N$ 的图像，其计算量如表3.1所示。从表3.1可以看出，Kirsch算子、Marr-Hildreth算子和Canny算子的运算量都比较大。

表3.1 各种算子计算量对比

算子名称	加法运算	乘法运算
Roberts 算子	$3 \times N^2$	0
Sobel 算子	$11 \times N^2$	$2 \times N^2$
Prewitt 算子	$11 \times N^2$	0
Kirsch 算子	$56 \times N^2$	$16 \times N^2$
Laplacian 算子	$4 \times N^2$	N^2
Marr-Hildreth 算子	运算量大，具体看所取卷积模块的大小	
Canny 算子	运算量大，具体看所取卷积模块的大小	

其次，在边缘检测中边缘定位能力和噪声抑制能力是一对矛盾，就各种算法而言，有的边缘定位能力比较强，有的抗噪声能力比较好；而对某些算子，参数的选择也直接影响到边缘定位能力和噪声抑制能力的强弱。下面根据实际测试结果，简单介绍各个算子的特点。

1. Roberts 算子

Roberts边缘算子利用局部差分算子寻找边缘，边缘定位精度较高，但容易丢失一部分边缘信息，同时由于没有经过图像平滑处理，所以不能抑制噪声。该算子对具有边缘陡峭的低噪声图像响应最好。

2. Sobel 算子和 Prewitt 算子

Sobel算子和Prewitt算子都是对图像进行差分和滤波运算，差别只是平滑部分的权值有些差异，因此对噪声具有一定的抑制能力，但不能完全排除检测结果中出现伪边缘；同时这两个算子边缘定位比较准确和完整。这两类算子对灰度渐变和具有噪声的图像处理效果较好。

3. Kirsch 边缘算子

Kirsch算子对8个方向边缘信息进行检测，因此具有较好的边缘定位能力，并且对噪声有一定的抑制作用。就边缘定位能力和抗噪声能力而言，该算子的处理效果比较理想。

4. Laplacian 边缘算子

拉普拉斯边缘算子为二阶微分算子，对图像中的阶跃状边缘点定位准确且具有旋转不变性，即无方向性。但是该算子容易丢失一部分边缘的方向信息，造成一些不连续的检测边缘，同时抗噪声能力比较差。拉普拉斯算子比较适用于屋顶状边缘的检测。

5. Marr-Hildreth 边缘算子

Marr-Hildreth算子首先通过高斯函数对图像作平滑处理，因此对噪声的抑制作用比较明显，但同时也可能将原有的边缘也平滑了，造成某些边缘无法检测到。此外，高斯函数中方差参数 σ 的选择，对图像边缘检测效果有很大的影响。 σ 越大，检测到的图像细节越丰富，但对噪声抑制能力相对下降，易出现伪边缘；反之，则抗噪声性能提高，但边缘定位准确性下降，易丢失许多真边缘，因此，对于不同的图像应该选择不同的参数。

6. Canny 边缘算子

Canny边缘算子同样采用高斯函数对图像进行平滑处理，因此具有较强的去噪能力，但同样容易平滑掉一些边缘信息。同时，其后所采用的一阶微分算子的方向性较Marr-Hildreth算子要好，因此边缘定位准确性较高。通过实验结果可以看出，该算子是传统边缘检测算子中效果较好的算子之一。

3.3 霍夫 (Hough) 变换

在数字图像处理中, 霍夫 (Hough) 变换属于特征提取技术, 它由 Paul Hough 于 1962 年提出, 最初只是用于二值图像直线检测, 后来扩展到任意形状的检测。现在常用的变换技术称作广义霍夫变换, 1981 年被 Dana H. Ballard 扩展后应用到计算机视觉领域。

在实际应用中, 图像像素由于噪声、不均匀照明引起的边缘断裂和杂散的亮度不连续而难以得到完全的边缘特征。因此, 典型的边缘检测算法遵循用链接过程把像素组装成有意义的边缘的方法。一种寻找并链接图像中线段的处理方式就是霍夫变换。

