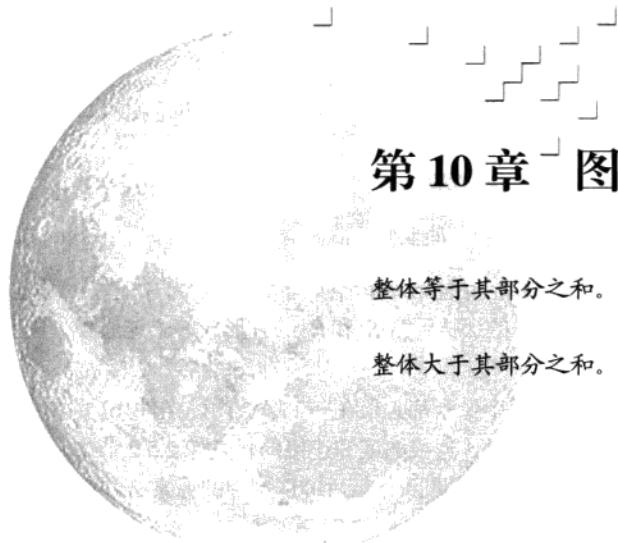




第10章 图像分割

整体等于其部分之和。

——欧几里得

整体大于其部分之和。

——Max Wertheimer

引言

从前一章的资料开始了一个转变，即从输入和输出都是图像的图像处理方法，转到了输入为图像而输出为从这些图像中提取出来的属性的图像处理方法(在1.1节中的定义的意义)。分割是这一方向的另一个主要步骤。

分割将图像细分为构成它的子区域或物体。细分的程度取决于要解决的问题。也就是说，在应用中，当感兴趣的物体或区域已经被检测出来时，就停止分割。例如，在电子元件的自动检测中，我们关注的是分析产品的图像，客观地确定是否存在特定的异常现象，如丢失元件或断裂的连线。超过识别这些元素所需细节的分割是没有意义的。

异常图像的分割是图像处理中最困难的任务之一。分割的精度决定着计算分析过程最终的成败。因此，应该对改进准确分割的可能性给予相当的关注。在某些情形下，比如在工业检测的应用中，至少对环境某些范围的典型控制还是可能的。有经验的图像处理系统设计师总是非常关注这种机会。在其他应用中，比如自主目标获取，系统设计者无法控制操作环境，常用方法是將注意力集中于传感器类型的选择上，在减少图像无细节的影响的同时，很可能增强感兴趣物体。一个很好的例子是军方使用红外线成像来检测有着较强热特性的物体，如运动中的装备和部队。

本章中的多数分割算法均基于灰度值的两个基本性质之一：不连续性和相似性。在第一类中，方法是以灰度突变为基础分割一幅图像，比如图像的边缘。在第二类中，主要方法是根据一组预定义的准则将一幅图像分割为相似的区域。阈值处理、区域生长、区域分裂和区域聚合都是这类方法的例子。在本章中，我们将讨论和说明这些方法，并说明综合运用不同种类的方法可以改善分割的性能，例如，边缘检测就结合了阈值处理的技术。我们还将讨论基于形态学的图像分割。这种方法特别有吸引力，因为它基于本章第一部分给出的技术，结合了分割的一些值得肯定的属性。最后，我们将以分割中运动线索运用的简单讨论结束本章。

基于多个灰度(亮度)值的分割技术的讨论，见6.7节和10.3.8节。

10.1 基础知识

令 R 表示一幅图像占据的整个空间区域。我们可以将图像分割视为把 R 分为 n 个子区域 $R_1, R_2, \dots, R_n$ 的过程，满足：

$$(a) \bigcup_{i=1}^n R_i = R。$$

(b) R_i 是一个连通集, $i=1,2,\dots,n$ 。

(c) $R_i \cap R_j = \emptyset$, 对于所有 i 和 j , $i \neq j$ 。

(d) $Q(R_i) = \text{TRUE}$, $i=1,2,\dots,n$ 。

(e) $Q(R_i \cup R_j) = \text{FALSE}$, 对于任何 R_i 和 R_j 的邻接区域。

关于连通集的讨论, 请参阅 2.5.2 节。

其中, $Q(R_k)$ 是定义在集合 R_k 的点上的一个逻辑属性, 并且 $\emptyset$ 表示空集。如 2.6.4 节中定义的那样, 符号 $\cup$ 和 $\cap$ 分别表示集合的并和交。如 2.5.2 节中讨论的那样, 若 R_i 和 R_j 的并形成是一个连通集, 则我们说这两个区域是邻接的。

条件(a)指出, 分割必须是完全的; 也就是说, 每个像素都必须在一个区域内。条件(b)要求一个区域中的点以某些预定义的方式来连接(即这些点必须是 4 连接的或 8 连接的, 就像 2.5.2 节中定义的那样)。条件(c)指出, 各个区域必须是不相交的。条件(d)涉及分割后的区域中的像素必须满足的属性——例如, 如果 R_i 中的所有像素都有相同的灰度级, 则 $Q(R_i) = \text{TRUE}$ 。最后, 条件(e)指出, 两个邻接区域 R_i 和 R_j 在属性 Q 的意义上必须是不同的^①。

这样, 我们看到, 分割中的基本问题就是把一幅图像分成满足前述条件的多个区域。通常, 针对单色图像的分割算法基于处理灰度值的两类特性之一: 不连续性和相似性。在第一类特性中, 假设这些区域的边界彼此完全不同, 且与背景不同, 从而允许基于灰度的局部不连续性来进行边界检测。基于边缘的分割是这一类中所用的主要方法。第二类中基于区域的分割方法是根据事先定义的一组准则把一幅图像分割成相似的几个区域。图 10.1 说明了前述概念。图 10.1(a) 显示了一幅在恒定灰度的暗背景上叠加一个恒定灰度区域的图像。这两个区域构成了整个图像区域。图 10.1(b) 显示了基于灰度的不连续性来计算内部区域的边界所得到的结果。边界内侧和外侧的点都是黑色(0), 因为在这些区域中灰度不存在不连续性。为了分割该图像, 我们对该边界上或该边界内的像素分配了一个灰度级(譬如白), 而对该边界外部的所有点分配了另一个灰度(譬如黑)。图 10.1(c) 显示了这一处理方法的结果。我们看到, 该结果满足本节开始时说明的条件(a)到(c)。条件(d)的属性是: 如果一个像素位于边界上或边界内, 则将其标为白色; 否则标为黑色。我们看到, 在图 10.1(c) 中, 对于标为黑色和白色的点, 该属性为“真”。类似地, 两个分割后的区域(物体和背景)满足条件(e)。

下面三幅图像说明了基于区域的分割。图 10.1(d) 类似于图 10.1(a), 但内部区域的灰度形成了一幅纹理模式。图 10.1(e) 显示了计算该图像的边缘的结果。很清楚, 灰度中的大量寄生变化使得识别原图像中的唯一边界很困难, 因为很多非零灰度变化连接到了边界上, 故基于边缘的分割不是一种合适的方法。然而, 我们注意到, 外部区域是恒定的, 因此, 我们需要解决的这个简单分割问题的全部就是区分纹理区域和恒定区域的一个属性。像素值的标准差是完成该任务的一种度量, 因为在纹理区域标准差非零, 而在其他区域标准差为零。图 10.1(f) 显示了将原图像分成大小为 4×4 的几个子区域后的结果。若某个子区域中的像素的标准差为正(即如果属性为“真”), 则将该子区域标记为白色; 其他为零。其结果是在区域边缘周围有“块”效应出现, 因为 4×4 的方块组被标记为相同的灰度。最后, 注意这些结果也满足本节开始处说明的 5 个条件。

① 一般来说, Q 可以是一个复合表达式, 例如, 如果 R_i 中像素的平均灰度小于 m_i , 且如果它们的灰度的标准差大于 σ_i , 其中 m_i 和 σ_i 是指定的常数, 则 $Q(R_i) = \text{TRUE}$ 。

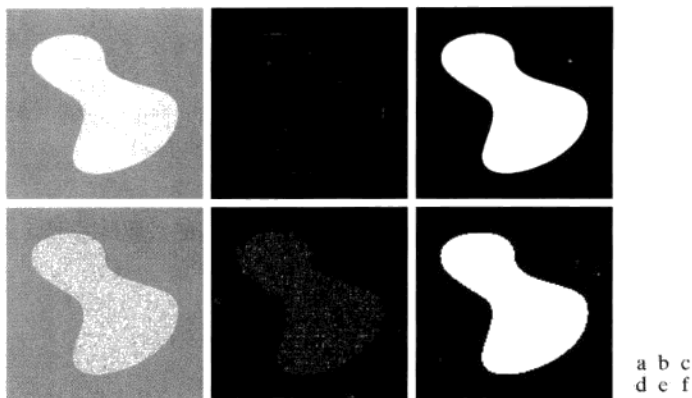


图 10.1 (a) 包含恒定灰度区域的图像; (b) 显示内部区域边界的图像, 该图像是由灰度不连续性获得的; (c) 将图像分割成两个区域后的结果; (d) 包含一个纹理区域的图像; (e) 计算边缘后的结果。注意, 由于存在大量连接到原始边界的小边缘, 仅使用边缘信息是很难找到一条唯一的边界的; (f) 基于区域特性的分割结果

10.2 点、线和边缘检测

本节将集中在以灰度局部剧烈变化检测为基础的分割方法上。我们感兴趣的三种图像特征是孤立点、线和边缘。边缘像素是图像中灰度突变的那些像素, 而边缘(或边缘线段)是连接的边缘像素的集合(关于连接性的定义, 见 2.5.2 节)。边缘检测器是设计用来检测边缘像素的局部图像处理方法。一条线可视为一条边缘线段, 该线两侧的背景灰度要么远亮于该线像素的灰度, 要么远暗于该线像素的灰度。事实上, 如下节及 10.2.4 节中讨论的那样, 线会引起所谓的“屋顶边缘”。类似地, 孤立点可视为一条线, 只是其长度和宽度都为一个像素。

当我们说到线时, 实际上是指那些较细的结构, 它通常只有几个像素粗, 譬如数字化后的建筑设计图中的线, 或卫星图像中的道路。

10.2.1 背景知识

正如在 2.6.3 节和 3.5.1 节我们所看到的局部平均平滑一幅图像那样, 假设平均处理类似于积分, 对于灰度的突变, 局部变化可以用微分来检测应该并不奇怪。由于变化非常短促, 因此一阶微分和二阶微分特别适合于这种目的。

数字函数的导数可用差分来定义。就像 3.6.1 节中解释的那样, 近似这些差分的方法有多种, 但对于用于一阶导数的任何近似, 我们要求: (1) 在恒定灰度区域必须为零; (2) 在灰度台阶或斜坡开始处必须不为零; (3) 在沿灰度斜坡点处也必须不为零。类似地, 对于所用的二阶导数的近似, 我们要求: (1) 在恒定灰度区域必须为零; (2) 在灰度台阶或斜坡的开始处和结束处必须不为零; (3) 沿灰度斜坡必须为零。因为我们处理的是数字量, 故它们的值是有限的, 最大可能的灰度变化也是有限的, 且在其上发生变化的最短距离是在两个邻接像素间的距离。

回忆 2.4.2 节可知, 为表达清晰, 图像样本间的增量定义为 1, 因此在式(10.2-1)的推导中使用了 $\Delta x = 1$ 。

我们按如下方式来得到一维函数 $f(x)$ 在点 x 处的导数的近似: 将函数 $f(x + \Delta x)$ 展开为关于 x 的泰勒级数, 令 $\Delta x = 1$, 且只保留该级数的线性项(见习题 10.1)。结果是数字差分

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x) \quad (10.2-1)$$

当我们考虑一个两变量的图像函数 $f(x, y)$ 时, 为了表示的一致性, 这里我们使用了偏微分, 此时, 我们将处理沿两个空间轴的偏微分。很明显, 当函数 f 只有一个变量时, 有 $\partial f / \partial x = df / dx$ 。

对式(10.2-1)关于 x 微分, 我们得到一个二阶导数表达式:

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{\partial f'(x)}{\partial x} = f'(x+1) - f'(x) \\ &= f(x+2) - f(x+1) - f(x+1) + f(x) \\ &= f(x+2) - 2f(x+1) + f(x)\end{aligned}$$

其中, 第二行来自式(10.2-1)。这一展开是关于点 $x+1$ 的。我们的兴趣是关于点 x 的二阶导数, 故将上式中的变量减 1 后, 得到

$$\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x+1) + f(x-1) - 2f(x) \tag{10.2-2}$$

很容易证明式(10.2-1)和式(10.2-2)满足本节开始时说明的关于一阶和二阶导数的条件。为了说明这一点, 并强调一阶导数和二阶导数在图像处理中的基本相同点和不同点, 考虑图 10.2。

图 10.2(a) 显示了一幅图像, 该图像中包含有不同的实心物体、一条线和单个噪声点。图 10.2(b) 显示了近似通过该图像中心的水平灰度剖面(扫描线), 其包括孤立点。实心物体和沿扫描线的背景之间的灰度过渡显示了两种类型的边缘: 斜坡边缘(左侧)和台阶边缘(右侧)。正如稍后我们要讨论的那样, 灰度过渡涉及较细物体, 如常称为屋顶边缘的线。图 10.2(c) 显示了剖面线的简化, 构成曲线的点刚好足够我们在遇到一个噪声点、一条线和物体的边缘时在数量上分析一阶导数和二阶导数的特性。在这幅简化图中, 斜坡的过渡跨越 4 个像素, 噪声点是单个像素, 线为 3 个像素粗, 灰度台阶的过渡发生在邻接像素间。为简化起见, 灰度级数限制为 8 级。

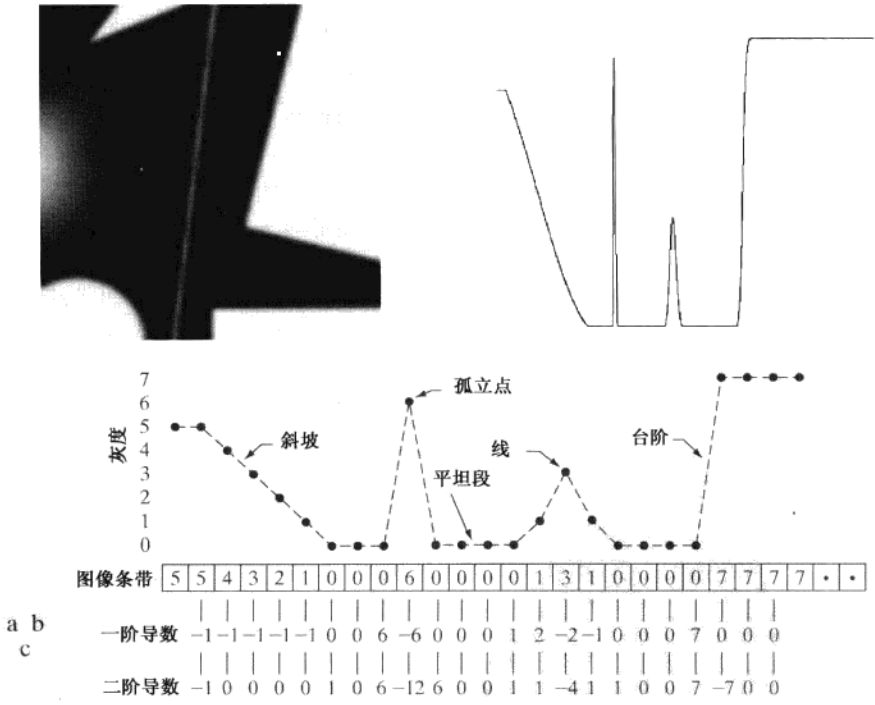


图 10.2 (a) 图像; (b) 通过该图像中心的水平灰度剖面, 包括孤立噪声点; (c) 简化后的剖面线(为清楚起见, 点已用虚线连接起来)。图像条带对应于灰度剖面曲线, 方框中的数字是剖面线中所示的点的灰度值。导数是使用式(10.2-1)和式(10.2-2)得到的

现在我们考虑当从左到右横穿该剖面时一阶导数和二阶导数的性质。首先，我们注意到在灰度斜坡的开始处并沿着整个灰度斜坡，一阶导数不为零，同时二阶导数仅在斜坡的开始处和结尾处不为零。因为数字图像的边缘类似于这种类型的过渡，故我们断定一阶导数会产生“粗”边缘，而二阶导数则会产生细得多的边缘。接下来，我们遇到孤立噪声点。该点处的二阶导数响应幅度远强于一阶导数响应幅度。这并不意外，因为在增强剧烈变化方面，二阶导数要比一阶导数更为激进。这样，我们就可以预期，在增强细节(包括噪声)方面二阶导数远强于一阶导数。在本例中线是如此之细，以至于它也是更精细的细节，并且我们再次看到二阶导数的幅度更大。最后，我们注意到在斜坡和台阶边缘中，进入边缘和离开边缘过渡时的二阶导数的符号相反(从负到正或从正到负)。正如我们在 10.2.6 节中显示的那样，这种“双边缘效应”是可用于定位边缘的一种重要特性。二阶导数的符号也可用于确定一个边缘是从亮到暗(负二阶导数)过渡还是从暗到亮(正二阶导数)过渡，这里，当我们移进边缘时，就观察到这个符号。

总之，我们得出如下结论：(1)一阶导数通常在图像中产生较粗的边缘；(2)二阶导数对精细细节，如细线、孤立点和噪声有较强的响应；(3)二阶导数在灰度斜坡和灰度台阶过渡处会产生双边缘响应；(4)二阶导数的符号可用于确定边缘的过渡是从亮到暗还是从暗到亮。

用于计算图像中每个像素位置处的一阶导数和二阶导数的可选择方法是使用空间滤波器。对于图 10.3 中的 3×3 滤波器模板，该过程是计算模板系数与被该模板覆盖的区域中的灰度值的乘积之和。也就是说，参考式(3.4-3)，模板在该区域中心点处的响应是

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{k=1}^9 w_k z_k \quad (10.2-3)$$

其中 z_k 是像素的灰度，该像素的空间位置对应于模板中第 k 个系数的位置。3.4 节和 3.6 节中已详细讨论了对图像中所有像素执行这种操作的细节。换句话说，基于空间模板的导数的计算是用这些模板对一幅图像进行空间滤波，正如在那些章节描述的那样^①。

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

图 10.3 一个普通的 3×3 空间滤波器掩模

10.2.2 孤立点的检测

基于前一节得到的结论，我们可知点的检测应以二阶导数为基础。由 3.6.2 节的讨论，这意味着使用拉普拉斯：

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (10.2-4)$$

其中，偏微分用式(10.2-2)得到：

$$\frac{\partial^2 f(x, y)}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad (10.2-5)$$

和

$$\frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \quad (10.2-6)$$

① 像 3.4.3 节中解释的那样，式(10.2-3)要么是由式(3.4-1)给出的空间相关的简化表示，要么是由式(3.4-2)给出的空间卷积的简化表示。因此，当在一幅图像中的所有位置处计算 R 时，结果都是一个阵列。本章中的所有空间滤波操作都是使用相关完成的。在某些情形下，我们通常会使用术语“模板与图像卷积”。但是，我们仅在滤波器是对称滤波器时才使用这一术语，因为这一情形下相关和卷积会产生相同的结果。

故拉普拉斯是

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \tag{10.2-7}$$

如 3.6.2 节中解释的那样，该表达式可以用图 3.37(a) 中的模板来实现。此外，如该节描述的那样，我们可以把式(10.2-7)扩展为包括对角项，并使用图 3.37(b) 中的模板。使用图 10.4(a) 中的拉普拉斯模板，它与图 3.37(b) 中的模板相同，如果在某点处该模板的响应的绝对值超过了一个指定的阈值，那么我们就在模板中心位置(x, y)处的该点已被检测到了。在输出图像中，这样的点被标注为 1，而所有其他点则被标注为 0，从而产生一幅二值图像。换句话说，输出是使用如下表达式得到的：

$$g(x, y) = \begin{cases} 1, & |R(x, y)| \geq T \\ 0, & \text{其他} \end{cases} \tag{10.2-8}$$

其中，g 是输出图像，T 是一个非负的阈值，R 由式(10.2-3) 给出。该式简单地度量一个像素及其 8 个相邻像素间的加权差。从直观上看，这一概念是一个孤立点的灰度将完全不同于其周围像素的灰度，因而，使用这种类型的模板可很容易地检测出这个孤立点。考虑的重点仅仅是灰度的不同对于研究孤立点已很充分了。注意，通常对于一个导数模板，这些系数之和为零表明在恒定灰度区域模板响应将是 0。