1. 霍夫变换原理与性质

传统霍夫变换主要用于检测二值图像中的直线或者曲线, 具体方法是把二值图像变换到霍夫参数计算空间 (HPCS), 并在该空间的特定位置上出现峰值, 因此检测曲线可以简化为找出峰值位置的问题。霍夫变换的主要优点是: 检测出曲线的能力受曲线中断点等干扰的影响较少, 因而是一种快速的形状检测方法。以直线检测为例, 平面中任意一条直线可以用两个参数 ρ 和 θ 完全确定下来, 其中 ρ 指明了该直线到原点的距离, θ 确定了该直线的方位, 其函数关系可表示为

$$f((\rho, \theta), (x, y)) = \rho - x \cos \theta - y \sin \theta = 0 \quad (3.36)$$

给定一幅图像中的一个点集合, 假设想要找到位于直线上的所有点的子集。一种可能的解决方法就是先找到所有由每一对点所决定的线, 然后找到接近特殊线的所有点的子集。这种处理方法的问题是需找 $n(n-1)/2 \sim n^3$ 条线, 然后进行 $n(n(n-1)/2 \sim n^3)$ 次每一点与所有线的比较。这种方法计算起来很麻烦, 因此一般不予采用。

另外, 采用霍夫变换时, 考虑一个点 (x_i, y_i) 和所有通过该点的线。通过点 (x_i, y_i) 的线有无数条, 这些线对某些常数 a 和 b 来说, 均满足 $y_i = ax_i + b$ 。将其改写为 $b = -ax_i + y_i$, 并考虑 ab 平面 (也称为参数空间), 可对一个固定点 (x_i, y_i) 产生单独的一条直线。此外, 第二个点 (x_j, y_j) 也有这样一条在参数空间上与之相关的直线, 这条直线和与 (x_i, y_i) 相关的直线相交于点 (a, b) , 其中 a 和 b 分别是 xy 平面上包含点 (x_i, y_i) 和 (x_j, y_j) 的直线的斜率和截距。如图 3.16(a) 和 (b) 所示说明了上述这些概念。

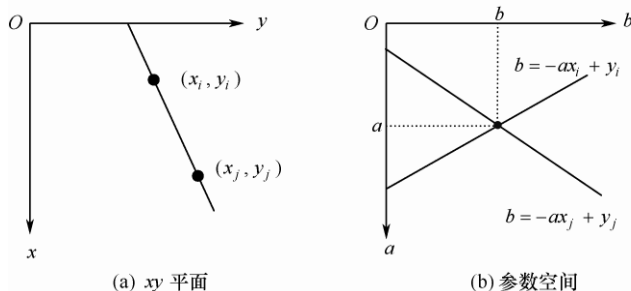


图 3.16 霍夫变换中 xy 平面和参数空间示意图

从原理上讲, 可以绘制出与所有图像点 (x_i, y_i) 相对应的参数空间直线, 而图像直线可以通过许多参数空间直线相交来识别。然而, 这种方法的实际困难是直线的斜率 a 接近无限大, 也就是接近垂直方向。解决这个问题的一种方法是使用直线的标准表示法, 用参数方程表示为

$$x \cos \theta + y \sin \theta = \rho \quad (3.37)$$

如图 3.17(a)所示说明了参数 ρ 和 θ 的几何解释。相对于水平线, $\theta = 0^\circ$, ρ 等于正的 x 截距。同样, 相对于垂直线而言, $\theta = 90^\circ$, ρ 等于正的 y 截距, 或 $\theta = -90^\circ$, ρ 等于负的 y 截距。如图 3.17(b)所示, 其中的每一条正弦曲线表示通过特定点 (x_i, y_i) 的一簇直线。 $\rho\theta$ 平面中的交点 (ρ', θ') , 对应于 xy 平面中通过点 (x_i, y_i) 和 (x_j, y_j) 的直线。图像空间 xy 平面中位于同一条直线上的点 (x_i, y_i) 经过霍夫变换后, 在参数空间中相交于同一点 (ρ', θ') 。在参数空间中确定该点的位置比在图像中确定多个点容易些, 确定了该点的位置, 如图 3.17(b)所示, 就知道了图像空间中直线的参数。