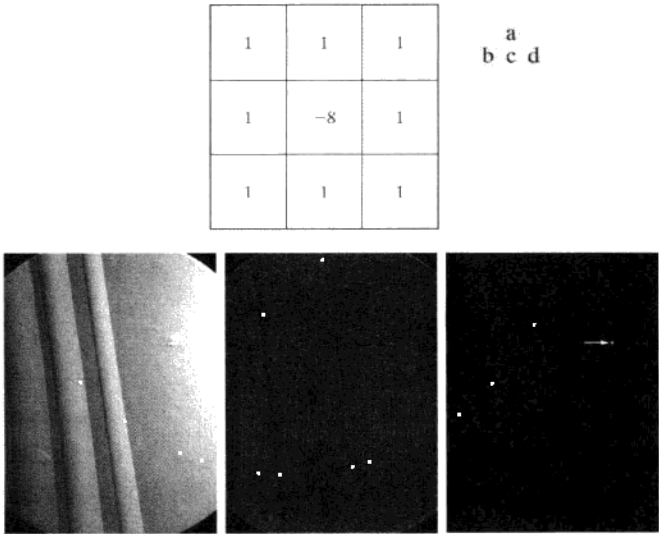


图 10.4 (a) 点检测(拉普拉斯) 模板；(b) 带有一个通孔的涡轮叶片的 X 射线图像。该通孔含有一个黑色像素；(c) 模板与图像卷积的结果；(d) 使用式(10.2-8) 得到的结果，结果中显示了一个点(为便于观看，该点已被放大)(原图像由 X-TEK Systems 有限公司提供)

例 10.1 图像中孤立点的检测。

借助于图 10.4(b)，我们来说明如何从一幅图像中将孤立点分割出来。该图像是一幅喷气发动机涡轮叶片的 X 射线图像。图像右上象限中的叶片有一个通孔，该通孔中已嵌入一个黑色像素。图 10.4(c) 是将点检测模板应用到该幅 X 射线图像后的结果，图 10.4(d) 显示了当 T 取图 10.4(c) 中像素的最高绝对值的 90% 时，应用式(10.2-8) 所得到的结果。在这幅图像中，这个单一像素清晰可见(为增加其可视性，该像素已被人为放大)。这种类型的检测过程相当特殊，因为它基于单个像素位置处灰度的突变，这些位置被检测模板区域中的同质背景所围绕。当这一条件不能满足时，则本章中讨论的其他方法会更适合于检测灰度变化。

10.2.3 线检测

复杂度更高的检测是线检测。基于10.2.1节中的讨论,我们知道,对于线检测,可以预期二阶导数将导致更强的响应,并产生比一阶导数更细的线。这样,对于线检测,我们也可以使用图10.4(a)中的拉普拉斯模板,记住,二阶导数的双线效应必须做适当的处理。下面的例子说明了这一处理过程。

例 10.2 用拉普拉斯进行线检测。

图10.5(a)显示了一幅486×486电子电路的接线掩模的一部分(二值图像),图10.5(b)显示了其拉普拉斯图像。因为拉普拉斯图像包含有负值^①,为便于显示,做比例调节是必要的。如放大部分显示的那样,中等灰度表示零,较暗的灰度调表示负值,而较亮的色调表示正值。在放大部分,双线效应清晰可见。

首先,负值看起来可通过取拉普拉斯图像的绝对值来简单地处理。然而,如图10.5(c)所示,该方法会使线的宽度加倍。一个更合适的方法是仅使用拉普拉斯的正值(在有噪声的情形下,我们使用超过正阈值的某些值,去掉那些由噪声导致的零附近的随机变化)。如图10.5(d)中的图像所示,这种方法产生了更细的线,这些线更有用。注意,在图10.5(b)到图10.5(d)中,当线的宽度比拉普拉斯模板的尺寸宽时,这些线就被一个零值“山谷”分开了。

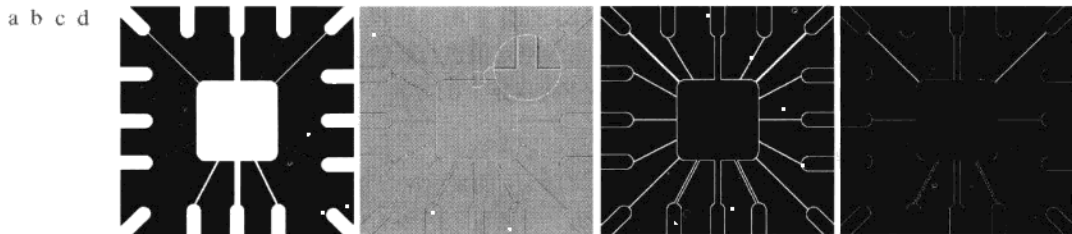


图 10.5 (a)原图像;(b)拉普拉斯图像,放大部分显示了拉普拉斯的正/负双线效应;(c)拉普拉斯的绝对值;(d)拉普拉斯的正值

这是预料之中的。例如,当把这个3×3滤波器居中放在一条宽为5个像素的恒定灰度线上时,其响应将为零,这样就产生了刚才提到的效应。当我们谈论线检测时,假设这些线要细于检测子的尺寸。不满足这一假设的线最好当做区域,并使用本节稍后讨论的边缘检测方法来处理。

图10.4(a)中的拉普拉斯检测子是各向同性的,因此其响应与方向无关(相对于该3×3拉普拉斯模板的4个方向:垂直方向、水平方向和两个对角方向)。通常,我们的兴趣在于检测特定方向的线。考虑图10.6中的模板。假设使用第一个模板对一幅具有恒定背景并包含各种线(方向为0°,±45°和90°)的图像进行滤波。最大响应将出现在图像中通过该模板中间行的一条水平线处。通过勾画一个元素为1的简单阵列,并画一条水平穿越该阵列的具有不同灰度(假设为5)的线,可很容易地验证这一点。类似的实验表明,图10.6中的第二个模板对于45°方向的线有最佳响应;第三个模板对于垂直线有最佳响应;第四个模板对于-45°方向的线有最佳响应。每个模板的首选方向用一个比其他方向更大的系数(如2)加权。每个模板中的系数之和为零,这表明恒定灰度区域中的响应为零。

回忆2.4.2节可知图像轴约定如下:原点位于左上角,正x轴指向下方,正y轴指向右方。本节中讨论的线的角度是指相对于正x轴度量的角度。例如,垂直线的角度为0°,+45°线向右下方延伸。

① 当其系数之和为零的掩模与一幅图像进行卷积时,结果图像中的像素之和也为零(见习题3.16),这表明结果中存在正像素和负像素。比例调节的目的是,使所有的负值适合于显示。

-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
水平			+45°			垂直			-45°		

图 10.6 线检测模板。角度是相对于图 2.18(b)中的坐标轴系统的

令 R_1, R_2, R_3 和 R_4 表示图 10.6 中从左到右的各个模板的响应, 其中 R 值由式(10.2-3)给出。假设使用这 4 个模板对一幅图像滤波。在该图像中的某个给定点处, 如果对于所有 $j \neq k$ 有 $|R_k| > |R_j|$, 则称该点可能与模板 k 方向的一条线更相似。例如, 如果在图像中的某个点处, 对于 $j = 2, 3, 4$ 有 $|R_1| > |R_j|$, 则说该点可能与一条水平线更相似。换句话说, 我们可能对检测特定方向上的线感兴趣。在这种情况下, 我们会使用与该方向相关的模板, 并对其输出进行阈值处理, 就像在式(10.1-2)中那样。换句话说, 如果我们对检测图像中由给定模板定义的方向上的所有线感兴趣, 则只须简单地对该图像运行这个模板, 并对结果的绝对值进行阈值处理。留下的点是有最强响应的点, 对于 1 个像素宽度的线来说, 相应的点最接近于模板定义的方向。下例说明了这一过程。

注意, 这里使用 R 来表示模板响应, 而 10.1 节中则用来表示区域, 请不要混淆。

例 10.3 特定方向线的检测

图 10.7(a) 显示了前一例子中所用的图像。假设我们的兴趣在于寻找所有的宽度为 1 个像素、方向为 45° 的线。为了这一目的, 我们使用图 10.6 中的第二个模板。图 10.7(b) 是用该模板对图像滤波后的结果。如之前那样, 图 10.7(b) 中比背景暗的色调对应于负值。图像中有两个取向为 $+45^\circ$ 方向的主要线段: 一个在左上方, 另一个在右下方。图 10.7(c) 和 (d) 显示了图 10.7(b) 中对应于这两个区域的放大部分。注意, 图 10.7(d) 中的直线段比图 10.7(c) 中的该线段亮得多。原因是图 10.7(a) 中右下方线段的宽度为 1 个像素, 而左上方线段的宽度则不是。该模板被“调谐”到检测 $+45^\circ$ 方向的 1 个像素宽的线, 我们期望, 当检测这样的线时, 其响应较强。图 10.7(e) 显示了图 10.7(b) 的正值。因为我们的兴趣在于最强响应, 我们令 T 等于图 10.7(e) 中的最大值。图 10.7(f) 使用白色显示了其值满足条件 $g \geq T$ 的点, 其中 g 是图 10.7(e) 中的图像。该图中的孤立点是对模板也有类似强响应的点。在原图像中, 这些点与它们的邻点都按这样一种方法来取向, 即模板在这些位置会产生最大的响应。用图 10.4(a) 中的模板可检测这些孤立点, 然后删除这些点; 或者可以使用上一章中讨论的形态学算子来删除这些孤立点。

10.2.4 边缘模型

边缘检测是基于灰度突变来分割图像的最常用的方法。我们从介绍一些边缘建模的方法开始, 然后讨论一些边缘检测方法。

边缘模型根据它们的灰度剖面来分类。台阶边缘是指在 1 个像素的距离上发生两个灰度级间理想的过渡。图 10.8(a) 显示了一个垂直台阶边缘的一部分和通过该边缘的一个水平剖面。例如, 用于诸如固体建模和动画领域的出现在由计算机生成的图像中的台阶边缘。这些清晰、理想的边缘可出现在 1 个像素的距离上, 不需要提供任何使它们看上去“很真实”的附加处理(如平滑)。在算法开发中, 数字台阶边缘常用做边缘模型。例如, 10.2.6 节中讨论的坎尼边缘检测算法就是用一个台阶边缘模型推导的。

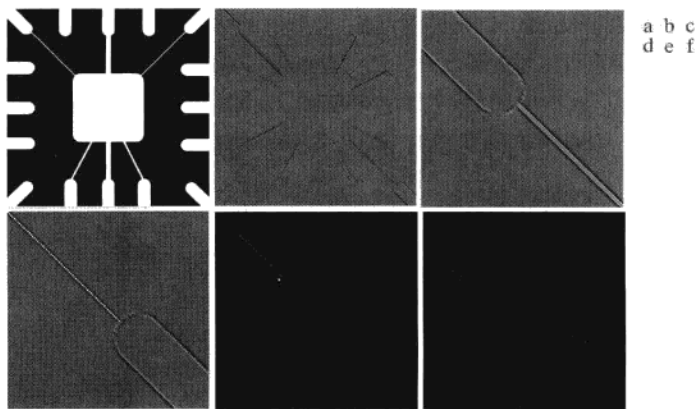


图 10.7 (a) 接线板掩模图像; (b) 使用图 10.6 中的+45°线检测子模板处理后的结果; (c) 图(b)左上方区域的放大观察图; (d) 图(b)右下区域的放大观察图; (e) 将图(b)中所有负值置为零后的图像; (f) 其值满足条件 $g \geq T$ 的所有点(白色), 其中 g 是图(e)中的图像 [为便于查看, 图(f)中的点已被放大]

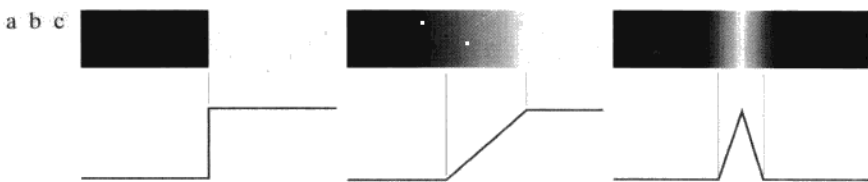


图 10.8 从左到右分别为一个台阶模型、一个斜坡模型和一个屋顶边缘模型(理想表示)及它们的相应灰度剖面

实际中, 数字图像都存在被模糊且带有噪声的边缘, 模糊的程度主要取决于聚焦机理(如光学成像中的镜头)中的限制, 而噪声水平主要取决于成像系统的电子元件。在这种情况下, 边缘被建模为一个更接近灰度斜坡的剖面, 如图 10.8(b)中的边缘。斜坡的斜度与边缘的模糊程度成反比。在这一模型中, 不再存在一条细的(1 像素宽)轨迹。相反, 一个边缘点现在是斜坡中包含的任何点, 而一条边缘线段将是一组已连接起来的这样的点。

边缘的第三种模型是所谓的“屋顶”边缘, 这种边缘具有图 10.8(c)所示的特性。屋顶边缘是通过一个区域的线的模型, 屋顶边缘的基底(宽度)由该线的宽度和尖锐度决定。在极限情形下, 当其基底为 1 个像素宽时, 屋顶边缘只不过是一条穿过图像中一个区域的一条 1 像素宽的线。例如, 在深度成像中, 当细物体(如管子)比它的等距离的背景(如墙)更接近传感器时, 出现屋顶边缘。管道更亮, 因而产生了一幅类似于图 10.8(c)中模型的图像。如先前提及的那样, 经常出现屋顶边缘的其他领域是在数字化的线条图和卫星图像中, 此时如道路这样的较细特征可由这种类型的边缘建模。

包含所有三种类型边缘的图像并不罕见。虽然模糊和噪声会导致与理想形状的偏差, 但图像中有适当锐度和适中的噪声的边缘确实存在类似于图 10.8 中边缘模型的特性, 如图 10.9 所示的剖面^①。图 10.8 中的模型允许我们在图像处理算法的开发中写出边缘的数学表达式。这些算法的性能将取决于实际边缘和在算法开发中所用模型之间的差别。

图 10.10(a)显示了从图 10.8(b)中的一段中提取出来的一幅图像。图 10.10(b)显示了一条水平灰度剖面线。该图形还显示了灰度剖面的一阶导数和二阶导数。如 10.2.1 节中讨论的那样, 当沿着灰度剖面从左到右移动时, 我们注意到, 在斜坡的开始处和在斜坡上的各个点处, 一阶导数为正。而在恒定灰度区

① 有着几个像素的剧烈斜率的斜坡边缘, 通常被当做台阶边缘处理, 目的是由同一图像中斜率更为舒缓的斜坡来对它们进行微分。

域的一阶导数为零。在斜坡的开始处，二阶导数为正；在斜坡的结束处，二阶导数为负；在斜坡上的各点处，二阶导数为零；在恒定灰度区域的各点处，二阶导数为零。对于从亮到暗过渡的边缘，刚刚讨论的导数的符号则正好相反。零灰度轴和二阶导数极值间的连线的交点称为该二阶导数的零交叉点。

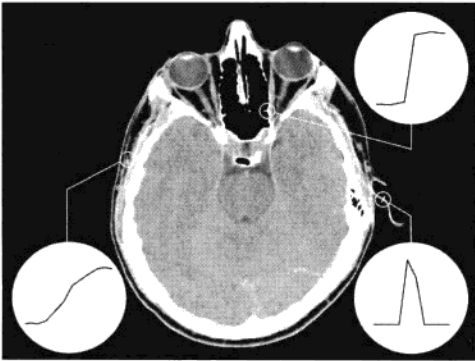


图 10.9 显示了(放大后的)实际斜坡(左下)、台阶(右上)和屋顶边缘剖面的一幅大小为 1508×1970 的图像。在由小圆所示短线段指出的区域中，剖面从暗到亮。斜坡和台阶剖面分别跨越 9 个像素和 2 个像素。屋顶边缘的基底是 3 个像素(原图像由 Vanderbilt 大学的 David R. Pickens 博士提供)

由这些观察我们可以得出结论：一阶导数的幅度可用于检测图像中的某个点处是否存在一个边缘。同样，二阶导数的符号可用于确定一个边缘像素位于该边缘的暗的一侧还是亮的一侧。我们注意到围绕一条边缘的二阶导数的两个附加性质：(1)对图像中的每条边缘，二阶导数生成两个值(一条不希望的特点)；(2)二阶导数的零交叉点可用于定位粗边缘的中心，正如我们将在本节稍后将要说明的那样。某些边缘模型可用来在进入和离开斜坡的地方平滑过渡(见习题 10.7)。然而，使用这些模型得到的结论与使用一个理想斜坡得到的结论相同，且用后者可简化理论公式。最后，尽管迄今为止我们的注意力限制在一维水平剖面上，但类似的结论适用于图像中任何方向的边缘。我们简单地定义了一个在任何期望的点与边缘方向垂直的剖面，并用刚刚讨论的对垂直边缘相同的方法对结果进行了解释。

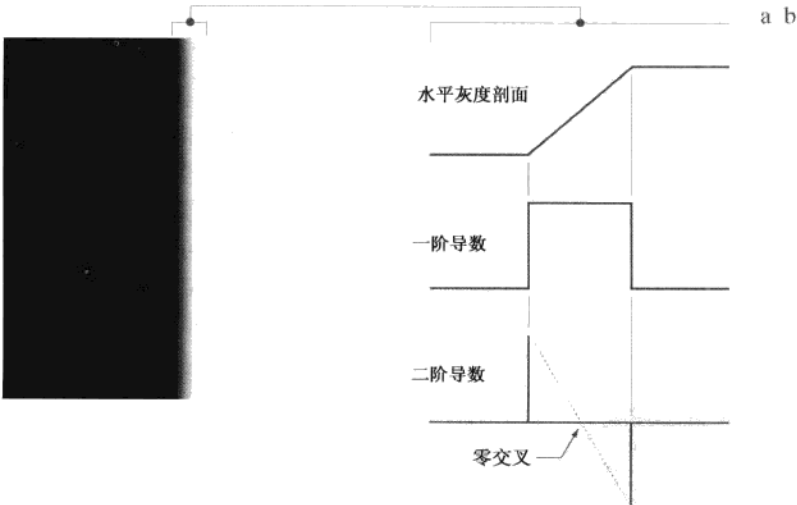


图 10.10 (a)由一条理想垂直边缘分开的两个恒定灰度区域；(b)边缘附近的细节，显示了一个水平灰度剖面及其一阶导数和二阶导数

例 10.4 一条有噪声边缘的一阶导数和二阶导数的性质。

图 10.8 中的边缘是无噪声的。图 10.11 中第一列图像段是 4 个斜坡边缘的特写，这些边缘从左边的黑色区域到右边的白色区域过渡（记住，从黑到白的整个过渡是一个单一的边缘）。左上方的图像段无噪声。第一列中的其他三幅图像被均值为零、标准差分别为 0.1, 1.0 和 10.0 个灰度级的加性高斯噪声污染。每幅图像下面的图形是一个通过图像中心的水平灰度剖面线。所有图像具有 8 比特的灰度分辨率，并用 0 和 255 分别表示黑色与白色。

整个图像段的导数计算将在下一节中讨论。现在，我们的目的只是分析灰度剖面。

考虑中间一列顶部的图像。就像我们对图 10.10 (b) 的讨论那样，左侧扫描线的导数在灰度恒定区域为零。这是在导数图像中显示的两个黑色条带。在斜坡上的各点处的导数是恒定的，并等于斜坡的斜率。在导数图像中这些恒定值显示为灰色。当我们沿中间列向下移动时，导数会变得与无噪声情形时越来越不相同。实际上，将中间列中最后一个剖面与斜坡边缘的一阶导数联系起来将会很困难。造成这种有意思的结果的原因是左列图像中的噪声几乎不可见。这些例子很好地说明了导数对噪声的敏感性。

如预想的那样，二阶导数对于噪声甚至更为敏感。无噪声图像的二阶导数显示在右列的上部。白色和黑色的细垂直线是二阶导数的正分量和负分量，就像图 10.10 中解释的那样。这些图像中的灰色表示零（如先前讨论的那样，比例缩放导致零显示为灰色）。唯一——个类似于无噪声情况的有噪声二阶导数图像对应于标准差为 0.1 的噪声。另两幅二阶导数图像和剖面清楚地表明了检测这些图像中的正分量和负分量的确很困难，而这些分量在边缘检测中确实是非常有用的二阶导数特性。