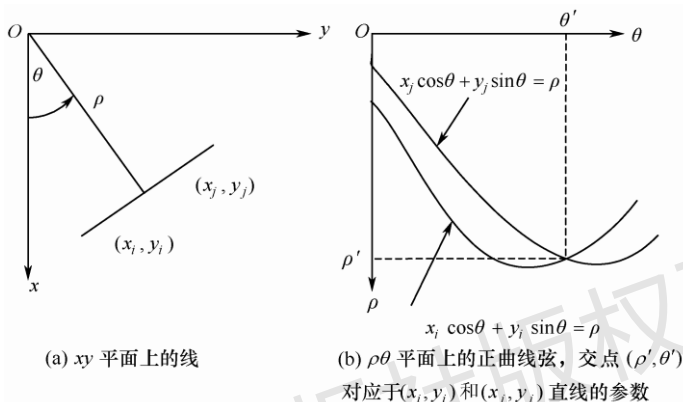


图 3.17 xy 平面上直线映射到 $\rho\theta$ 平面

霍夫变换具有如下性质:

- ① 通过 xy 平面上一点的一簇直线变换到 $\rho\theta$ 平时, 将形成一条正弦状曲线;
- ② $\rho\theta$ 平面中的一点对应于 xy 平面中的一条直线;
- ③ xy 平面中一条直线上的 n 个点对应于 $\rho\theta$ 平面中经过一个公共点的 n 条曲线;
- ④ $\rho\theta$ 平面中一条曲线上的 n 个点对应于 xy 平面中过一公共点的 n 条直线。

2. 霍夫变换应用

霍夫变换计算时把 $\rho\theta$ 参数空间细分为所谓的累加器单元, 如图 3.18 所示, 其中 $(\rho_{\min}, \rho_{\max})$ 和 $(\theta_{\min}, \theta_{\max})$ 是参数值的期望范围。一般来说, θ 和 ρ 的最大范围为 $-90^\circ \leq \theta \leq 90^\circ$ 和 $-D \leq \rho \leq D$, 其中 D 是图像中交点间的距离。坐标 (i, j) 处的单元 (累加器的值为 $A(i, j)$) 对应于与参数空间坐标 (ρ_i, θ_j) 相关的方形。最初, 这些单元被设置为零。然后, 对于图像平面上的每一个非背景点 (x_k, y_k) , 令 θ 等于 θ 轴上允许的细分值, 并通过式 $\rho = x_k \cos \theta + y_k \sin \theta$ 求出相应的 ρ 值。然后, 将得到的 ρ 值四舍五入为最接近的 ρ 轴上的允许单元值。相应的累加器单元加 1。在这个过程的最后, $A(i, j)$ 意味着 xy 平面上的 n 个点位于线 $x \cos \theta_j + y \sin \theta_j = \rho_i$ 上。 $\rho\theta$ 平面上的细分分数决定了这些点共线的精度。

通常, 将 xy 平面称为图像平面, $\rho\theta$ 平面称为参数平面, 提取直线的霍夫变换步骤如下:

- ① 在 ρ 和 θ 的最大值和最小值之间建立一个离散参数空间;

- ② 建立一个累加器 $A(\rho, \theta)$, 并置每个元素的初值为 0;

- ③ 在 $\rho\theta$ 参数平面, 对每一点做霍夫变换, 计算出网格上所有点的 (ρ, θ) 值, 并在相应的累加器加 1, 即 $A(\rho, \theta) = A(\rho, \theta) + 1$;