微弱的可见噪声对检测边缘所用的两个关键导数的严重影响的这一事实，是我们应记住的一个重要问题。特别地，在类似于我们刚刚讨论的水平的噪声很可能存在的应用中，使用导数之前的图像平滑处理是应该认真考虑的问题。

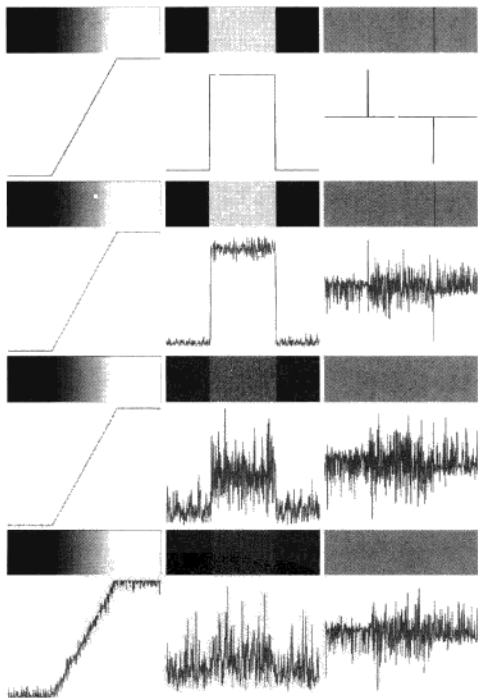


图 10.11 第一列：被均值为零、标准差分别为 0.0, 0.1, 1.0 和 10.0 个灰度级的随机高斯噪声所污染的斜坡边缘的图像和灰度剖面。第二列：一阶导数图像和灰度剖面线。第三列：二阶导数图像和灰度剖面线

我们用上面的解释给出执行边缘检测的三个基本步骤:

1. 为降噪对图像进行平滑处理。需要这一步的原因已由图10.11的第二列和第三列中的结果做了详细说明。
2. 边缘点的检测。如先前提及的那样,这是一个局部操作,从一幅图像中提取所有的点,这些点是变为边缘点的潜在候选者。
3. 边缘定位。这一步的目的是从候选边缘点中选择组成边缘点集合中的真实成员。

本节剩下的内容探讨实现这些目标的一些技术。

10.2.5 基本边缘检测

正如前一节说明的那样,为了达到寻找边缘的目的,检测灰度变化可用一阶或二阶导数来完成。在这一节中,我们讨论一阶导数,二阶导数将在10.2.6节中讨论。

图像梯度及其性质

为了在一幅图像 f 的 (x, y) 位置处寻找边缘的强度和方向,所选择的工具就是梯度,梯度用 ∇f 来表示,并用向量来定义:

为方便起见,这里重复给出 3.6.4 节中的一些公式。

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (10.2-9)$$

该向量有一个重要的几何性质,它指出了 f 在位置 (x, y) 处的最大变化率的方向。

向量 ∇f 的大小(长度)表示为 $M(x, y)$, 即

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \quad (10.2-10)$$

它是梯度向量方向变化率的值。注意, g_x, g_y 和 $M(x, y)$ 都是与原图像大小相同的图像,是 x 和 y 在 f 中的所有像素位置上变化时产生的。实践中,我们通常称后一图像为梯度图像,或者在含义很清楚时简称为梯度。如 2.6.1 节中定义的那样,求和、平方和开方操作都是阵列操作。

梯度向量的方向由下列对于 x 轴度量的角度给出:

$$\alpha(x, y) = \arctan \left[\frac{g_y}{g_x} \right] \quad (10.2-11)$$

如在梯度图像的情况那样, $\alpha(x, y)$ 也是与由 g_y 除以 g_x 的阵列创建的尺寸相同的图像。任意点 (x, y) 处一个边缘的方向与该点处梯度向量的方向 $\alpha(x, y)$ 正交。

例 10.5 梯度的性质。

图 10.12 显示了包含一段直的边缘线段放大的一部分。所显示的每个方块对应于一个像素,我们的兴趣是得到用一个方框强调的点处边缘的强度和方向。灰色像素的值为 0,白色像素的值为 1。下面我们说明该例子,为计算 x 方向和 y 方向的梯度,该方法使用一个关于一点为中心的 3×3 邻域,简单地从底部一行的像素中减去顶部一行邻域中的像素,得到 x 方向的偏导数。类似地,我们从右列的像素中减去左列的像素得到 y 方向的偏导数。接下来,用这些差值作为偏导数的估计,在这一点处有 $\partial f / \partial x = -2$ 和 $\partial f / \partial y = 2$ 。从而有

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

由此, 我们可以得到这一点处的 $M(x, y) = 2\sqrt{2}$ 。类似地, 相同点处梯度向量的方向遵循式(10.2-11): $\alpha(x, y) = \arctan(g_y/g_x) = -45^\circ$, 它与相对于 x 轴的正方向度量的 135° 相同。图 10.12(b) 显示了该梯度向量及其方向角。

回忆 2.4.2 节可知, 图像坐标系统的原点位于左上角, 正 x 轴向下延伸, 正 y 轴向右延伸。

图 10.12(c) 说明了之前提到的一个重要事实, 即某点的边缘与该点的梯度向量正交。因此, 在这个例子中, 边缘的方向角是 $\alpha - 90^\circ = 45^\circ$ 。图 10.12(a) 中的所有边缘点都有相同的梯度, 所以, 整个边缘段都处在相同的方向上。梯度向量有时也称为边缘法线。当向量通过除以其幅值 [见式(10.2-10)] 而归一化为单位长度时, 结果向量通常称为边缘单位法线。

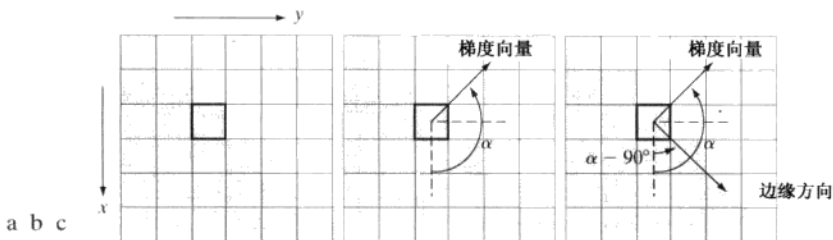


图 10.12 用梯度确定某个点处的边缘强度和方向。注意, 某点处的边缘垂直于该点处的梯度向量的方向。图中的每个方块表示一个像素

梯度算子

要得到一幅图像的梯度, 则要求在图像的每个像素位置处计算偏导数 $\partial f/\partial x$ 和 $\partial f/\partial y$ 。我们处理的是数字量, 因此要求关于一点的邻域上的偏导数的数字近似。由 10.2.1 节, 我们可知

$$g_x = \frac{\partial f(x, y)}{\partial x} = f(x+1, y) - f(x, y) \quad (10.2-12)$$

和

$$g_y = \frac{\partial f(x, y)}{\partial y} = f(x, y+1) - f(x, y) \quad (10.2-13)$$

这两个公式对所有 x 和 y 的有关值可用图 10.13 中的一维模板通过对 $f(x, y)$ 的滤波来执行。

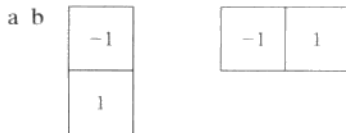


图 10.13 用于实现式(10.2-12)和式(10.2-13)的一维模板

当对对角线方向的边缘感兴趣时, 我们需要一个二维模板。罗伯特交叉梯度算子(Roberts[1965])是最早尝试使用具有对角优势的二维模板之一。考虑图 10.14(a) 中的 3×3 区域。罗伯特算子以求对角像素之差为基础:

$$g_x = \frac{\partial f}{\partial x} = (z_9 - z_5) \quad (10.2-14)$$

和

$$g_y = \frac{\partial f}{\partial y} = (z_8 - z_6) \quad (10.2-15)$$

用于计算梯度偏导数的滤波器模板, 通常称为梯度算子、差分算子、边缘算子或边缘检测子。

这些导数可以使用图 10.14(b) 和(c) 中的模板对图像进行滤波来实现。

2×2 大小的模板在概念上很简单,但是它们对于用关于中心点对称的模板来计算边缘方向不是很有用,其最小模板大小为 3×3。这些模板考虑了中心点对端数据的性质,并携带有关于边缘方向的更多信息。用大小为 3×3 的模板来近似偏导数的最简单的数字近似由下式给出:

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \tag{10.2-16}$$

和

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \tag{10.2-17}$$

尽管这些公式包含了较大的邻域,但我们仍处理灰度值间的差。因此之前讨论一阶导数时得到的结论仍然适用。

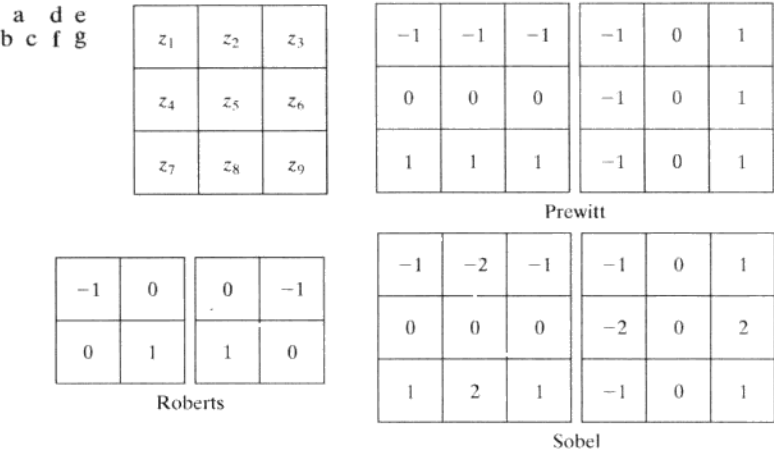


图 10.14 一幅图像的 3×3 区域(z 项是灰度值)和用于计算标记点 z_5 处的梯度的不同模板

在这些公式中, 3×3 区域的第三行和第一行之差近似为 x 方向的导数, 第三列和第一列之差近似为 y 方向的导数。直观上, 我们可以预料这些近似要比用罗伯特算子得到的近似更准确。式(10.2-16)和式(10.2-17)可用图 10.14(d)和(e)中的两个模板通过滤波整个图像来实现。这两个模板称为 Prewitt 算子(Prewitt[1970])。

对这两个公式的一个小小的变化是在中心系数上使用一个权值 2:

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \tag{10.2-18}$$

和

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \tag{10.2-19}$$

可以证明, 在中心位置处使用 2 可以平滑图像(见习题 10.10)。图 10.14(f)和(g)显示了用于实现式(10.2-18)和式(10.2-19)的模板。这些模板称为 Sobel 算子(Sobel[1970])。

Prewitt 模板实现起来比 Sobel 模板更为简单,但它们之间计算上的小差别并不是问题。正如前一节提到的那样, Sobel 模板能较好地抑制(平滑)噪声的特性这一事实使得它们更可取,因为在处理导数时噪声抑制是一个重要的问题。注意,图 10.14 中所有模板中的系数之和为零,正如导数算子所预示的那样,恒定灰度区域的响应为零。

刚才讨论的模板用于在图像的每个像素位置处得到梯度分量 g_x 和 g_y 。然后,用这两个偏导数估计边缘的强度和方向。计算梯度的幅度时,要求按式(10.2-10)所示的方法联合使用 g_x 和 g_y 。然而,

图10.16(a)中所示原图像的分辨率相当高(834×1114 像素),且在图像获取的距离上,墙砖对于图像细节的贡献仍十分显著。这种精细的细节在边缘检测中通常是不符合需要的,因为它往往表现为噪声,导数计算会增强这种噪声,从而使图像中主要边缘的检测变得复杂。减少精细细节的一种方法是对图像进行平滑处理。图10.18显示了与图10.16中相同的图像序列,但先使用一个大小为 5×5 的均值滤波器对原图像进行了平滑(关于平滑滤波器,见 3.5 节)。现在每个模板的响应几乎未显示出由砖块造成的影响,得到的结果几乎都是主要的边缘。

在图 10.16 和图 10.18 中,水平和垂直 Sobel 模板不区分±45°方向的边缘是很明显的。如果沿对角方向强调边缘很重要,则应使用图10.15 中所示的一个模板。图10.19(a)和(b)分别显示了45°和-45°的 Sobel 模板的绝对响应。在这些图中,这些模板的较强的对角响应很明显。两个对角模板对水平边缘和垂直边缘都有类似的响应,但正如所期望的那样,它们在这些方向上的响应要比之前讨论的水平和垂直模板的响应弱。

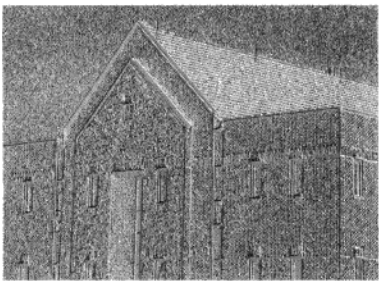


图 10.17 使用式(10.2-11)计算的梯度角度图像。在该图像中的恒定灰度区域指出这些区域中所有像素位置处的梯度向量的方向均相同

平滑后图像的最大边缘强度(幅度)会随着平滑模板尺寸的增加而减小(见习题 10.13)。

a b c d

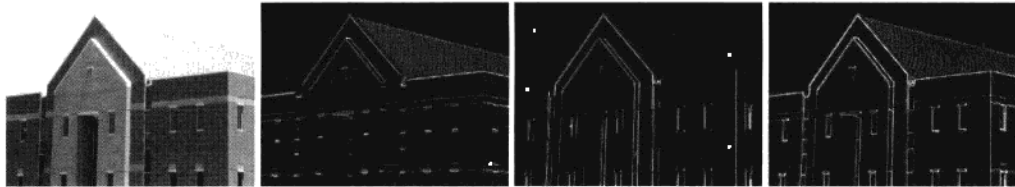
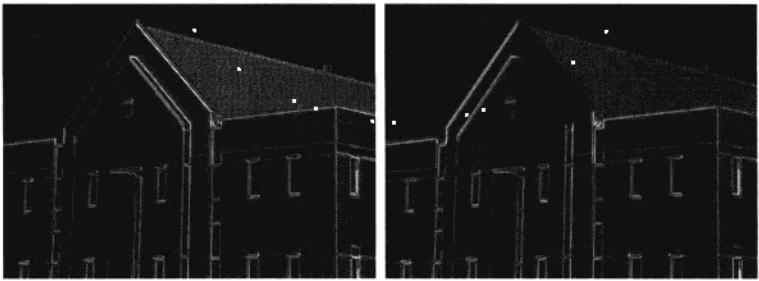


图 10.18 与图 10.16 中的相同图像序列,但在边缘检测之前,用大小为 5×5 的均值滤波器对原图像进行了平滑



a b

图 10.19 对角边缘检测:(a)使用图 10.15(c)中的模板得到的结果;(b)使用图 10.15(d)中的模板得到的结果。两种情况下的输入图像都是图 10.18(a)

与阈值处理相结合的梯度

图 10.18 中的结果说明,在计算梯度前对图像进行平滑处理,边缘检测可做更多的选择。实现相同的基本目标的另一种方法是对梯度图像进行阈值处理。例如,图 10.20(a)显示了图 10.16(d)经阈值处理后的梯度图像,此时其值大于或等于梯度图像的最大值的 33% 的像素显示为白色,而低于该阈值的像素则显示为黑色。该图像与图 10.18(d)进行比较,我们看到阈值处理后的图像中边缘更少,并且

选择用于生成图 10.20(a)的阈值的目的,是消除砖块导致的多数较小边缘。回忆可知,这是在计算梯度前平滑图 10.16 中的图像的基本目标。

734
~
735

这幅图像中的边缘要尖锐得多(例如,见屋瓦中的边缘)。另一方面,在阈值处理后的图像中,许多边缘,如确定屋顶远边缘的45°线,被断开了。

当目的是突出主要边缘并尽可能维护连接性时,实践中通常平滑处理和阈值处理两者都使用。图10.20(b)显示了图10.18(d)经阈值处理后的结果,它是平滑后图像的梯度。该结果显示断开的边缘减少了一些;例如,比较图10.20(a)和(b)中的45°边缘。当然,这些由于模糊处理其灰度值严重降低的边缘(即屋瓦中的边缘)可能因阈值处理而完全消除。在10.2.7节我们再回来讨论边缘断线的问题。

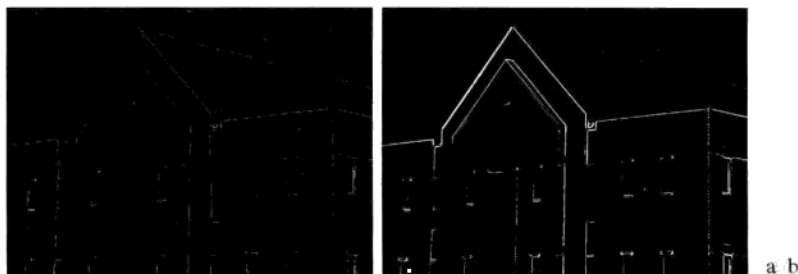


图10.20 (a)图10.16(d)中图像经阈值处理后的图像,阈值选择为图像中最高值的33%;该阈值刚好高到足以消除梯度图像中的多数砖块边缘;(b)图10.18(d)中图像经阈值处理后的图像,它是使用等于图像中最高值的33%的阈值得到的

10.2.6 更先进的边缘检测技术

前一节中讨论的边缘检测方法是使用一个或多个模板对图像进行滤波为基础的,而未对图像特性和噪声内容采取预防措施。在这一节,我们讨论更为先进的技术,尝试考虑诸如图像噪声和边缘本身特性的因素改进简单的边缘检测方法。

Marr-Hildreth 边缘检测器

最早的成功地尝试将更高级的分析结合到边缘检测处理之一应归功于 Marr 和 Hildreth[1980]。正如前节所讨论的那样,当时,边缘检测方法是使用较小的算子为基础的(如 Sobel 模板)。Marr 和 Hildreth 证明了:(1)灰度变化与图像尺寸无关,因此他们的检测要求使用不同尺寸的算子;(2)灰度的突然变化会在一阶导数中引起波峰或波谷,或在二阶导数中等效地引起零交叉(就像我们在图10.10中见到的那样)。

这些概念建议,用于边缘检测的算子应有两个显著的特点。第一个和最重要的特点是它应该是一个能计算图像中每一点处的一阶导数或二阶导数的数字近似的微分算子。第二,它应能被“调整”以便在任何期望的尺寸上起作用,因此,大的算子也可用于检测模糊边缘,小的算子可用于检测锐度集中的精细细节。

Marr 和 Hildreth 论证过,满足这些条件的最令人满意的算子是滤波器 $\nabla^2 G$, 如在3.6.2节中定义的那样, ∇^2 是拉普拉斯算子($\partial^2/\partial x^2 + \partial^2/\partial y^2$), 而 G 是标准差为 σ (有时 σ 也称为空间常数)的二维高斯函数

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (10.2-21)$$

为证明边缘检测与图像尺寸无关,可以考虑图10.8(c)中的屋顶边缘。如果图像的尺寸减小,则边缘会更细。

式(10.2-21)与二维高斯概率密度函数的定义的不同在于常数项 $1/2\pi\sigma^2$ 。如果在某个给定应用中期望得到一个精确的表达式,则乘以常数可添加到式(10.2-23)的最终结果中。

为求 $\nabla^2 G$ 的表达式, 我们执行如下微分:

$$\begin{aligned}\nabla^2 G(x, y) &= \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} = \frac{\partial}{\partial x} \left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right] + \frac{\partial}{\partial y} \left[\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right] \\ &= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}\end{aligned}$$

(10.2-22)

整理各项后给出如下最终表达式:

$$\nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

(10.2-23)

该表达式称为高斯拉普拉斯(LoG)。

图 10.21 (a) 到 (c) 显示了一个 LoG 的负函数的三维图、图像和剖面(注意 LoG 的零交叉出现在 $x^2 + y^2 = \sigma^2$ 处, 它定义了一个中心位于原点、半径为 $\sqrt{2}\sigma$ 的圆)。因为图 10.21 (a) 中显示的形状, LoG 函数有时也称为墨西哥草帽算子。图 10.21 (d) 显示了一个 5x5 的模板, 它近似于图 10.21 (a) 中的形状(在实践中, 我们会用该模板的负模板)。这个近似并不是唯一的, 其目的是获取 LoG 函数的基本形状; 根据图 10.21 (a), 这意味着一个正的中心项由紧邻的负区域包围着, 中心项的值以距原点的距离为函数而增大, 而外层区域的值为零。系数之和必须为零, 从而模板的响应在恒定灰度区域为零。

注意图 10.21(c)中剖面与图 4.37(c)中高通滤波器之间的相似性。这样, 我们可期望 LoG 的作用类似于一个高通滤波器。

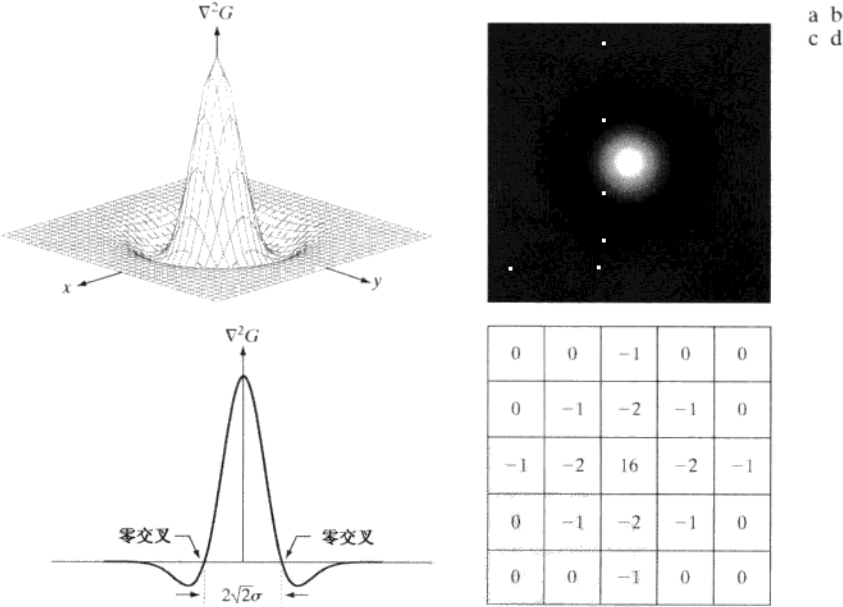


图 10.21 (a) 负 LoG 的三维图; (b) 显示为一幅图像的负 LoG 图; (c) 显示了零交叉的图 (a) 的横截面; (d) 对图 (a) 中形状 5x5 模板的近似, 实际中将使用该模板的负值

736
~
737

任意尺寸的模板可以通过对式 (10.2-23) 取样并标定系数以使系数之和为零来生成。生成 LoG 滤波器的一种更有效的方法是以希望的 $n \times n$ 尺寸对式 (10.2-21) 取样, 然后将结果阵列与一个拉普拉斯模

板进行卷积^①，例如图 10.4(a) 中的模板。因为用一个系数之和为零的模板对图像阵列卷积产生一个元素之和也为零的结果(见习题 3.16 和习题 10.14)，故这种方法自动满足 LoG 滤波器系数之和为零的要求。在本节的后面将讨论 LoG 滤波器尺寸的选取问题。

选择算子 $\nabla^2 G$ 的基础有两个基本概念。第一，算子的高斯部分会模糊图像，从而在尺寸上将结构的灰度(包括噪声)降低到远小于 σ 的程度。与 3.5 节中讨论的和图 10.18 中使用的平均形式不同的是高斯函数在空间和频率两个域平滑图像(见 4.8.3 节)，因而在原图像中引入不存在的人为干扰(如振铃)的可能性很小。另一个概念涉及 ∇^2 ，即滤波器的二阶导数部分。尽管一阶导数可用于检测灰度突变，但它们是有方向的算子。另一方面，拉普拉斯有各向同性(旋转不变)的重要优点，这不仅符合人的视觉系统特性(Marr[1982])，而且对任何模板方向的灰度变化有相等的响应，从而避免了使用多个模板去计算图像中任何点处的最强响应。

Marr-Hildreth 算法由 LOG 滤波器与一幅输入图像 $f(x, y)$ 卷积组成，即

$$g(x, y) = [\nabla^2 G(x, y)] \star f(x, y) \quad (10.2-24)$$

然后寻找 $g(x, y)$ 的零交叉来确定 $f(x, y)$ 中边缘的位置。因为这些都是线性操作，故式(10.2-24)也可写为

$$g(x, y) = \nabla^2 [G(x, y) \star f(x, y)] \quad (10.2-25)$$

它指出我们可以先使用一个高斯滤波器平滑图像，然后计算该结果的拉普拉斯。这两个公式给出了相同的结果。

该表达式是使用式(3.4-2)在空间中实现的。也可使用式(4.7-1)在频率域中实现。

Marr-Hildreth 边缘检测算法可小结如下：

1. 用一个对式(10.2-21)取样得到的 $n \times n$ 的高斯低通滤波器对输入图像滤波。
2. 计算由第一步得到的图像的拉普拉斯，例如，使用图 10.4(a) 的 3×3 模板 [第一步和第二步实现式(10.2-25)]。
3. 找到步骤 2 所得图像的零交叉。

为确定高斯滤波器的大小，仍采用位于二维高斯表面下其均值在 $\pm 3\sigma$ 之间的 99.7%。这样，作为一种经验法则，一个大小为 $n \times n$ 的 LoG 离散滤波器，其 n 值应是大于等于 6σ 的最小奇整数。 n 小于该值的滤波器模板会“截断”LoG 函数，截断的程度与模板的大小成反比；而使用较大的模板对结果的影响不大。

在滤波后的图像 $g(x, y)$ 的任意像素 p 处，寻找零交叉的一种方法是用以 p 为中心的一个 3×3 邻域。 p 点处的零交叉意味着至少有两个相对的邻域像素的符号不同。有 4 种要测试的情况：左/右、上/下和两个对角。如果 $g(x, y)$ 的值与一个阈值比较(一种通常的方法)，那么不仅相对邻域的符号不同，而且它们的数值差的绝对值还必须超过这个阈值。这时，我们称 p 为一个零交叉像素。下面的例 10.7 中将说明这一方法。

试图通过查找满足 $g(x, y) = 0$ 的坐标 (x, y) 来找到零交叉是不实际的，因为噪声和/或计算不准确。

零交叉是 Marr-Hildreth 边缘检测方法的关键特征。前段中讨论的方法是很有吸引力的，因为它实现简单，并且通常能给出好的结果。

如果在某个特殊应用中使用这一方法找到的零交叉位置的准确性不够，那么可采用由 Huertas and Medioni[1986]提出的采用亚像素精度来寻找零交叉方法。

① LoG 是一个对称滤波器，所以使用相关或卷积的空间滤波将产生相同的结果。为与关于这一主题的文獻保持一致，这里我们使用卷积这一术语来表示线性滤波。此外，也为读者提前了解这一术语，因为在其他内容中也会遇到它。记住，3.4.2 节末针对这一主题给出的注释很重要。

例 10.7 Marr-Hildreth 边缘检测算法说明。

图 10.22(a) 显示了早先使用过的建筑物图像, 图 10.22(b) 是 Marr-Hildreth 算法的步骤 1 和步骤 2 得到的结果, 其中 $\sigma = 4$ (约为图像短边长度的 0.5%) 和 $n = 25$ (如早先讨论的那样, 大于等于 6σ 的最小奇整数)。如在图 10.5 中那样, 这幅图像中的灰度调已被标定过。图 10.22(c) 显示了使用上面讨论过的阈值为零的 3×3 邻域方法得到的零交叉。注意, 所有的边缘形成了一个闭环。这种所谓的意大利通心粉效应是使用零阈值方法的严重缺点(见习题 10.15)。我们使用正阈值避免闭环边缘。

图 10.22(d) 显示了使用近似等于 LoG 图像最大值的 4% 的阈值的结果。注意, 很容易地检测出了大多数的主要边缘, 并且滤掉了一些“无关”特征, 例如砖块和屋瓦的边缘。正如我们在下一节要说明的那样, 使用前节讨论的基于梯度的边缘检测技术实际上是不可能得到这种类型的性能的。使用零交叉检测边缘的另一个重要结果是可得到一个像素宽的边缘。这一特性简化了诸如边缘连接的后续阶段的处理。

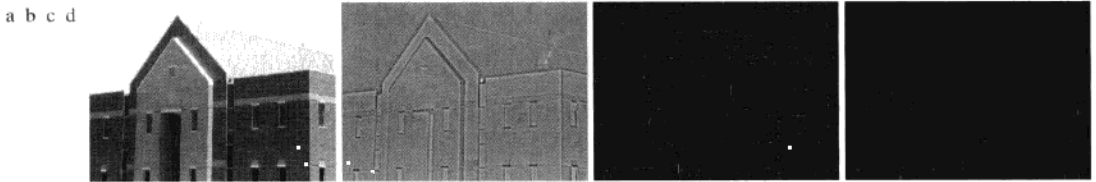


图 10.22 (a) 大小为 834×1114 像素、灰度值已标定到范围 $[0, 1]$ 的原图像; (b) Marr-Hildreth 算法的步骤 1 和步骤 2 得到的结果, 其中 $\sigma = 4$ 和 $n = 25$; (c) 使用零阈值的图(b)的零交叉(注意闭环边缘); (d) 使用等于图(b)中图像的最大值的 4% 的阈值时, 找到的零交叉。注意细边缘

考虑到早先提及的灰度变化取决于数值范围的事实, 有时所用的过程是使用各种 σ 值来对一幅图像进行滤波。然后, 所得零交叉边缘图与仅为全部图形保留的公共边缘相结合。这种方法可得到很有用的信息, 但由于其复杂性, 实践中它多被用做使用单一滤波器选择合适的 σ 值的设计工具。

Marr and Hildreth[1980]指出过, 使用高斯差分(DoG)来近似式(10.2-23)中的 LoG 滤波器是可能的:

$$\text{DoG}(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \quad (10.2-26)$$

其中 $\sigma_1 > \sigma_2$ 。实验结果表明, 在人的视觉系统中, 某些“通道”就方向和频率而论是有选择性的, 且可以使用式(10.2-26)以 1.75:1 的标准差比率来建模。Marr 和 Hildreth 建议过, 使用 1.6:1 的比率不仅可以保持这些观察的基本特性, 而且还可对 LoG 函数提供一个更接近的“工程”近似。为在 LoG 和 DoG 之间进行有意义的比较, 对于 LOG, σ 值必须按照如下公式选择, 以便 LoG 和 DoG 具有相同的零交叉(见习题 10.17):

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln \left[\frac{\sigma_1^2}{\sigma_2^2} \right] \quad (10.2-27)$$

当使用这个 σ 值时, 尽管 LoG 和 DoG 的零交叉相同, 但它们的幅度大小会不同。我们可以通过标定这两个函数使得它们兼容, 以便它们在原点处有相同的值。

图 10.23(a) 和 (b) 中的剖面线是分别使用标准差比率 1:1.75 和 1:1.6 产生的(按照惯例, 所显示的曲线被反转了, 就像在图 10.21 中那样)。LoG 剖面线显示为实线, 而 DoG 剖面线显示为虚线。所示曲线是通过 LoG 和 DoG 阵列中心的灰度剖面线, 这两个阵列分别是通过对式(10.2-23) (前面有一个常数 $1/2\pi\sigma^2$) 和式(10.2-26) 取样产生的。所有曲线在原点处的幅度均被归一化为 1。如图 10.23(b) 所示, 1:1.6 的比率在 LoG 和 DoG 函数之间产生了一个更接近的近似。

高斯差分是一个高通滤波器, 详见 4.7.4 节中的讨论。

LoG 和 DoG 滤波操作均可直接使用一维卷积代替二维卷积来实现(见习题 101.9)。对于一幅大小为 $M \times N$ 的图像和一个大小为 $n \times n$ 的滤波器,这样做可将每次卷积所需的乘法和加法次数,从二维卷积的与 $n^2 MN$ 成正比,减少到一维卷积的与 $n MN$ 成正比。这种实现上的差别是很有意义的。例如,如果 $n = 25$,则一维卷积实现的乘法与加法操作次数将比二维卷积实现少 12 倍。



图 10.23 (a)用比率为 1.75 : 1 的标准差的 LOG(实线)和 DOG(虚线)的负值剖面线,(b)使用 1.6 : 1 的比率得到的剖面线

坎尼边缘检测器

虽然其算法更为复杂,但本节讨论的坎尼检测器(Canny[1968])是迄今为止讨论过的边缘检测器中最为优秀的。坎尼方法基于三个基本目标:

1. 低错误率。所有边缘都应被找到,并且应该没有伪响应。也就是检测到的边缘必须尽可能是真实的边缘。
2. 边缘点应被很好地定位。已定位边缘必须尽可能接近真实边缘。也就是由检测器标记为边缘的点和真实边缘的中心之间的距离应该最小。
3. 单—的边缘点响应。对于真实的边缘点,检测器仅应返回一个点。也就是真实边缘周围的局部最大数应该是最小的。这意味着在仅存一个单一边缘点的位置,检测器不应指出多个边缘像素。

坎尼的工作的本质是从数学上表达了前面的三个准则,并试图找到这些表达式的最佳解。通常,寻找一个满足前述目标的、严格形式的解是很困难的(或不可能的)。然而,对由加性高斯白噪声污染的一维台阶边缘使用数字最佳化会得到这样一个结论,即对最佳台阶边缘检测器的一个较好近似^①是高斯一阶导数:

$$\frac{d}{dx} e^{-\frac{x^2}{2\sigma^2}} = \frac{-x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (10.2-28)$$

回忆可知,白噪声是指具有这样一个频谱的噪声,该频谱在一个指定的频率段中是连续且均匀的。白高斯噪声则是幅度值分布为高斯的白噪声。高斯白噪声是许多真实世界情形的较好近似,并且可产生数学上可处理的模型。高斯白噪声的一个有用性质是其值是统计独立的。

将该结果推广到二维情况会涉及识别一维方法是否仍适用于边缘法线的方向(见图 10.12)。因为预先法线方向未知,这就要求在所有可能的方向上应用一维边缘检测器。这一任务可由如下方法来近似:首先使用一个环形二维高斯函数平滑图像,计算结果的梯度,然后使用梯度幅度和方向来估计每一点处的边缘强度和方向。

令 $f(x, y)$ 表示输入图像, $G(x, y)$ 表示高斯函数:

$$G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (10.2-29)$$

我们用 G 和 f 的卷积形成一幅平滑后的图像 $f_s(x, y)$:

$$f_s(x, y) = G(x, y) \star f(x, y) \quad (10.2-30)$$

① Canny[1986]证明,使用高斯近似与使用最优数值解相比,所得结果仅差约 20%。这种幅度差在多数应用中通常是难以察觉的。

像 10.2.5 节讨论的那样, 在这一操作源自计算梯度幅度和方向:

$$M(x, y) = \sqrt{g_x^2 + g_y^2} \tag{10.2-31}$$

和

$$\alpha(x, y) = \arctan \left[\frac{g_y}{g_x} \right] \tag{10.2-32}$$

其中, $g_x = \partial f_s / \partial x$, $g_y = \partial f_s / \partial y$ 。图 10.14 中的任何滤波器模板对都可用来得到 g_x 和 g_y 。式(10.2-30)是使用一个大小为 $n \times n$ 的高斯模板实现的, 其大小将在下面讨论。记住, $M(x, y)$ 和 $\alpha(x, y)$ 是与计算出它们的图像的尺寸相同的阵列。

因为它是使用梯度产生的, 典型地, $M(x, y)$ 在局部最大值周围通常包含更宽的范围(回忆 10.2.1 节中关于使用梯度得到边缘的讨论)。下一步是细化那些边缘。一种方法是使用非最大抑制。这可通过几种方式来做, 但该方法的本质是指定边缘法线的许多离散方向(梯度向量)。例如, 在一个 3×3 区域内, 对于一个通过该区域中心点的边缘, 我们可以定义四个方向^①: 水平、垂直、+45°和-45°。图 10.24(a)显示了一个水平边缘的两个可能方向。因为我们必须把所有可能的边缘方向量化为四个方向, 故必须定义一个方向范围, 在该范围内, 我们考虑一个水平方向的边缘。我们由边缘法线的方向来确定边缘方向, 边缘法线方向可以直接使用式(10.2-32)从图像数据得到。如图 10.24(b)所示, 如果边缘法线方向的范围是从-22.5°到 22.5°, 或者是从-157.5°到 157.5°, 则我们称该边缘为水平边缘。图 10.24(c)显示了对应于所考虑的四个人方向的角度范围。

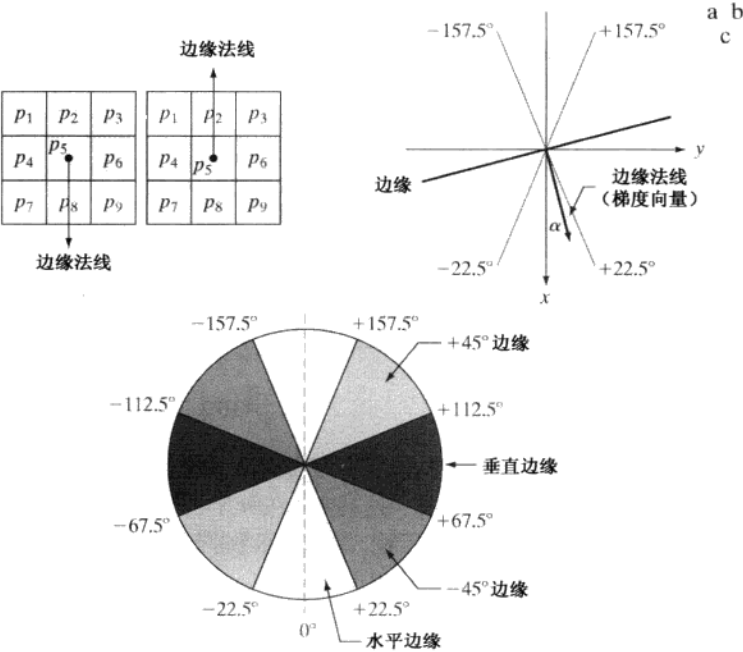


图 10.24 (a)在一个 3×3 邻域中, 一个水平边缘(灰色)的两个可能方向; (b)一个水平边缘的边缘法线的方向角 α 的值的范围(灰色); (c)在一个 3×3 邻域中, 4 种类型的边缘方向的边缘法线的角度范围。每个边缘方向都有两个范围, 以对应的灰色显示

① 记住, 每个边缘均有两个可能的方向。例如, 其法线为 0° 的边缘和法线为 180° 的边缘是同一个水平边缘。

令 d_1, d_2, d_3 和 d_4 表示刚才讨论的 3×3 区域的四个基本边缘方向：水平、 -45° 、垂直、 $+45^\circ$ 。对于在 $\alpha(x, y)$ 中以每一点 (x, y) 为中心的 3×3 区域，我们可给出如下非最大抑制方案：

1. 寻找最接近 $\alpha(x, y)$ 的方向 d_k 。
2. 如果 $M(x, y)$ 的值至少小于沿 d_k 的两个邻居之一，则令 $g_N(x, y) = 0$ (抑制)；否则，令 $g_N(x, y) = M(x, y)$ ，这里 $g_N(x, y)$ 是非最大抑制后的图像。例如，参考图 10.24(a)，令 (x, y) 在 p_5 处，并假设一个水平边缘通过 p_5 ，在步骤 2 中我们感兴趣的像素是 p_2 和 p_8 。图像 $g_N(x, y)$ 仅包含细化后的边缘；它等于抑制了非最大边缘点的 $M(x, y)$ 。

最后的操作是对 $g_N(x, y)$ 进行阈值处理，以便减少伪边缘点。在 10.2.5 节中，我们使用单阈值进行阈值处理，其中低于该阈值的所有值均置为零。如果我们设的阈值过低，则仍会存在一些伪边缘(称为伪正值)。如果阈值设得过高，则会删除实际上有效的边缘点(伪负值)。坎尼算法通过使用滞后阈值试图改进这一状况，如我们在 10.3.6 节中讨论的那样，使用两个阈值：一个低阈值 T_L 和一个高阈值 T_H 。坎尼建议，高阈值和低阈值的比率应为 2:1 或 3:1。

我们可以将阈值操作想象为创建两幅附加的图像

$$g_{NH}(x, y) = g_N(x, y) \geq T_H \quad (10.2-33)$$

和

$$g_{NL}(x, y) = g_N(x, y) \geq T_L \quad (10.2-34)$$

其中，开始时 $g_{NH}(x, y)$ 和 $g_{NL}(x, y)$ 均被置为零。阈值处理后， $g_{NH}(x, y)$ 的非零像素通常比 $g_{NL}(x, y)$ 少，但是 $g_{NH}(x, y)$ 中的所有非零像素都将包含在 $g_{NL}(x, y)$ 中，因为后一图像是使用一个低阈值形成的。通过令

$$g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y) \quad (10.2-35)$$

我们从 $g_{NL}(x, y)$ 中删除所有来自 $g_{NH}(x, y)$ 的非零像素。 $g_{NH}(x, y)$ 和 $g_{NL}(x, y)$ 中的非零像素可分别看成是“强”和“弱”边缘像素。