④ 完成累加后，统计 $A(\rho, \theta)$ 的值就得到 (ρ, θ) 处共线点的个数和共线的参数，由此可以得出点所在的线。

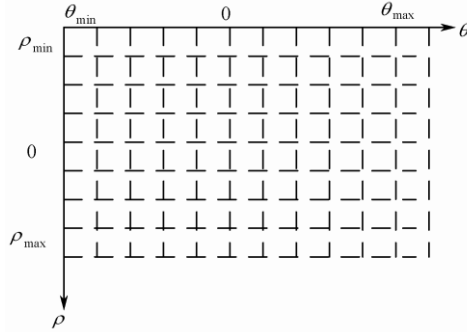


图 3.18 把 $\rho\theta$ 平面划分为累加器单元

综上所述，霍夫变换算法的输入为包含有黑条纹白背景的二值图，输出为 HPCS 空间的累加器矩阵的值。基本原理是将图像空间中直线检测问题转换到参数空间中对点的检测问题，通过在参数空间中进行简单的累加统计完成检测任务的方法。而且在参数空间中，噪声点或孤立点是随机的，得不到有效的积累，因此其数值很小。而图像空间的直线对参数空间的点的积累贡献很大，因此提高了信噪比，达到了较好的检测效果。

3.4 纹理分析

图像的纹理分析已经在许多领域得到了广泛的应用。纹理一般指人们所观察到的图像中像素的灰度变化规律，通常将局部不规则而在宏观上有规律的特性称为纹理。图像的纹理特征描述了在图像中反复出现的局部模式及其排列规则，反映了宏观意义上灰度变化的一些规律。图像可以看成是不同纹理区域的组合，纹理通常定义为图像的某种局部性质，或是对局部区域像素之间关系的一种度量，可用于定量描述图像中的空间信息。

把图像分成不同的部分，各部分内部具有相近纹理的过程称为纹理分割。由于不同物体往往具有不同的纹理特征，纹理分割将图像划分为互不相交的若干区域，每一个区域内部具有相对一致的纹理特征。两种常见的纹理分割方法是基于区域的方法和基于边界的方法，用于识别具有相同纹理的区域和检测相邻纹理的突变点。纹理分割包括有监督和无监督的纹理分割。有监督纹理分割是指在对待分割图像掌握一定先验知识的情况下进行纹理分割，反之称之为无监督分割。纹理的先验知识包括确定待分割图像的纹理类别数目、不同纹理的表现特性等。

3.4.1 基于邻域特征统计的方法

将图像中某一局部区域内灰度的统计特征计算出来作为图像纹理特征，主要方法概括总结如下。

1. 最大最小值法

设原图像为 $\{f(x, y)\}$ ，图像的大小为 $M \times N$ ，最大最小值法是以 (i, j) 点像素为中心的窗口 $(2k+1) \times (2k+1)$ 内的灰度最大值与最小值的差值作为窗口中心的纹理统计值，表示为

$$I(i, j) = \max_{\substack{|x-i| \leq k \\ |y-j| \leq k}} \{f(x, y)\} - \min_{\substack{|x-i| \leq k \\ |y-j| \leq k}} \{f(x, y)\} \quad (3.38)$$

式中，窗口大小决定了被检测的纹理尺寸， $I(i, j)$ 表示纹理的强度。

2. 方差法

方差法是以 (i, j) 点像素为中心的窗口 $(2k+1) \times (2k+1)$ 内的灰度方差作为窗口中心的纹理特征统计值，表示为

$$\sigma^2(i, j) = \sum_{x=i-k}^{i+k} \sum_{y=j-k}^{j+k} [f(x, y) - \mu]^2 \quad (3.39)$$

式中， μ 为窗口内灰度平均值，灰度方差反映了纹理强度信息。

3. 绝对值法

以 (i, j) 点像素为中心的窗口 $(2k+1) \times (2k+1)$ 内的每个像素灰度值与窗口内的灰度平均值的绝对值的和作为窗口中心的纹理特征统计值，表示为

$$A(i, j) = \sum_{x=i-k}^{i+k} \sum_{y=j-k}^{j+k} |f(x, y) - \mu| \quad (3.40)$$

式中， μ 为窗口内灰度平均值， $A(i, j)$ 反映了窗口内纹理的强度信息。

4. 高斯滤波差值法

高斯滤波差值法表示为

$$W(i, j) = 1 - \sigma \exp \left[-\frac{(i-i_0)^2 + (j-j_0)^2}{2\sigma^2} \right] \quad (3.41)$$

式中， (i_0, j_0) 为高斯模板中心坐标， σ 为标准差。算法步骤如下：

- ① 采用两个不同的标准差 σ_1 、 σ_2 ，对原图像进行滤波；
- ② 将两个滤波输出的差值图像作为纹理统计值。

这种方法适用于尺度较小的纹理。

5. 信息熵法

以 (i, j) 点像素为中心的窗口 $(2k+1) \times (2k+1)$ 内的每个像素的灰度值熵的和作为窗口中心的纹理特征统计值，表示为

$$B(i, j) = - \sum_{x=i-k}^{i+k} \sum_{y=j-k}^{j+k} f(x, y) / \sum_{x=i-k}^{i+k} \sum_{y=j-k}^{j+k} f(x, y) \quad (3.42)$$

式中， $B(i, j)$ 反映灰度变化速率。

3.4.2 傅里叶频谱方法提取纹理特征

由于傅里叶变换的共轭对称性，即满足 $|F(u, v)| = |F(-u, -v)|$ ，频谱分布以图像原点为中心是对称的。图像的纹理呈现出一定的周期性，或者说它在图像空间中具有一定的发生频率，可以对图像进行频谱分析来提取纹理特征。对于较粗的纹理，它的能量主要集中在低频部分；对于较细的纹理，较大的能量集中在高频部分。

将 u, v 以极坐标表示为

$$r = \sqrt{u^2 + v^2} \quad (3.43)$$

$$\theta = \arctan \frac{v}{u} \quad (3.44)$$

这时的频谱可用函数 $S(r, \theta)$ 表示。对于每个确定的频率 r ， $S(r, \theta)$ 是一个一维函数 $S_r(\theta)$ ；对于每个确定的方向 θ ， $S(r, \theta)$ 是一个一维函数 $S_\theta(r)$ 。

对于每个确定的频率 r ，可以得到以原点为圆心的圆形上的频谱特性，表示为

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \quad (3.45)$$

同样，对于每个确定的方向 θ ，可以得到沿着自原点的辐射方向上的频谱所表现的特性，表示为

$$S(\theta) = \sum_{r=1}^R S_r(\theta) \quad (3.46)$$

式中， R 表示以原点为中心的圆的半径。 $S(r)$ 和 $S(\theta)$ 构成整幅图像或图像区域纹理频谱能量的描述。

3.4.3 灰度共生矩阵

由于纹理是由灰度分布在空间位置上反复出现而形成的，因而在图像空间中相隔某距离的两像素间会存在一定的灰度关系，这种关系被称为图像中灰度空间的相关特性。通过研究灰度的空间相关特性来描述纹理，这正是灰度共生矩阵的思想基础。

从灰度级为 i 的像素点出发，距离为 δ 的另一个像素点同时发生的灰度级为 j ，定义这两个灰度在整个图像中发生的概率分布，称为灰度共生矩阵。如图 3.19 所示，其中 Δx 和 Δy 的范围由像素间距 δ 和方向 θ 两个参数决定，即 $\Delta x = \delta \cos \theta$ ， $\Delta y = \delta \sin \theta$ 。灰度共生矩阵可以表示纹理的稀疏度、对比度、复杂度及其纹理力度等特性。