阈值处理后， $g_{NH}(x, y)$ 中的所有强像素均被假设为有效的边缘像素，并被立即标记。取决于 T_H 的值， $g_{NH}(x, y)$ 中的边缘通常会存在缝隙。较长的边缘用下列步骤形成：

- (a) 在 $g_{NH}(x, y)$ 中定位下一个未被访问的边缘像素 p 。
- (b) 在 $g_{NL}(x, y)$ 中将所有的弱像素标记为有效边缘像素，用 8 连通性的连接方法连接到 p 。
- (c) 如果 $g_{NH}(x, y)$ 中的所有非零像素已被访问，则跳到步骤(d)，否则返回步骤(a)。
- (d) 将 $g_{NL}(x, y)$ 中未标记为有效边缘像素的所有像素置零。

在这一过程的末尾，将来自 $g_{NL}(x, y)$ 的所有非 0 像素附加到 $g_{NH}(x, y)$ ，用坎尼算子形成最终的输出图像。

我们使用了两幅附加的图像 $g_{NH}(x, y)$ 和 $g_{NL}(x, y)$ 来简化讨论。在实际中，滞后阈值处理可在非最大抑制期间直接执行，并且，可以通过形成一个与它们相联系的强像素和弱像素的列表在 $g_N(x, y)$ 中直接实现。

总结一下，坎尼边缘检测算法是由下列基本步骤组成的：

1. 用一个高斯滤波器平滑输入图像。
2. 计算梯度幅值图像和角度图像。
3. 对梯度幅值图像应用非最大抑制。
4. 用双阈值处理和连接分析来检测并连接边缘。

尽管非最大抑制后的边缘比原来的梯度边缘细,但仍保持了比 1 像素粗的边缘。为得到 1 个像素粗的边缘,典型地,紧跟步骤 4,用一遍边缘细化算法(见 9.5.5 节)。

正如先前提到的那样,平滑是用指定的大小为 $n \times n$ 的高斯模板与输入图像进行卷积来完成的。我们可以使用前面章节讨论过的与 Marr-Hildreth 算法有关的方法决定 n 值。也就是,用对式(10.2-29)取样所产生的一个滤波器模板,从而使 n 是大于或等于 6σ 的最小奇整数,以便基本上提供高斯滤波器的“完美”平滑能力。如果实际考虑要求一个较小的滤波器模板,则折中方案是较小 n 值的平滑效果会差一些。

关于实现的最终意见:正如先前讨论 Marr-Hildreth 边缘检测器那样,式(10.2-29)中的二维高斯函数被分解为两个一维高斯函数的乘积。这样,坎尼算法的步骤 1 就可用一维卷积公式来表达,即一次对图像中的一行(列)进行操作,然后再对处理结果中的列(行)操作。此外,如果我们使用式(10.2-12)和式(10.2-13)中的近似,则也可将步骤 2 中要求的梯度计算以一维卷积来实现(见习题 10.20)。

例 10.8 坎尼边缘检测方法说明。

图 10.25(a)显示了我们熟悉的建筑物图像。为了比较,图 10.25(b)和(c)分别显示了先前在图 10.20(b)中用阈值化后的梯度得到的结果和在图 10.22(d)中使用 Marr-Hildreth 检测子得到的结果。回忆一下,选取生成这两幅图像所用的参数来检测主要边缘,同时要减少“无关”特征,如由砖块和屋瓦引起的边缘。

图 10.25(d)显示了用坎尼算法得到的结果,使用的参数为: $T_L = 0.04$, $T_H = 0.10$ (低阈值的 2.5 倍), $\sigma = 4$, 以及对应于大于 6σ 的最小奇整数的、大小为 25×25 的模板。这些参数可交互式地选择,以达到前面章节中对梯度和 Marr-Hildreth 图像扫描规定的目标。将坎尼图像与其他两幅图像比较,我们看到主要边缘在细节上的明显改进,同时,在坎尼结果中抑制了更多的无关特性。例如,注意坎尼算法检测到了图像上部镶嵌在墙砖中间的水泥条带的两个边缘,然而,阈值处理后的梯度图像中则消除了这两个边缘,并且, Marr-Hildreth 图像中则仅包含上面的一个边缘。在滤除无细节方面,坎尼图像确实不包含由屋瓦引起的单个边缘;在其他两幅图像中却不是这样。在坎尼图像中,关于连续性、细度和笔直度等线的质量也很出众。这种结果使得坎尼算法成为边缘检测的一种首选工具。

这里给出的几个阈值仅应在相对情形下考虑。多数算法实现会涉及各个标定步骤,如将图像的值域标定为范围[0, 1]。从这一例子可以看出,不同的标定方案明显会要求不同的阈值。

a b c d

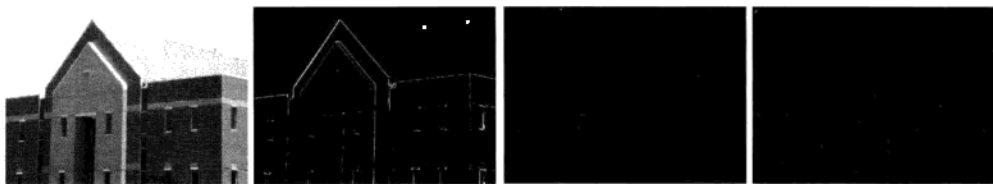


图 10.25 (a)大小为 834×1114 像素、灰度值标定为范围[0, 1]内的原图像; (b)平滑过的图像经阈值处理后得到的梯度图像; (c)使用 Marr-Hildreth 算法得到的图像; (d)使用坎尼算法得到的图像。注意,与其他两幅图像相比,坎尼图像有明显的改进

例 10.9 本节中讨论的三种主要边缘检测方法的另一说明。

作为本节中讨论的三种主要边缘检测方法的另一比较,考虑图10.26(a),它显示了一幅大小为 512×512 的头部 CT 图像。在这个例子中,我们的目的是提取大脑外轮廓(图像中的灰色区域)、脊髓区域轮廓(直对着鼻子的后边、朝着大脑前)和头部外轮廓的边缘。在消除眼部和大脑区域中与灰色有关的边缘细节时,我们希望生成尽可能最细且连续的轮廓,。

图 10.26(b)显示了一幅阈值处理后的梯度图像,它已先用一个 5×5 均值滤波器平滑过。达到所示结

果要求的阈值是该梯度图像的最大值的 15%。图 10.26(c) 显示了使用 Marr-Hildreth 边缘检测算法得到的结果, 所用的阈值为 0.002, $\sigma = 3$, 模板大小为 19×19 像素。图 10.26(d) 是使用坎尼算法得到的, 所用的 $T_L = 0.05$, $T_H = 0.15$ (低阈值的 3 倍), $\sigma = 2$, 模板大小为 13×13 , 像 Marr-Hildreth 情况那样, 它对应于大于 6σ 的最小奇整数。

在边缘质量及消除无关细节的能力方面, 图 10.26 中的结果相当接近前一例子中的结果和结论。还要注意, 坎尼算法是唯一能为大脑后边界产生完整无断裂的边缘的过程, 它也是在消除原始图像中与灰度物质有关的所有边缘时, 有可能发现最好轮廓的过程。

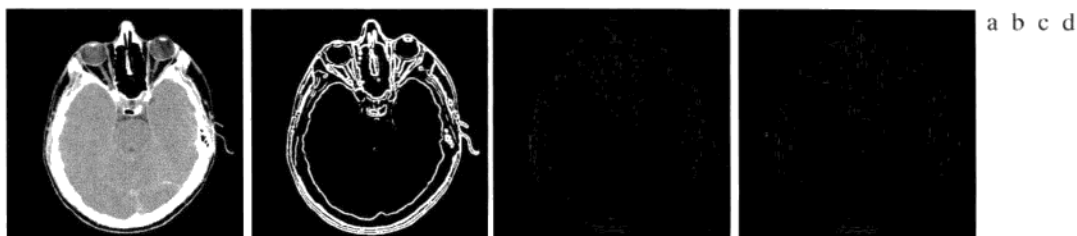


图 10.26 (a) 大小为 512×512 像素、灰度值已标定到范围 $[0, 1]$ 内的原始头部 CT 图像; (b) 平滑后的图像经阈值处理后的梯度图像; (c) 用 Marr-Hildreth 算法得到的图像; (d) 用坎尼算法得到的图像(原图像由 Vanderbilt 大学的 David R. Pickens 博士提供)

正像预期的那样, 坎尼算法改进性能所付出的代价是比早些时候讨论的两种方法实现起来更复杂, 同时执行时间会更长。在某些应用中, 如实时工业图像处理, 成本和速度需求通常要求我们使用更简单的技术, 主要是阈值梯度方法。当关注的主要是边缘质量时, 通常会使用 Marr-Hildreth 算法和坎尼算法, 特别是坎尼算法。

10.2.7 边缘连接和边界检测

理想情况下, 边缘检测应该仅产生位于边缘上的像素集合。实际上, 由于噪声、不均匀照明引起的边缘间断, 以及其他引入灰度值虚假的不连续的影响, 这些像素并不能完全描述边缘特性。因此, 一般是在边缘检测后紧接着连接算法, 所设计的算法将边缘像素组合成有意义的边缘或区域边界。在这一节中, 我们讨论三种基本的边缘连接方法, 它们是实际中使用的有代表性的技术。第一种方法需要有关局部区域中边缘点 (如一个 3×3 邻域) 的知识; 第二种方法要求区域边界上的点已知; 第三种技术是处理整个边缘图像的全局方法。

局部处理

连接边缘点最简单的方法之一是在每个点 (x, y) 处的小邻域内分析像素的特点, 该点是用前节讨论过的技术之一声明了的边缘点。根据预定的准则, 将所有的相似点连接起来, 以形成根据指定准则满足相同特性像素的一条边缘。

在这种类型的分析中, 用于确定边缘像素相似性的两个主要性质是: (1) 梯度向量的强度 (幅度); (2) 梯度向量的方向。第一个性质基于式 (10.2-10)。令 S_{xy} 表示一幅图像中以点 (x, y) 为中心的一个邻域的坐标集合。如果

$$|M(s, t) - M(x, y)| \leq E \quad (10.2-36)$$

其中 E 是一个正阈值。

梯度向量的方向角由式 (10.2-11) 给出。如果

$$|\alpha(s, t) - \alpha(x, y)| \leq A \quad (10.2-37)$$

其中 A 是一个正角度阈值, 则 S_{xy} 中, 坐标 (s, t) 处的一个边缘像素有一个与 (x, y) 处像素类似的角度。如 10.2.5 节中注释的那样, (x, y) 处的边缘的方向垂直于该点处梯度向量的方向。

如果既满足幅度准则, 也满足方向准则, 则 S_{xy} 中, 坐标为 (s, t) 的像素被连接到坐标为 (x, y) 的像素。在图像中的每个位置重复这一处理。当邻域的中心从一个像素移到另一个像素时, 必须将已连接的点记录下来。一个简单的记录过程是对每组被连接的像素分配不同的灰度值。

前面的公式计算代价很高, 因为必须检验每个点的所有邻点。一种特别适合于实时应用的一种简化由如下步骤组成:

1. 计算输入图像 $f(x, y)$ 的梯度幅度阵列 $M(x, y)$ 和梯度角度阵列 $\alpha(x, y)$ 。
2. 形成一幅二值图像 g , 任何坐标对 (x, y) 处的值由下式给出:

$$g(x, y) = \begin{cases} 1, & M(x, y) > T_M \text{ 且 } \alpha(x, y) = A \pm T_A \\ 0, & \text{其他} \end{cases}$$

其中, T_M 是一个阈值, A 是一个指定的角度方向, $\pm T_A$ 定义了一个关于 A 的可接受方向“带宽”。

3. 扫描 g 的行, 并在不超过指定长度 K 的每一行中填充(置 1)所有缝隙(0 的集合)。注意, 按照定义, 缝隙一定要限制在一个 1 或多个 1 的两端。分别地处理各行, 它们之间没有记忆。
4. 为在任何其他方向 θ 上检测缝隙, 以该角度旋转 g , 并应用步骤 3 中的水平扫描过程。然后, 将结果以 $-\theta$ 旋转回来。

当兴趣在于水平边缘连接和垂直边缘连接时, 步骤 4 就变为一个简单的过程, 在该过程中, g 被旋转 90° , 扫描各行后, 结果再被旋转回来。这是实践中最常用的方法, 如下例所示, 这种方法可以产生很好的结果。通常, 图像旋转是代价很高的计算处理, 因此, 在要求多角度方向上的连接时, 把步骤 3 和步骤 4 组合成单个放射状扫描过程更为实用。

例 10.10 使用局部处理的边缘连接。

图 10.27(a) 显示了一辆汽车尾部的图像。该例的目的是说明用前述算法来寻找大小适合车牌的矩形的应用。该矩形可以通过检测强的水平和垂直边缘构成。图 10.27(b) 显示了梯度幅度图像 $M(x, y)$, 图 10.27(c) 和 (d) 显示了该算法步骤 3 和步骤 4 的结果, 其中, 令 T_M 等于最大梯度值的 30%, $A = 90^\circ$, $T_A = 45^\circ$, 并填充了全部 25 个或更少像素的缝隙(约为图像宽度的 5%)。为检测车牌壳的全部拐角和汽车的后窗, 要求使用一个较大范围的容许角度方向。图 10.27(e) 是前两幅图像逻辑“或”(OR)操作的结果, 图 10.27(f) 是使用 9.5.5 节讨论的细化过程细化图 10.27(e) 得到的。如图 10.16(f) 所示, 在图像中清楚地检测到了对应于车牌的矩形。由于美国汽车牌照的宽高比有与众不同的 2:1 的比例, 所以利用这一事实从图像的所有矩形中简单地分离出牌照是一件简单的事情。

区域处理

通常, 图像中感兴趣区域的位置已知或可以确定。这意味着在相应的边缘图像中关于像素的区域成员的知识是可用的。在这种情况下, 我们可以使用在区域的基础上连接像素的技术, 所期望的结果是该区域边界的近似。对这类处理的一种方法是函数近似, 这里, 我们对已知点拟合一条二维曲线。典型地, 我们的兴趣在于快速执行的技术, 它可对边界的基本特性产生一个近似, 例如端点和凹点。多边形近似尤其有吸引力, 因为在保持边界表示(如多边形的顶点)相对简单的情况下, 它们可以捕捉基本形状特性。在这一节, 我们将阐述和说明一种适合这一目的算法。

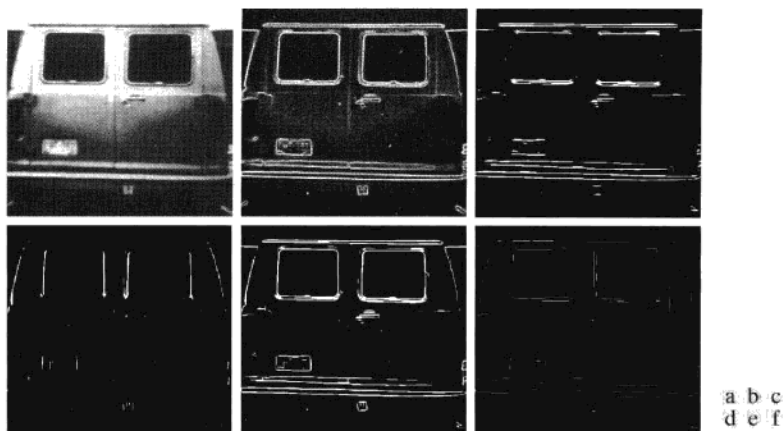


图 10.27 (a)一幅大小为 534×566 的汽车后部图像; (b)梯度幅度图像; (c)水平连接的边缘像素; (d)垂直连接的边缘像素; (e)前两幅图像的逻辑“或”(OR); (f)用形态学细化得到的最终结果(原图像由 Perceptics Corporation 提供)

在说明该算法之前, 我们用一个简单的例子来讨论该过程的机理。图 10.28 显示了表示一条非闭合曲线点的集合, 其中端点标记为 A 和 B 。按照定义, 这两个点是多边形的顶点。开始, 我们计算通过 A 点和 B 点的一条直线的参数。然后, 计算曲线上所有其他点到这条直线的垂直距离, 并选择产生最大距离的点(目的是解决任意性)。如果该距离超过了指定的阈值 T , 则将标为 C 的相应点宣布为一个顶点, 如图 10.28(c)所示。然后, 建立从 A 到 C 和从 C 到 B 的直线, 从而得到从 A 至 C 间的所有点到线 AC 的距离。如果距离超过了 T , 相应于最大距离的点被宣布为一个顶点 D ; 否则, 对于该线段就没有新的顶点要宣布。类似的过程用于 C 和 B 之间的点。图 10.28(b)显示了结果, 图 10.28(c)显示了下一步的结果。继续这一迭代过程, 直到没有点满足阈值测试为止。图 10.28(d)显示了最终的结果, 正如您看到的那样, 该结果是拟合给定点的曲线形状的一种合理近似。

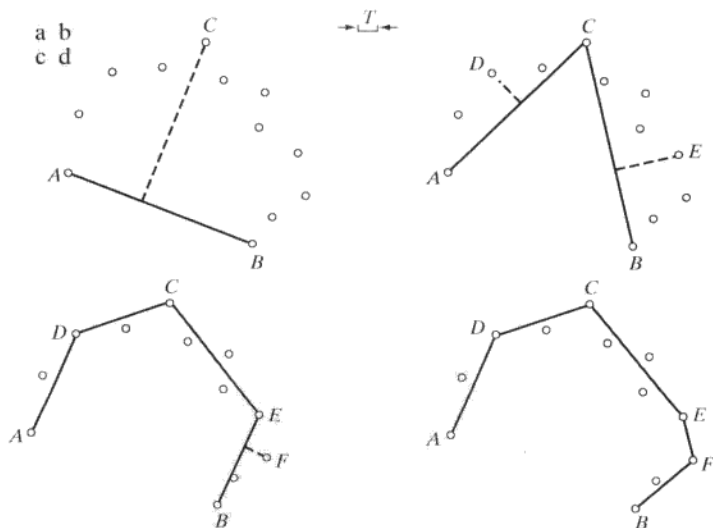


图 10.28 多边形拟合算法迭代说明

刚才解释的过程中隐含了两个重要的需求。首先,必须指定两个起始点;其次,所有的点都必须排序(即按顺时针方向或逆时针方向)。当任意一个二维点集未形成一个连通路径时(如典型的边缘图像中的情况),它并不总是很明显,无论是点属于边界线段(开放曲线),还是属于一条边界(闭合曲线)。给定已排序的点后,我们可以用分析点间的距离推断我们处理的是开放的还是闭合的曲线。当我们穿过这些点的序列时,相对于其他点的距离的两个连续点间较大的距离是开放曲线的一个好的指示。然后使用端点开始该过程。如果点间的分离趋于均匀,那么我们很可能处理的是一条闭合曲线。在这种情况下,对于选取两个起始点我们有几种选择。一种方法是选择点集中的最右边的点和最左边的点。另一种方法是寻找曲线的极端点(11.2.1 节中讨论了一种这样的方法)。寻找一个多边形拟合开放和闭合曲线的算法描述如下:

1. 令 P 是一个已排序序列,显然,这些点是一幅二值图像中的 1 值点。指定两个起始点 A 和 B 。它们是多边形的两个起始顶点。
2. 指定一个阈值 T ,以及两个空堆栈“开”(OPEN)和“闭”(CLOSED)。
3. 如果 P 中的点对应于一条闭合曲线,则把 A 放到“开”中,并把 B 放到“开”和“闭”中。如果点对应于一条开放曲线,则把 A 放到“开”中,而把 B 放到“闭”。
4. 计算从“闭”中最后一个顶点到“开”中最后一个顶点的线的参数。
5. 计算步骤 4 所得的直线至 P 中所有点的距离,序列把它们放到步骤 4 所得的两个顶点之间。选择具有最大距离 $D_{\max}$ 的点 $V_{\max}$ (解决任意性问题)。
6. 如果 $D_{\max} > T$,则把 $V_{\max}$ 作为一个新顶点放在“开”堆栈的末尾。转到步骤 4。
7. 否则,从“开”中移除最后一个顶点,并把它作为“闭”的最后一个顶点插入。
8. 如果“开”非空,转到步骤 4。
9. 否则,退出。“闭”中的顶点就是拟合 P 中的点的多边形的顶点。