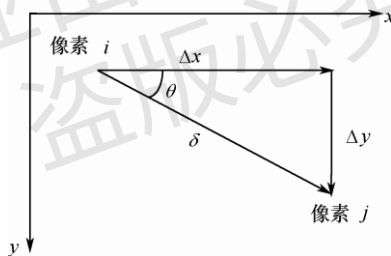


图 3.19 像素对示意图

设 $f(x, y)$ 为一幅灰度图像，对图像中任一区域 R ，定义 S 为区域中具有特定空间联系的像素对集合，则灰度共生矩阵可表示为

$$m_{(d, \theta)}(i, j) = \text{card}\{[(x_1, y_1), (x_2, y_2)] \in S \mid f(x_1, y_1) = i \& f(x_2, y_2) = j\} \quad (3.47)$$

式中， $x_2 = x_1 + d \cos \theta$ ， $y_2 = y_1 + d \sin \theta$ ， $\text{card}(S)$ 表示集合 S 中对 $m_{(d, \theta)}(i, j)$ 有贡献的元素个数。在实际应用中，常需要对式(3.47)进行归一化，表示为

$$m'_{(d, \theta)}(i, j) = \frac{\text{card}\{[(x_1, y_1), (x_2, y_2)] \in S \mid f(x_1, y_1) = i \& f(x_2, y_2) = j\}}{\text{card}(S)} \quad (3.48)$$

为减少计算量，计算中往往需要先对图像进行灰度变换，降低灰度等级，同时减少 θ 的方向数，通常取 0° 、 45° 、 90° 、 135° 这 4 个方向。

假设图像的灰度级有 8 级，则对应于如图 3.20(a)所示的图像区域，其共生矩阵 $m_{(1,90^\circ)}$ 如图 3.20(b)所示。

		0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0	0
1	0	1	2	0	0	0	0	0	0
2	0	1	0	2	0	0	0	0	0
3	0	0	1	1	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0

(a) 图像数据

(b) 灰度共生矩阵

图 3.20 灰度共生矩阵示例

灰度共生矩阵是分析图像局部模式和排列规则的基础，为了有效利用灰度共生矩阵所提供的灰度方向、间隔和变化幅度的信息，在共生矩阵的基础上提取了一些有意义的统计量，基于这些数字统计量，对图像纹理特征进行了定量描述。所提出的纹理特征主要有以下几种。

(1) 角二阶距（能量）

该特征反映了图像区域的均匀性或平滑性，可以表示为

$$W_1 = \sum_i \sum_j \{m(i, j)\}^2 \quad (3.49)$$

在均匀区域，灰度值变化较小，大部分像素对具有相同或相近的取值，主要发生在矩阵的对角线附近，其他大部分元素为零；而在非均匀区，灰度值变化大的像素相对较多，在整个灰度共生矩阵上概率均匀分布，而且矩阵中元素的值都很小，所以，非均匀区的角二阶距比均匀区的角二阶距要小。该测度对区域内有无灰度变化较敏感，但对灰度值变化数值大小不敏感，即具有高的局部灰度值对比度的区域角二阶距值不一定高。当共生矩阵中所有 $m(i, j)$ 都相等时， W_1 达到最小值。

(2) 对比度

又称为非相似性，可理解为图像的清晰度，即纹理清晰度。数学描述为

$$W_2 = \sum_i \sum_j (i - j)^2 m(i, j) \quad (3.50)$$

其中， $|i-j|$ 表示图像特定位置关系下像素对灰度值差，灰度值差值大的像素对越多，这个值就越大。对比度反映了近邻像素的反差，当图像中两个灰度级点对的统计个数接近共生矩阵对角线时，纹理变化较小，对比度较小；反之，则表明近邻像素的反差较大，纹理较细。因此对比度值的大小反映了纹理的粗细度。

(3) 相关性

相关系数在一定程度上反映了矩阵行与列的线性相关程度。相关系数较大时，图像区域灰度分布比较均匀。相关性可表示为

$$W_3 = \frac{\sum_i \sum_j ijm(i, j)^2 - \mu_x \mu_y}{\delta_x \delta_y} \quad (3.51)$$

式中, μ_x , μ_y , δ_x , δ_y 分别定义为

$$\begin{aligned}\mu_x &= \sum_i i \sum_j m(i, j) \\ \mu_y &= \sum_j j \sum_i m(i, j) \\ \delta_x &= \sum_i (j - \mu_x)^2 \sum_j m(i, j) \\ \delta_y &= \sum_j (j - \mu_y)^2 \sum_i m(i, j)\end{aligned}\quad (3.52)$$

(4) 熵

熵给出了图像内容随机性的度量, 是图像具有信息量的度量。表示为

$$W_4 = \sum_i \sum_j m(i, j) \log[m(i, j)] \quad (3.53)$$

纹理是图像信息的一种, 若图像没有任何纹理, 则灰度共生矩阵几乎为零矩阵, 熵值接近于 0; 若图像有较多的细纹理, 则矩阵中元素近似相等, 该图像的熵值最大; 若仅有较少的纹理, 则矩阵中的元素的差别较大, 图像的熵值就较小。

(5) 差分矩

差分矩的定义为

$$W_5 = \sum_i \sum_j (i - \mu)^2 m(i, j) \quad (3.54)$$

式中, μ 为 $m(i, j)$ 的均值。

(6) 逆差分矩

逆差分矩又称均匀性, 可表示为

$$W_6 = \sum_i \sum_j \frac{m(i, j)}{1 + (i - j)^2} \quad (3.55)$$

对于匀质区域, 其灰度共生矩阵的元素集中在对角线上, $(i - j)^2$ 值小, 则均匀性特征值较大; 对于非匀质区域, 由于其灰度共生矩阵的元素集中在远离对角线上, $(i - j)^2$ 值大, 则均匀性的值较小, 所以该特征是图像分布平滑性的测度。

(7) 和平均

和平均表示为

$$W_7 = \sum_k \sum_i \sum_j im(i, j) \quad (3.56)$$

式中, $k = i + j$ 。

(8) 和方差

和方差表示为

$$W_8 = \sum_k \sum_i \sum_j (i - W_7)^2 m(i, j) \quad (3.57)$$

式中, $k = i + j$ 。

(9) 和熵

和熵表示为

$$W_9 = \sum_k \sum_i \sum_j m(i, j) \log_2 m(i, j) \quad (3.58)$$

(10) 差方差

差方差表示为

$$W_{10} = \sum_{k=2}^{2N} \sum_{i=1}^N \sum_{k=1}^N m(i, j) \quad k = |i - j| = 0 \cdots N - 1 \quad (3.59)$$

式中, $d = \sum_{i=1}^N \sum_{j=1}^N |i - j| m(i, j)$ 。

由于上述分量的物理意义和取值范围不同, 因此, 分割时需要对它们进行内部归一化。这样, 在进行相似量度量时, 可使各分量具有相同的权重。

习 题 3

3.1 图像分割有何实际意义? 具体的分割方法有哪些?

3.2 图像的边缘具有哪些特征?

3.3 什么是区域? 什么是图像分割?

3.4 边缘检测的理论依据是什么? 有哪些常用算法? 各种边缘检测算子具有什么特点?

3.5 设图像表示为

$$f(x, y) = \begin{bmatrix} 1 & 5 & 30 & 10 & 10 & 20 & 20 & 20 \\ 1 & 7 & 30 & 15 & 15 & 15 & 10 & 9 \\ 3 & 7 & 10 & 10 & 2 & 6 & 7 & 6 \\ 1 & 0 & 8 & 7 & 2 & 1 & 0 & 0 \\ 1 & 1 & 6 & 50 & 2 & 2 & 2 & 2 \\ 2 & 3 & 9 & 7 & 2 & 0 & 0 & 0 \\ 1 & 5 & 25 & 10 & 20 & 20 & 10 & 50 \\ 50 & 20 & 30 & 2 & 7 & 2 & 0 & 0 \end{bmatrix}$$

试设计一种算法求出进行图像二值化的阈值 T , 并且对图像进行二值分割。

3.6 设计一种利用 Sobel 边缘算子进行图像边缘检测的程序框图。