关于创建已排序点序列的算法,见 11.1.1 节。

为堆栈名称使用的 OPEN 和 CLOSED 与非闭合曲线和闭合曲线无关。堆栈名称仅指出一个存储最终(CLOSED)顶点或过渡期间(OPEN)的顶点的堆栈。

下面的两个例子说明了该算法的机理。

例 10.11 用多边形近似连接边缘。

考虑图 10.29(a)中的点集 P 。假设这些点属于一条闭合曲线,且它们已按顺时针方向排序(注意有些点并不相邻), A 和 B 分别被选为 P 中的最左侧点和最右侧点。这些是起始顶点,如表 10.1 所示。选取序列中的第一个点作为最左侧的点 A 。图 10.29(b)仅显示了 A 和 B 间线段上方的点(标为 C),该点满足算法的步骤 6,因此将它指定为一个新顶点,并将它添加到“开”堆栈内的顶点中。表 10.1 的第二行显示了被检测的 C ,第三行表明它作为最后一个顶点被添加到“开”中。图 10.29(b)中的阈值 T 约等于在网格图中的一格半。

注意,在图 10.29(b)中,直线 AB 下方有一个点,该点也满足步骤 6。然而,因为这些点已被排序,故一次只能检测这两个顶点间的点的一个子集。该线段下方的其他点将在后面检测,如图 10.29(c)所示。关键是总要按给出的顺序来追踪这些点。

表 10.1 说明了导致图 10.29(h)中的解的各个步骤。图中检测到了 4 个顶点,并且该图用直线段连接这 4 个顶点形成了一个近似给定边界点的多边形。注意,尽管是按顺时针方向追踪这些点而生成的顶点,但表中检测到顶点 B, C, A, D, B 是按逆时针方向的。若输入是一条开放曲线,那么顶点将是顺时针顺序的。导致这种差异的原因是初始化“开”和“闭”堆栈的方法。对开放和闭合曲线形成的堆栈“闭”的差别也会导致闭合曲线中第一个顶点与最后一个顶点的重复。这与只给出顶点如何来区分开放和闭合多边形是一致的。

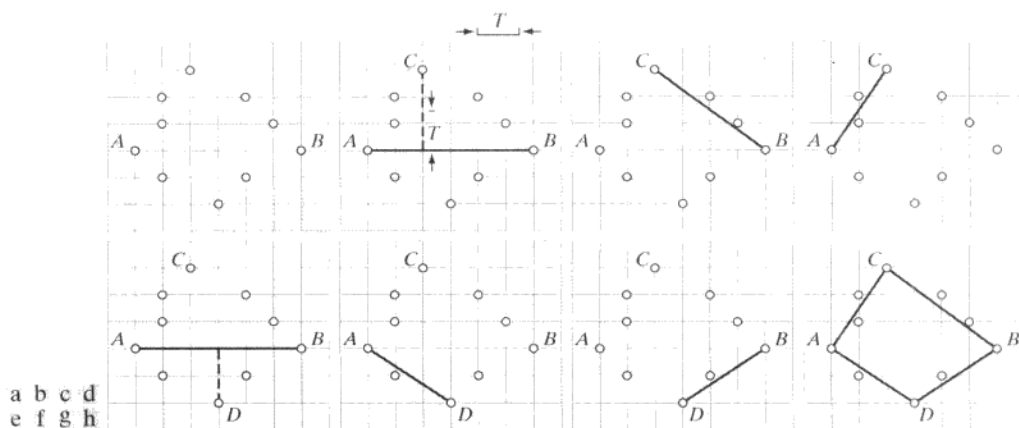


图 10.29 (a)一条顺时针路径中的点集(标为 A 和 B 的点被选为起始顶点); (b)点 C 到过 A 和 B 的直线的距离是 A 和 B 间所有点中最大的, 并且通过了阈值测试, 因此 C 是一个新顶点; (d)~(g)算法的各个进程; (h)用直线连接形成一个多边形所显示的最后顶点。表 10.1 说明了一步步的细节

表 10.1 例 10.11 机理的详细步骤

闭 合	非 闭 合	已处理的曲线段	生成的顶点
B	B, A	—	A, B
B	B, A	(BA)	C
B	B, A, C	(BC)	—
B, C	B, A	(CA)	—
B, C, A	B	(AB)	D
B, C, A	B, D	(AD)	—
B, C, A, D	B	(DB)	—
B, C, A, D, B	空	—	—

例 10.12 一幅图像的边界的多边形拟合。

图 10.30 显示了多边形拟合的一个更为实际的例子。图 10.30(a) 中的输入图像是一幅大小为 550×566 人的牙齿 X 射线图像, 其灰度已标定到区间 $[0, 1]$ 。该例的目的是提取牙齿的边界, 该方法对于运用对照数据库匹配这一领域非常有用。图 10.30(b) 是使用 Sobel 模板和阈值 $T = 0.1$ (最大灰度的 10%) 得到的梯度图像。如预期的那样, X 射线图像中的噪声成分很高, 因此, 第一步是降噪。因为该图像是二值图像, 故形态学技术可很好地适用于这一目的。图 10.30(c) 显示了滤除大多数噪声后的结果, 如果在 3×3 邻域内有 5 个或更多个像素为 1, 则置像素为 1, 否则置像素为 0。虽然噪声已减少了, 但某些噪声点仍清晰可见。图 10.30(d) 显示了形态学收缩后的结果, 它进一步将噪声降低为孤立的点。这些噪声是用例 9.4 中描述的方式使用形态学滤波消除的 [见图 10.30(e)]。在这一点上, 图像由粗边界组成, 这些边界可通过流行的形态学骨架的方法来加以细化, 如图 10.30(f) 所示。最后, 图 10.30(g) 显示了预处理中最后一步去除毛刺后的结果, 如 9.5.8 节讨论的那样。

接着, 我们用一个多边形来拟合图 10.30(g) 中的点。图 10.30(h)~(j) 显示了用多边形拟合算法的结果, 其中阈值分别等于图像宽度的 0.5%, 1% 和 2% (即 $T = 3, 6$ 和 12)。前两个结果较好地近似了边界, 但第三个处在临界状态。所有这三种情况中, 过度的锯齿清楚地表明需要进行边界平滑。图 10.30(k) 和 (l) 显示了一个一维均值模板分别与图 (j) 和图 (h) 中的边界卷积后的结果。所用的模板是元素为 1 的 1×31 阵列, 约

为对应图像宽度的 5%。正如所期望的那样,图 10.30(k)中的结果在保护重要形状特征方面仍是临界的(即右侧严重失真)。另一方面,图 10.30(l)中的结果显示了有效地边界平滑和形状特征的合理保留。例如,图像以合理的保真度保留了左上方尖点的圆度和右上方尖点的细节。

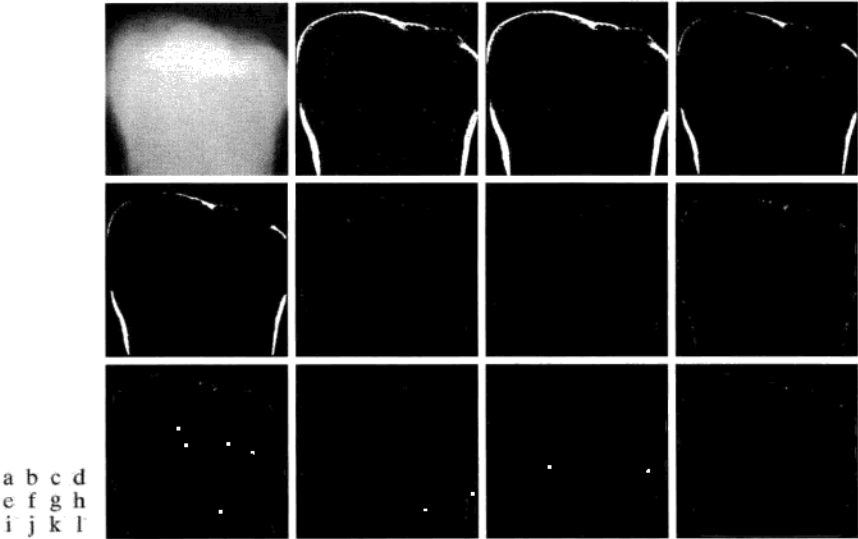


图 10.30 (a)一幅大小为 550×566 像素的人的牙齿图像;(b)梯度图像;(c)滤除大多数噪声的结果;(d)形态学收缩的结果;(e)形态学清除的结果;(f)骨架;(g)去除毛刺;(h)~(j)使用阈值约为图像宽度的 0.5%、1%和 2%(即 $T = 3, 6$ 和 12) 的多边形拟合;(k)使用一个大小为 1×31 (约为图像宽度的 5%)的一维均值滤波器平滑图(j)中边界后的结果;(l)使用相同滤波器平滑图(h)中边界后的结果

上例中的结果是使用本节讨论的多边形拟合算法所能得到的典型结果。该算法的优点是实现简单,且产生的结果通常是可以接受的。在 11.1.3 节中,我们将讨论一种更完善的过程,通过计算最小周长多边形可产生更接近的拟合。

使用霍夫变换的全局处理

前两节中讨论的方法可用于这样的场合,即属于各个目标的像素的知识至少部分可用。例如,在区域处理中,只有当我们知道它们是一个有意义区域边界的一部分时,连接给定的像素集才有意义。通常,我们不得不工作在无结构的环境中,在这种环境中,我们所拥有的一切只是一幅边缘图像,没有关于感兴趣目标可能在哪里的知识。在这种情况下,所有像素都是连接的候选点,这样,必须基于预先定义的全局性质接受或删除某些点。在这一节中,我们介绍一种基于像素集是否位于指定形状的曲线上的方法。一旦检测到,这些曲线就会形成边缘或感兴趣的区域边界。

若给定一幅图像中的 n 个点,假设我们希望找到这些点中一个位于直线上的子集。一种可行的解决方法是先找到所有由每对点确定的直线,然后寻找靠近特定直线的点的所有子集。这种方法涉及寻找 $n(n-1)/2 \sim n^2$ 条直线,然后,对于每个点,要与所有直线执行 $(n)(n(n-1))/2 \sim n^3$ 次比较。这种方法因为计算量太大而基本上没有应用价值。

霍夫[Hough 1962]提出一种替代方法,通常称为霍夫变换。考虑 xy 平面上的一个点 (x_i, y_i) 和斜截式形式为 $y_i = ax_i + b$ 的一条直线。通过点 (x_i, y_i) 的直线有无数条,且对 a 和 b 的不同值,它们都

754
~
755

满足方程 $y_i = ax_i + b$ 。然而, 将该等式写为 $b = -x_i a + y_i$, 并考虑 ab 平面(也称为参数空间), 将得到固定点 (x_i, y_i) 的单一直线的方程。此外, 第二个点 (x_j, y_j) 在参数空间中也有一条与之相关联的直线, 除非它们是平行的, 否则这条直线会与和 (x_i, y_i) 相关联的直线相交于点 (a', b') , 其中 a' 为斜率, b' 是包含 xy 平面中点 (x_i, y_i) 和点 (x_j, y_j) 的直线的截距。事实上, 这条直线上的所有点在参数空间中都有相交于点 (a', b') 的直线。图 10.31 说明了这些概念。

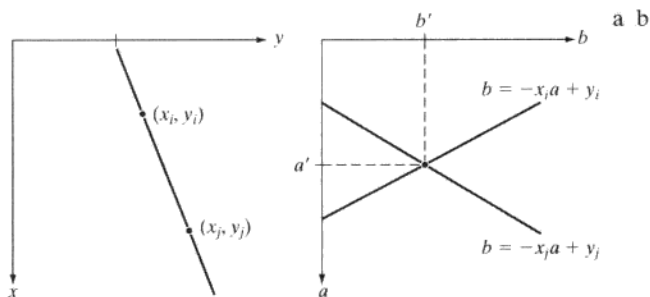


图 10.31 (a) xy 平面; (b) 参数空间

原理上, 可以画出对应于 xy 平面中所有点 (x_k, y_k) 的参数空间直线, 并且空间中的主要直线可以在参数空间中通过确定的点来找到, 大量的参数空间的线在此点处相交。然而, 这种方法的一个实际困难是, 当该直线逼近垂直方向时, a (直线的斜率) 会趋于无限大。解决该困难的方法之一是使用一条直线的法线表示:

$$x \cos \theta + y \sin \theta = \rho \quad (10.2-38)$$

图 10.32(a) 示例了参数 ρ 和 θ 的几何解释。水平直线有 $\theta = 0^\circ$, ρ 等于正的 x 截距。类似地, 垂直直线有 $\theta = 90^\circ$, ρ 等于正的 y 截距; 或者有 $\theta = -90^\circ$, ρ 等于负的 y 截距。图 10.32(b) 中的每条正弦曲线表示通过 xy 平面中一个特殊点 (x_k, y_k) 的一族直线。图 10.32(b) 中的交点 (ρ', θ') 对应于图 10.32(a) 中通过点 (x_i, y_i) 和点 (x_j, y_j) 的直线。

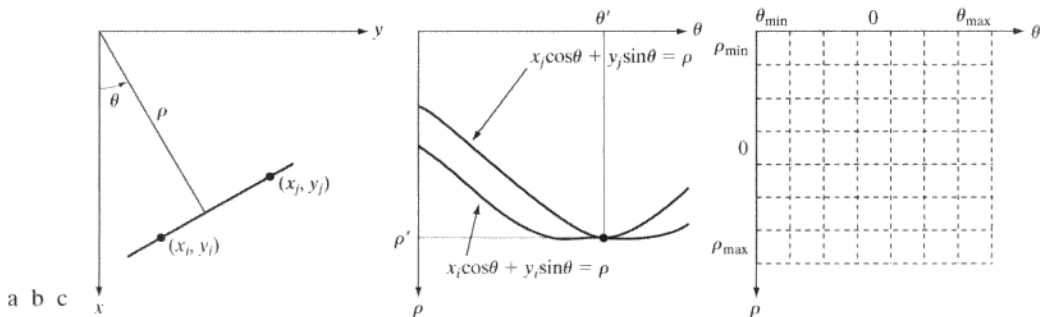


图 10.32 (a) xy 平面中直线的 (ρ, θ) 参数化; (b) $\rho\theta$ 平面中的正弦曲线, 交点 (ρ', θ') 对应于通过 xy 平面中点 (x_i, y_i) 和点 (x_j, y_j) 的直线; (c) $\rho\theta$ 平面划分为累加单元

霍夫变换计算上的魅力在于可将 $\rho\theta$ 参数空间划分为所谓的累加单元, 如图 10.32(c) 所说明的那样, 其中 $(\rho_{\min}, \rho_{\max})$ 和 $(\theta_{\min}, \theta_{\max})$ 是所期望的参数值范围: $-90^\circ \leq \theta \leq 90^\circ$ 和 $-D \leq \rho \leq D$, 其中 D 是图像中对角之间的最大距离。位于坐标 (i, j) 处的单元具有累加值 $A(i, j)$, 它对应于与参数空间坐标 (ρ_i, θ_j) 相关联的正方形。最初, 将这些单元置为零。然后, 对于 xy 平面中的每个非背景点 (x_k, y_k) , 令 θ 等于 θ

轴上每个允许的细分值,同时使用方程 $\rho = x_k \cos \theta + y_k \sin \theta$ 解出对应的 ρ 。对得到的 ρ 值进行四舍五入,得到沿 ρ 轴的最接近的允许单元值。如果选择的一个 θ_p 值得到解 ρ_q , 则令 $A(p, q) = A(p, q) + 1$ 。在这一过程结束后, $A(i, j)$ 中的值 P 将意味着 xy 平面中有 P 个点位于直线 $x \cos \theta_j + y \sin \theta_j = \rho_i$ 上。 $\rho\theta$ 平面中的细分数量决定了这些点的共线性的精确度。可以证明(见习题 10.24), 刚刚讨论的这种方法的计算次数与 xy 平面中非背景点的数量 n 成线性关系。

例 10.13 基本霍夫变换性质的说明。

图 10.33 显示了基于式(10.2-38)的霍夫变换。图 10.33(a)显示了一幅大小为 101×101 像素具有五个标记点的图像, 图 10.33(b)显示了使用 ρ 轴和 θ 轴的单位细分将每个点映射到 $\rho\theta$ 平面上的结果。 θ 的取值范围为 $\pm 90^\circ$, ρ 轴的范围为 $\pm \sqrt{2} D$, 其中 D 是图像中对角点间的距离。如图 10.33(c)所示, 每条曲线都有不同的正弦曲线形状。点 1 的映射得到的水平直线是具有零幅值的正弦曲线的一种特殊情形。

图 10.33(b)中标为 A (不要与累加值混淆)和 B 的点显示了霍夫变换的共线性检测性质。点 A 表示对应于 xy 图像平面内点 1, 3 和 5 的曲线的交点。点 A 的位置指出这三个点位于一条过原点($\rho = 0$)且方向为 45° 的直线上[见图 10.32(a)]。类似地, 在参数空间中相交于点 B 的曲线指出点 2, 3 和 4 位于方向为 -45° 且与原点的距离为 $\rho = 71$ (即从图像原点到对角的对角线距离的一半, 已四舍五入为最接近的整数值)的直线上。最后, 图 10.33(b)中标为 Q, R 和 S 的点说明了这样一个事实, 即霍夫变换展示了在参数空间左边缘和右边缘处的一种反射邻接关系。这一性质是 θ 和 ρ 在 $\pm 90^\circ$ 边界改变符号的结果。

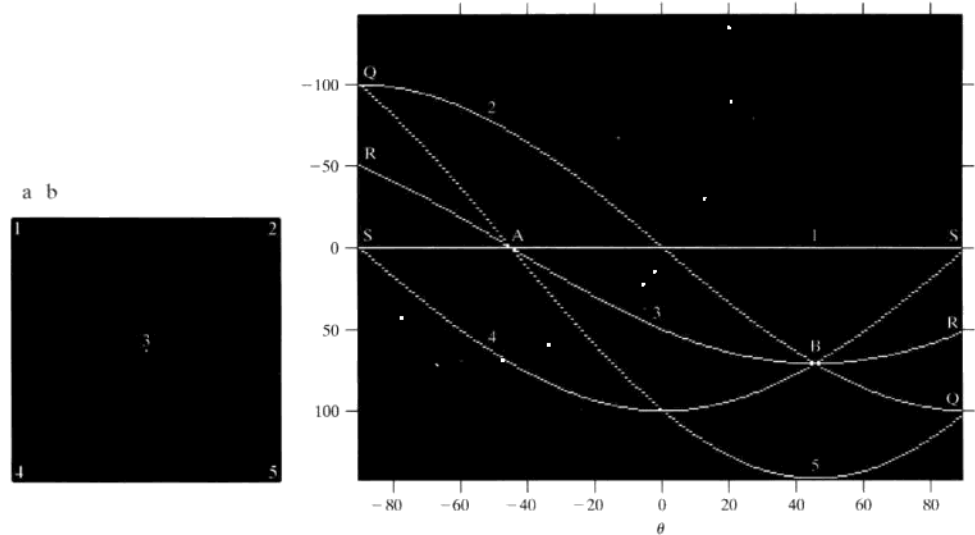


图 10.33 (a) 包含 5 个点的、大小为 101×101 像素的图像; (b) 对应的参数空间 [为便于观看, 图(a)中的点已被放大]

尽管到目前为止我们关注的重点始终在直线上, 但霍夫变换也适用于形如 $g(\mathbf{v}, \mathbf{c}) = 0$ 的任何函数, 其中 $\mathbf{v}$ 是坐标向量, $\mathbf{c}$ 是系数向量。例如, 位于圆

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2 \tag{10.2-39}$$

上的点可以使用刚才讨论的方法来检测。不同之处是存在 3 个参数(c_1, c_2 和 c_3), 在一个三维参数空间中, 这三个参数导致了类似立体的单元和形如 $A(i, j, k)$ 的累加器。该过程是增加 c_1 和 c_2 , 求出满足式(10.2-39)的 c_3 , 并更新与三元组(c_1, c_2, c_3)相关联的累加单元。很清楚, 霍夫变换的复杂性取

756
~
758

决于给定函数表达式中的坐标和系数数量。霍夫变换有可能进一步推广到检测无简单解析表达式的曲线，如变换为灰度级图像那样的应用。本章末尾包含了处理这些扩展功能的一些参考文献。

我们现在回到边缘连接问题上来。基于霍夫变换的一种连接方法如下：

1. 使用本节中先前讨论的任何技术得到一幅二值图像。
2. 指定 $\rho\theta$ 平面中的细分。
3. 对像素高度集中的地方检验其累加单元的数量。
4. 检验选中单元中像素间的关系(主要针对连续性)。

在这种情况下，连续性通常以对应于给定的累加单元计算不连续像素间的距离为基础。如果缝隙的长度比指定的阈值小，与给定单元相关联的一条直线中的缝隙则被桥接起来。注意，可能仅以方向为基础聚合直线这样的事实是一个可用于整个图像的全局概念，它仅要求我们考察与指定的累加单元相关联的像素。与前两节中讨论的方法相比，这是一个明显的优点。下例说明了这些概念。

例 10.14 使用霍夫变换连接边缘。

图 10.34(a) 显示了一幅航拍的机场图像。该例的目的是使用霍夫变换提取主要跑道的两条边。这样的问题可能很重要，例如，涉及飞行器自动导航的应用。

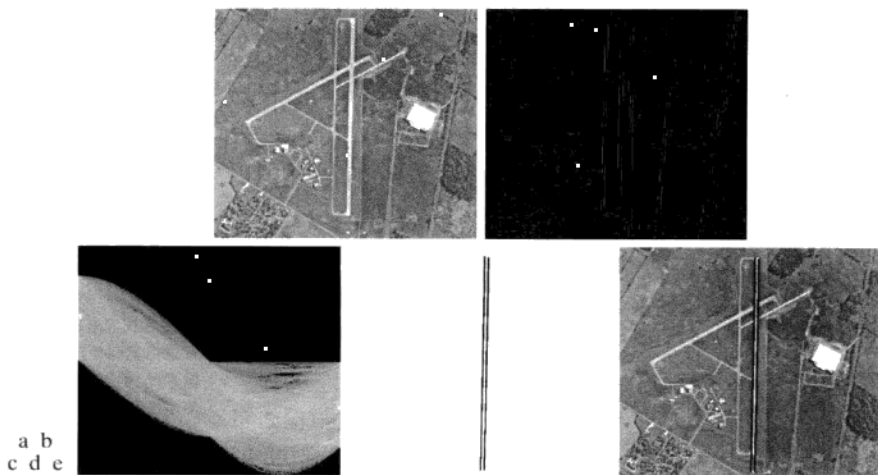


图 10.34 (a) 一幅大小为 502×564 的机场航拍图像；(b) 使用坎尼算法得到的边缘图像；(c) 霍夫参数空间(方框强调了与长垂直线相关联的点)；(d) 图像平面对应于方框所强调的点的直线；(e) 已叠加到原图像上的直线

第一步是得到一幅边缘图像。图 10.34(b) 显示了使用坎尼算法得到的边缘图像，算法中使用的参数和过程与例 10.9 中使用的相同。为计算霍夫变换，使用 10.2.5 节或 10.2.6 节中讨论的任何边缘检测技术可得到类似的结果。图 10.34(c) 显示了 θ 以 1° 递增和 ρ 以 1 个像素递增得到的霍夫参数空间。

感兴趣的跑道偏离正北方向约 1° ，所以，我们选取对应于 $\pm 90^\circ$ 并包含最高数量的单元，因为跑道在这些方向面向最长的线。图 10.34(c) 的边缘上的小白框强调了这些单元。结合图 10.33(b)，如先前提及的那样，霍夫变换展示了边缘处的邻接性。解释这一性质的另一种方法是，面向 $+90^\circ$ 的一条线和面向 -90° 的一条线是等价的(即它们是两条垂直线)。图 10.34(d) 显示了对应于刚才讨论的两个累加单元的直线，图 10.34(e) 显示了已叠加到原图像上的这些直线。这些直线是通过连接不超过图像高度的 20%(约 100 个像素)的所有间隙得到的。这些直线清楚地对应于感兴趣跑道的边缘。

注意, 解决这一问题的唯一关键知识是跑道的方向和观察者相对于跑道的位置。换句话说, 自动导航飞行器应该知道, 如果跑道朝北, 且飞行器的飞行方向也是北, 跑道会垂直地出现在图像中。其他相关的方向可采用类似的方式处理。全世界的跑道的方向在飞行图上都是可用的, 且使用 GPS (全球定位系统) 可很容易得到飞行方向的信息。该信息还可用于计算飞行器和跑道之间的距离, 这样, 允许我们估计这些参数, 例如相对于图像尺寸的直线的预期长度, 就像我们在该例中所做的那样。

10.3 阈值处理

由于阈值处理直观、实现简单且计算速度快, 因此图像阈值处理在图像分割应用中处于核心地位。在 3.1.1 节已介绍了阈值处理, 自那时起在各种讨论中我们就使用过阈值处理。在本节, 我们将以更为正式的方式讨论阈值处理, 并用比迄今为止已有的更为一般化的方式阐述该技术。

10.3.1 基础知识

在前面的章节中, 我们采用首先寻找边缘线段, 然后试图将这些线段连接为边界的方法来识别区域。在这一节中, 我们将讨论基于灰度值和/或灰度值的特性来将图像直接划分为区域的技术。

灰度阈值处理基础

假设图 10.35(a) 中的灰度直方图对应于图像 $f(x, y)$, 该图像由暗色背景上的较亮物体组成, 以这样的组成方式, 物体像素和背景像素所具有的灰度值组合成了两种支配模式。从背景中提取物体的一种明显方法是选择一个将这些模式分开的阈值 T 。然后, $f(x, y) > T$ 的任何点 (x, y) 称为一个对象点; 否则将该点称为背景点。换句话说, 分割后的图像 $g(x, y)$ 由下式给出:

$$g(x, y) = \begin{cases} 1, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases} \quad (10.3-1)$$

当 T 是一个适用于整个图像的常数时, 该公式给出的处理称为全局阈值处理。当 T 值在一幅图像上改变时, 我们使用可变阈值处理这个术语。术语局部阈值处理或区域阈值处理有时用于表示可变阈值处理, 此时, 图像中任何点 (x, y) 处的 T 值取决于 (x, y) 的邻域的特性 (例如邻域中的像素的平均灰度)。如果 T 取决于空间坐标 (x, y) 本身, 则可变阈值处理通常称为动态阈值处理或自适应阈值处理。这些术语的应用并不普遍, 在关于图像处理的文献中, 读者可能会看到它们交替地使用。

尽管我们遵从使用灰度 0 来表示背景像素而使用灰度 1 来表示物体像素的约定, 但在式(10.3-1)中也可使用任何两个明显不同的值。

图 10.35(b) 显示了一个更为困难的阈值处理问题, 它包含有三个支配模式的直方图, 例如, 这三个支配模式对应于暗色背景上的两个明亮物体。这里, 如果 $f(x, y) \leq T_1$, 则多阈值处理把点 (x, y) 分类为属于背景; 如果 $T_1 < f(x, y) \leq T_2$, 则分为一个物体; 如果 $f(x, y) > T_2$, 则为分另一个物体。即分割的图像由下式给出:

$$g(x, y) = \begin{cases} a, & f(x, y) > T_2 \\ b, & T_1 < f(x, y) \leq T_2 \\ c, & f(x, y) \leq T_1 \end{cases} \quad (10.3-2)$$

其中, a , b 和 c 是任意三个不同的灰度值。在 10.3.6 节中, 我们将讨论双阈值处理。要求两个以上阈值的分割问题很难解决 (通常是不可能的), 而较好的结果通常可用其他方法得到, 如 10.3.7 节中讨论的可变阈值处理方法或 10.4 节中讨论的区域生长方法。

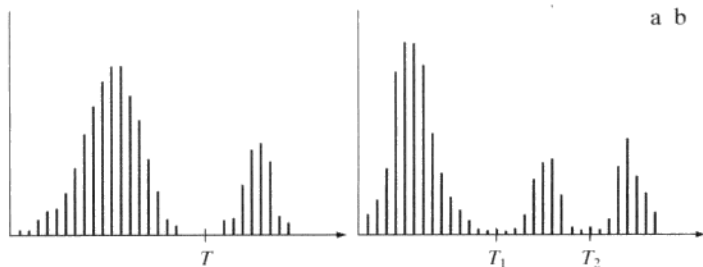


图 10.35 可被(a)单阈值和(b)双阈值分隔的灰度直方图

基于前面的讨论,我们可以凭直觉推断灰度阈值的成功与否直接关系到可区分的直方图模式的谷的宽度和深度。而影响波谷特性的关键因素是:(1)波峰间的间隔(波峰离得越远,分离这些模式的机会越好);(2)图像中的噪声内容(模式随噪声的增加而展宽);(3)物体和背景的相对尺寸;(4)光源的均匀性;(5)图像反射特性的均匀性。

图像阈值处理中噪声的作用

作为噪声如何影响一幅图像的直方图的说明,考虑图10.36(a)。这幅简单的合成图像中没有噪声,因此,其直方图由两个波峰模式组成,如图10.36(d)所示。将该图像分割为两个区域是很容易的任务,只需将一个阈值放到两个模式之间的任何位置。图10.36(b)显示了被均值为零、标准差为10个灰度级的高斯噪声污染了的原始图像。尽管相应的直方图模式现在较宽[见图10.36(e)],但它们的间隔足够大,故它们之间波谷的深度足以使得两个模式更易于分开。放在两个波峰之间的中间位置的一个阈值就可以很好地分割该图像。图10.36(c)显示了受到均值为零、标准差为50个灰度级的高斯噪声污染该图像的结果。如图10.36(f)中所示的直方图那样,现在情况是如此严重,以至于无法区分两个模式。若没有附加的处理(如10.3.4节和10.3.5节中讨论的方法),我们对于分割这样的图像寻找合适的阈值就没有什么希望。

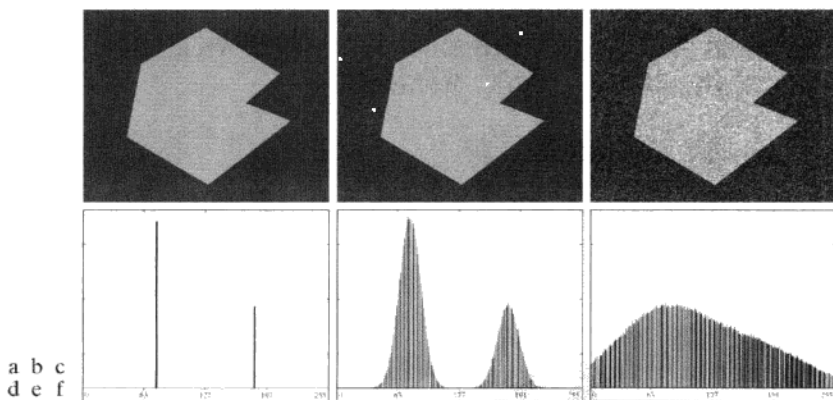


图 10.36 (a)无噪声的8比特图像;(b)带有均值为零、标准差为10个灰度级的加性高斯噪声的图像;
(c)带有均值为零、标准差为50个灰度级的加性高斯噪声的图像;(d)~(f)相应的直方图

光照和反射的作用

图10.37说明了光照对一幅图像的直方图的影响。图10.37(a)是来自图10.36(b)的噪声图像,图10.37(d)显示了它的直方图。像先前那样,使用一个单阈值很容易分割该图像。我们可以通过一个

可变的灰度函数乘以图 10.37(a)中的图像来说明非均匀光照的影响,例如,图 10.37(b)的斜坡灰度,其直方图显示在图 10.37(e)中。图 10.37(c)显示了这幅图像与该阴影模式的乘积。如图 10.37(f)所示,波峰之间的较深的波谷在模式的分离点处被污染了,如果没有附加的处理(见 10.3.4 节和 10.3.5 节),分离是不可能的。如果光照非常均匀,但图像的反射不均匀也会得到类似的结果。例如,物体的表面或背景自然反射变化的情况就是如此。

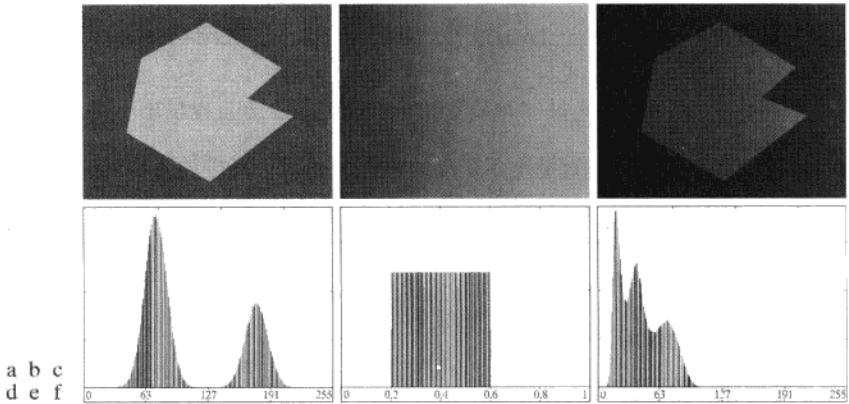


图 10.37 (a)带有噪声的图像; (b)在[0.2, 0.6]范围内的灰度斜坡图像; (c)图(a)和图(b)的乘积; (d)~(f)相应的直方图

前面章节中的关键是在使用阈值或其他分割技术的图像成功分割中,照明和反射起到了核心作用。因此,在可能的情况下,在解决分割的问题中控制这些参数应是首先考虑的问题。当不可能控制这些参数时,有三种针对该问题的基本解决方法。第一种方法是直接校正这种阴影模式。例如,非均匀(但固定的)光照可以用相反的模式与图像相乘来校正,相反的模式可以用一个恒定灰度的平坦表面成像来得到。第二种方法可通过用诸如 9.6.3 节中介绍的顶帽变换处理来尝试校正全局阴影模式。第三种方法是使用可变阈值近似处理非均匀性,如 10.3.7 节讨论的那样。

理论上,斜坡图像的直方图是均匀的。实际上,实现完美的均匀性取决于图像的尺寸和灰度比特数。例如,一幅大小为 256×256 的 256 级斜坡图像有一个均匀的直方图,但大小为 256×257 的具有相同灰度级数的斜坡图像则没有均匀的直方图。

10.3.2 基本的全局阈值处理

正像前节说明的那样,当物体和背景像素的灰度分布十分明显时,可以用适用于整个图像的单个(全局)阈值。在大多数应用中,通常图像之间有较大变化,即使全局阈值是一种合适的方法,对每一幅图像有能力自动估计阈值的算法也是需要的。下面的迭代算法可用于这一目的:

- 1. 为全局阈值 T 选择一个初始估计值。
- 2. 在式(10.3-1)中用 T 分割该图像。这将产生两组像素: G_1 由灰度值大于 T 的所有像素组成, G_2 由所有小于等于 T 的像素组成。
- 3. 对 G_1 和 G_2 的像素分别计算平均灰度值(均值) m_1 和 m_2 。
- 4. 计算一个新的阈值:

$$T = \frac{1}{2}(m_1 + m_2)$$

- 5. 重复步骤 2 到步骤 4, 直到连续迭代中的 T 值间的差小于一个预定义的参数 ΔT 为止。

当与物体和背景相关的直方图模式间存在一个相当清晰的波谷时, 这个简单的算法工作得很好。在速度是一个重要因素的情形下, 参数 ΔT 用于控制迭代的次数。通常, ΔT 越大, 则算法执行的迭代次数越少。所选的初始阈值必须大于图像中的最小灰度级而小于最大灰度级(见习题 10.28)。图像的平均灰度对于 T 来说是较好的初始选择。

例 10.15 全局阈值处理

图 10.38 显示了用前述算法以阈值估计为基础分割的例子。图 10.38(a)是原图像, 图 10.38(b)是该图像的直方图, 该直方图显示有一个明显的波谷。用 $T=m$ (平均图像灰度) 开始, 并令 $\Delta T=0$, 应用前述迭代算法经过 3 次迭代后, 得到阈值 $T=125.4$ 。图 10.38(c)显示了使用阈值 $T=125$ 来分割原图像得到的结果。如期望的那样, 从直方图清晰的模式分离中, 物体和背景间的分割相当有效。

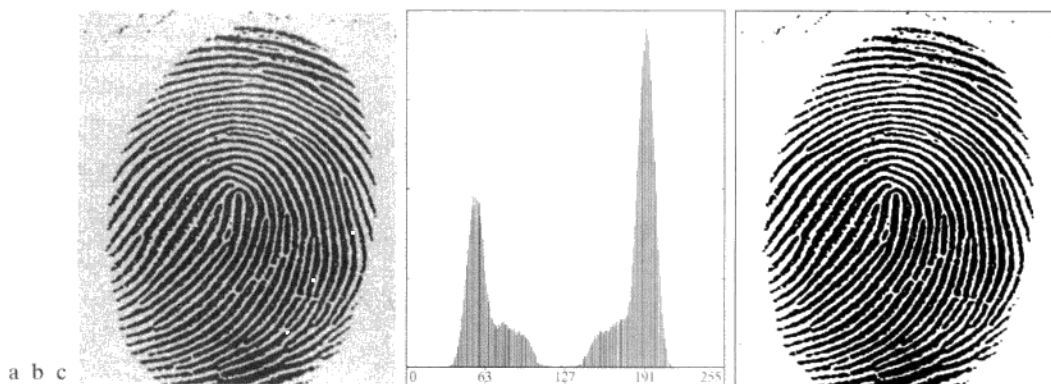


图 10.38 (a)带噪声的指纹; (b)直方图; (c)使用全局阈值对图像分割的结果(为清晰起见, 边界是加上去的)(原图像由美国国家标准和技术研究所提供)

通过对输入图像成功地进行阈值处理并在每一步计算均值说明了前述算法, 因为以这种方式来介绍该算法更为直观。然而, 有可能根据图像的直方图表达的全部计算开发一种更有效的过程, 它仅仅计算一次(见习题 10.26)。

10.3.3 用 Otsu 方法的最佳全局阈值处理

阈值处理可视为一种统计决策理论问题, 其目的是在把像素分配给两个或多个组(也称为分类)的过程中引入的平均误差最小。这一问题已知有一个精致的闭合形式的解, 称为贝叶斯决策规则(见 12.2.2 节)。该解仅以两个参数为基础: 即每一类的灰度级的概率密度函数(PDF)和在给定应用中每一类出现的概率。遗憾的是估计 PDF 并不容易, 通常采取假定一个 PDF 形式的可行方法来简化该问题, 例如, 假设它们是高斯函数。即便采用了简化, 用这些假设求解的过程也可能是复杂的, 并且对实际应用也不总是合适的。

本节中讨论的称为 Otsu 方法(Otsu[1979])是另一种有吸引力的方案。该方法在类间方差最大的情况下是最佳的, 即众所周知的统计鉴别分析中所用的度量。基本概念是好阈值分类就它的像素灰度值而论应该是截然不同的, 反过来说, 就它的灰度值而言给出最好的类间分离的阈值就是最好的(最佳的)阈值。除了其最佳性之外, Otsu 方法有一个重要的特性, 即它完全以在一幅图像的直方图上执行计算为基础, 直方图是很容易得到的一维阵列。

令 $\{0, 1, 2, \dots, L-1\}$ 表示一幅大小为 $M \times N$ 像素的数字图像中的 L 个不同的灰度级, n_i 表示灰度级为 i 的像素数。图像中的像素总数 MN 为 $MN = n_0 + n_1 + n_2 + \dots + n_{L-1}$ 。归一化的直方图(见 3.3 节)具有分

量 $p_i = n_i / MN$ ，由此有

$$\sum_{i=0}^{L-1} p_i = 1, \quad p_i \geq 0 \quad (10.3-3)$$

现在，假设我们选择一个阈值 $T(k) = k, 0 < k < L-1$ ，并使用它把输入图像阈值化处理为两类 C_1 和 C_2 ，其中， C_1 由图像中灰度值在范围 $[0, k]$ 内的所有像素组成， C_2 由灰度值在范围 $[k+1, L-1]$ 内的所有像素组成。用该阈值，像素被分到类 C_1 中的概率 $P_1(k)$ 由如下的累积和给出：

$$P_1(k) = \sum_{i=0}^k p_i \quad (10.3-4)$$

换一个角度看，这是类 C_1 发生的概率。例如，如果我们置 $k=0$ ，则任何像素分到类 C_1 中的概率为零。类似地，类 C_2 发生的概率为

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k) \quad (10.3-5)$$

由式(3.3-18)，分配到类 C_1 的像素的平均灰度值为

$$\begin{aligned} m_1(k) &= \sum_{i=0}^k iP(i/C_1) \\ &= \sum_{i=0}^k iP(C_1/i)P(i)/P(C_1) \\ &= \frac{1}{P_1(k)} \sum_{i=0}^k ip_i \end{aligned} \quad (10.3-6)$$

其中， $P_1(k)$ 由式(10.3-4)给出。式(10.3-6)第一行中的 $P(i/C_1)$ 项是值 i 的概率，已知 i 来自类 C_1 。该式中的第二行来自贝叶斯公式：

$$P(A/B) = P(B/A)P(A)/P(B)$$

第三行遵循这样一个事实，即给定 i 的 C_1 的概率 $P(C_1/i)$ 为 1，因为我们只处理来自类 C_1 的 i 值。此外， $P(i)$ 是第 i 个值的概率，很简单，它是直方图 $p(i)$ 的第 i 个分量。最后， $P(C_1)$ 是类 C_1 的概率，由式(10.3-4)我们知道它等于 $P_1(k)$ 。

类似地，分配到类 C_2 中的像素的平均灰度值为

$$\begin{aligned} m_2(k) &= \sum_{i=k+1}^{L-1} iP(i/C_2) \\ &= \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i \end{aligned} \quad (10.3-7)$$

直至级 k 的累加均值由下式给出：

$$m(k) = \sum_{i=0}^k ip_i \quad (10.3-8)$$

而整个图像的平均灰度(即全局均值)由下式给出：

$$m_G = \sum_{i=0}^{L-1} ip_i \quad (10.3-9)$$

下面两个公式的正确性可通过直接代入前面的结果来验证:

$$P_1 m_1 + P_2 m_2 = m_G \quad (10.3-10)$$

和

$$P_1 + P_2 = 1 \quad (10.3-11)$$

其中, 为有利于清楚地说明, 我们暂时忽略了 k 项。

为了评价级别 k 处的阈值的“质量”, 我们使用归一化的无量纲矩阵:

$$\eta = \frac{\sigma_B^2}{\sigma_G^2} \quad (10.3-12)$$

其中, σ_G^2 是全局方差 [即图像中所有像素的灰度方差, 如式 (3.3-19) 给出的那样],

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i \quad (10.3-13)$$

σ_B^2 为类间方差, 它定义为

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \quad (10.3-14)$$

该表达式还可写为

$$\begin{aligned} \sigma_B^2 &= P_1 P_2 (m_1 - m_2)^2 \\ &= \frac{(m_G P_1 - m)^2}{P_1(1 - P_1)} \end{aligned} \quad (10.3-15)$$

其中, m_G 和 m 如先前说明的那样。该式的第一行来自式 (10.3-14)、式 (10.3-10) 和式 (10.3-11)。第二行来自式 (10.3-5) 到式 (10.3-9)。这种形式在计算上稍微有效一些, 因为全局均值 m_G 仅计算一次, 故对 k 的任何值仅需要计算两个参数 m 和 P_1 。

式(10.3-15)的第二步仅在 P_1 大于0且小于1时才有意义, 鉴于式(10.3-11), 这表明 P_2 必须满足相同的条件。

从式 (10.3-15) 的第一行我们可以看出, 两个均值 m_1 和 m_2 彼此隔得越远, σ_B^2 越大, 这表明类间方差是类之间的可分性度量。因为 σ_G^2 是一个常数, 由此得出 η 也是一个可分性度量, 且最大化这一度量等价于最大化 σ_B^2 。然后, 目标是确定阈值 k , 它最大化类间方差, 就像本节开始时说明的那样。注意, 式 (10.3-12) 隐含假设了 $\sigma_G^2 > 0$ 。仅当图像中的所有灰度级相同时, 这一方差才为零, 这意味着仅存在一类像素。同样, 这也意味着对于常数图像有 $\eta = 0$, 因为来自其自身单个类的可分性为零。

再次引入 k , 我们有最终结果:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \quad (10.3-16)$$

和

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \quad (10.3-17)$$

从而最佳阈值是 k^* , 其最大化 $\sigma_B^2(k)$:

$$\sigma_B^2(k^*) = \max_{0 \leq k \leq L-1} \sigma_B^2(k) \quad (10.3-18)$$

换句话说, 为寻找 k^* , 我们对 k 的所有整数 [保持上述 $0 < P_1(k) < 1$ 的条件] 简单地计算式 (10.3-18), 并选取使得 $\sigma_B^2(k)$ 最大的 k 值。如果 $\sigma_B^2(k)$ 的最大值对应于多个 k 值, 习惯上是对于使 $\sigma_B^2(k)$ 最大的

k 的各个值做平均。可以证明(见习题 10.33), 在 $0 < P_1(k) < 1$ 的条件下, 总存在一个最大值。对于所有的 k 值, 计算式(10.3-17)和式(10.3-18)的代价相对较小, 因为 k 所具有的整数值的最大数量是 L 。

一旦得到 k^* , 输入图像就可像之前那样进行分割:

$$g(x, y) = \begin{cases} 1, & f(x, y) > k^* \\ 0, & f(x, y) \leq k^* \end{cases} \quad (10.3-19)$$

其中, $x = 0, 1, 2, \dots, M-1$ 和 $y = 0, 1, 2, \dots, N-1$ 。注意, 仅仅用 $f(x, y)$ 的直方图就可以得到计算式(10.3-17)所需的所有参量。除了最佳阈值外, 与分割图像有关的其他信息可从直方图中提取。例如, 在最佳阈值处计算出的类的概率 $P_1(k^*)$ 和 $P_2(k^*)$, 指出了经阈值处理后的图像中由该类(像素组)所占据的面积部分。类似地, 均值 $m_1(k^*)$ 和 $m_2(k^*)$ 是原图像中类的平均灰度的估计。

在最佳阈值处计算的归一化度量 η , 即 $\eta(k^*)$, 可用于得到类别可分性的定量估计, 同时给出了很容易对一幅给定图像进行阈值处理的概念。这一度量的值域为

$$0 \leq \eta(k^*) \leq 1 \quad (10.3-20)$$

如先前提到的那样, 仅由单一且恒定灰度级的图像就能得到下界, 且仅由灰度等于 0 和 $L-1$ 的二值图像就能得到上界(见习题 10.34)。

Otsu 算法小结如下:

1. 计算输入图像的归一化直方图。使用 $p_i, i = 0, 1, 2, \dots, L-1$ 表示该直方图的各个分量。
2. 用式(10.3-4), 对于 $k = 0, 1, 2, \dots, L-1$, 计算累积和 $P_1(k)$ 。
3. 用式(10.3-8), 对于 $k = 0, 1, 2, \dots, L-1$, 计算累积均值 $m(k)$ 。
4. 用式(10.3-9)计算全局灰度均值 m_G 。
5. 用式(10.3-17), 对于 $k = 0, 1, 2, \dots, L-1$, 计算类间方差 $\sigma_B^2(k)$ 。
6. 得到 Otsu 阈值 k^* , 即使得 $\sigma_B^2(k)$ 最大的 k 值。如果最大值不唯一, 用相应检测到的各个最大值 k 的平均得到 k^* 。
7. 在 $k = k^*$ 处计算式(10.3-16), 得到可分性度量 η^* 。

下面的例子说明了前述概念。

例 10.16 使用 Otsu 方法的最佳全局阈值处理。

图 10.39(a)显示了聚合细胞的光学显微镜图像, 图 10.39(b)显示了其直方图。该例子的目的是从背景中分割出分子。图 10.39(c)是使用前节给出的基本全局阈值处理算法得到的结果。因为直方图没有明显的波谷, 且背景和物体间的灰度差别很小, 所以算法未完成期望的分割。图 10.39(d)显示了使用 Otsu 方法得到的结果。该结果明显好于图 10.39(c)中的结果。使用基本算法计算出的阈值是 169, 而使用 Otsu 方法计算出的阈值是 181, 后者更接近图像中定义为细胞的较亮区域。可分性度量 η 是 0.467。

聚合细胞是使用聚合物人工生成的细胞。人眼看不到聚合细胞, 这种细胞可用于向人体内的目标区域输送药物。

有趣的是, 将 Otsu 方法应用于例 10.15 中的指纹图像, 得到的阈值和可分性度量将分别为 125 和 0.944。该阈值与使用基本算法得到的阈值(已四舍五入为最接近的整数)大小相同。对给定该直方图的特性, 这一结果并不意外。事实上, 由于原来的模式间的分离度相对较大, 并且它们之间的波谷较深, 所以可分性度量较高。

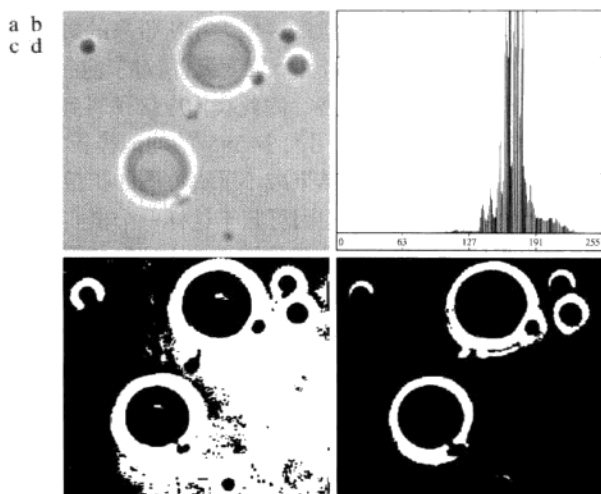


图 10.39 (a)原图像; (b)直方图(为突出低值中的细节, 高峰已被裁剪); (c)用 10.3.2 节的基本全局算法得到的分割结果; (d)用 Otsu 方法得到的结果(原图像由宾夕法尼亚大学的 Daniel A. Hammer 教授提供)

10.3.4 用图像平滑改善全局阈值处理

正如在图 10.36 中解释的那样, 噪声可将一个简单的阈值处理问题变为一个不可解决问题。当噪声不能在源头减少, 并且阈值处理又是所选择的分割方法时, 那么通常可增强性能的一种技术是在阈值处理之前平滑图像。我们用一个例子来说明这一方法。

图 10.40(a)是来自图 10.36(c)的图像, 图 10.40(b)显示了其直方图, 图 10.40(c)是使用 Otsu 方法对图像进行阈值处理后的结果。白色区域中的每个黑点和黑色区域中的每个白点是阈值处理的误差, 所以分割很不成功。图 10.40(d)显示了使用一个大小为 5×5 (图像大小为 651×814 像素) 的均值模板平滑噪声图像后的结果, 图 10.40(e)是其直方图。由于平滑处理, 直方图形状的改进很明显, 因此我们预料对平滑后的图像进行阈值处理所得到的结果将近于完美。如图 10.40(f)所示, 事实的确如此。经平滑和分割后的图像中, 物体和背景间的边界稍微有点失真是由于对边界的模糊造成的。事实上, 我们应该预料到, 我们平滑一幅图像侵蚀得越多, 分割后的结果中的边界误差就越大。

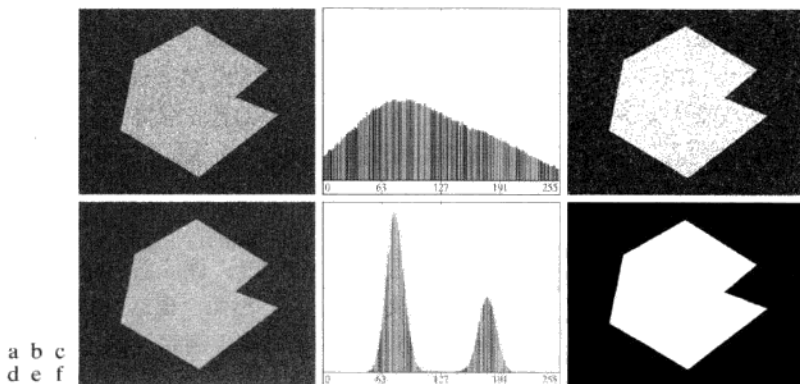


图 10.40 (a)来自图 10.36 的噪声图像; (b)该图像的直方图; (c)用 Otsu 方法得到的结果; (d)用一个大小为 5×5 的均值模板平滑噪声图像的结果; (e)结果图像的直方图; (f)用 Otsu 方法阈值处理后的结果

下面我们考虑降低图 10.40(a)中相对于背景的区域大小所带来的影响。图 10.41(a)显示了该结果。这幅图像中的噪声是均值为零、标准差为 10 个灰度级(相对于前一例子中的 50)的加性高斯噪声。如图 10.41(b)所示,直方图没有清晰的波谷,因此我们应该预料到分割会失败,图 10.41(c)中的结果确认了这一事实。图 10.41(d)显示了使用大小 5×5 的均值模板对图像进行平滑后的结果,图 10.41(e)是其直方图。如预料的那样,最终结果减小了直方图的扩散,但分布仍是单峰形式。如图 10.41(f)所示,分割再一次失败了。失败的原因可归于这样一个事实,即区域太小,以至于该区域对直方图的贡献与由噪声引起的灰度扩散相比无足轻重。在这种情形下,下一节讨论的方法可能更成功。

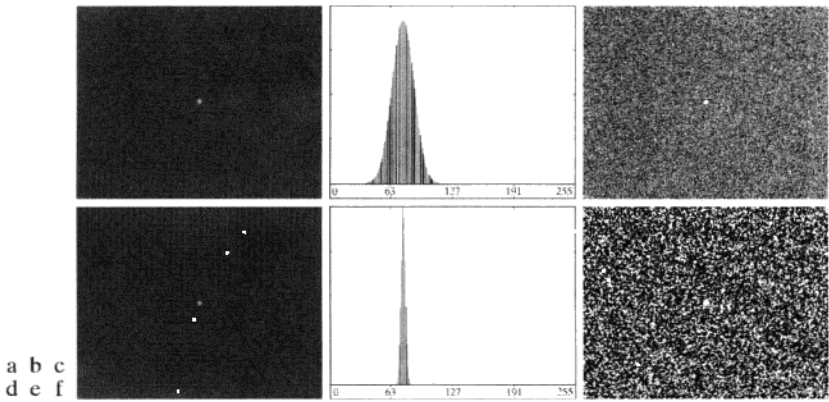


图 10.41 (a)噪声图像; (b)其直方图; (c)用 Otsu 方法得到的结果; (d)用大小为 5×5 的均值模板平滑噪声图像后的结果; (e)结果图像的直方图; (f)用 Otsu 方法对图像进行阈值处理后的结果。两种情形下, 阈值处理均失败

10.3.5 利用边缘改进全局阈值处理

基于前四节中的讨论,我们得出如下结论:如果直方图的波峰是高、窄、对称的,且被深的波谷分开,则选取一个“较好”阈值的机会是相当大的。改进直方图形状的一种方法是仅考虑那些位于或靠近物体和背景间的边缘的像素。一种直接且明显的改进是直方图很少依赖物体和背景的相对大小。例如,大背景区域上一个小物体(或相反)组成的图像的直方图将会有有一个较大的波峰,因为一种类型的像素高度集中。在前一节中我们已了解到,这可能会导致阈值处理失败。

如果仅用位于或接近物体和背景间的边缘的像素,则得到的直方图将有几个高度近似相同的波峰。此外,像素位于物体上的概率将近似等于其位于背景中的概率,从而改进了直方图模式的对称性。最后,如下段中指出的那样,用满足某些基于梯度和拉普拉斯算子的简单度量的像素有加深直方图波峰间的波谷的倾向。

刚才讨论的方法假设物体和背景间的边缘是已知的。很明显,该信息在分割期间是不可用的,因为寻找物体和背景间的分界线正好也是分割所要做的工作。然而,根据 10.2 节中的讨论,一个像素是否位于边缘上的迹象可通过计算其梯度或拉普拉斯来确定。例如,在一个边缘的过渡处,拉普拉斯的平均值为零(见图 10.10),因此,根据拉普拉斯准则选取的像素所形成的直方图的波谷应该很少。这一特性往往会产生上面讨论的较深波谷。实际上,无论是使用梯度图像还是使用拉普拉斯图像得到的典型结果相比较,后者比较有利,因为它在计算上更有吸引力,同时也是一个各向同性边缘检测器。

770
~
772

前述讨论可小结为如下算法, 其中 $f(x, y)$ 是输入图像:

1. 用 10.2 节讨论的任何一种方法来计算一幅边缘图像, 无论是 $f(x, y)$ 梯度的幅度还是拉普拉斯的绝对值都可以。
2. 指定一个阈值 T 。
3. 用步骤 2 中的阈值对步骤 1 中的图像进行阈值处理, 产生一幅二值图像 $g_T(x, y)$ 。在从 $f(x, y)$ 中选取对应于“强”边缘像素的下一步中, 该图像用做一幅模板图像。
4. 仅用 $f(x, y)$ 中对应于 $g_T(x, y)$ 中像素值为 1 的位置的像素计算直方图。
5. 用步骤 4 中的直方图全局地分割 $f(x, y)$, 例如使用 Otsu 方法。

有可能修改该算法以便同时使用梯度图像的幅度和拉普拉斯图像的绝对值。在这种情况下, 我们会为每幅图像指定一个阈值, 并求两个结果的逻辑“或”(OR), 以得到标记图像。当需要对有效的边缘点施加更多的控制时, 这种方法很有用。

如果将 T 设置为小于边缘图像的最小值的任何值, 根据式 (10.3-1), $g_T(x, y)$ 将由全 1 组成, 这意味着 $f(x, y)$ 的所有像素将用于计算图像的直方图。在这种情况下, 前述算法变成使用原始图像的直方图的全局阈值处理。习惯上以百分比的形式来指定 T 值, 百分比通常设置得较高 (如 90%), 以便在计算中用梯度/拉普拉斯图像中的较少像素。下面的例子说明了刚才讨论的概念。第一个例子使用梯度, 第二个例子使用拉普拉斯。在两个例子中, 无论使用哪一种方法都可得到类似的结果。重要的问题是生成一幅合适的导数图像。

第 n 个百分比是大于给定集合内所有数字的 $n\%$ 的最小数字。例如, 如果读者在一次测验中得到了 95 分, 且该分数大于参加此次测验的所有学生中的 85% 的学生得到的分数, 则读者在测试分数中位于第 85 个百分点处。

例 10.17 用以梯度为基础的边缘信息改进全局阈值处理。

图 10.42(a) 和 (b) 显示了来自图 10.41 的图像和直方图。您可以看到, 这幅图像不能用平滑后紧跟着阈值处理来分割。这个例子的目的是解决利用边缘信息的问题。图 10.42(c) 是在 99.7% 处经阈值处理后的梯度幅度图像。图 10.42(d) 是用输入图像与这个 (模板) 图像相乘形成的图像。图 10.42(e) 是图 10.42(d) 中非零元素的直方图。注意, 该直方图具有先前讨论的重要特性, 即它有被一个较深的波谷分开的对称模式。这样, 尽管原始带噪声图像的直方图未提供成功进行阈值处理的希望, 但图 10.42(e) 中的直方图表明, 从背景中提取出小物体的阈值处理的确是可能的。图 10.42(f) 中的结果表明事实的确如此。这幅图像是这样得到的: 以图 10.42(e) 的直方图为基础, 用 Otsu 方法得到一个阈值, 然后将该阈值全局地应用到图 10.42(a) 中的带噪声图像。其结果近于完美。

例 10.18 用拉普拉斯为基础的边缘信息改进全局阈值处理。

在这个例子中, 我们考虑一个更为复杂的阈值处理问题。图 10.43(a) 显示了一幅酵母细胞的 8 比特图像, 其中, 我们希望用全局阈值处理得到相对于亮点的区域。首先, 图 10.43(b) 显示了该图像的直方图, 图 10.43(c) 是把 Otsu 方法直接用于所示的直方图的图像上得到的结果。我们看到, Otsu 方法未能实现检测亮点这一原始目标, 并且, 尽管该方法能隔离出某些细胞区域本身, 但右侧的几个分割后的区域却没有分开。Otsu 方法计算出的阈值是 42, 可分性度量是 0.636。

图 10.43(d) 显示了图像 $g_T(x, y)$, 该图像是按如下方式得到的: 先计算拉普拉斯图像的绝对值, 然后用范围 $[0, 255]$ 内的灰度级 115 作为 T 值进行阈值处理。这个 T 值近似对应拉普拉斯图像绝对值中的 99.5%, 所以在这个水平上阈值处理应该得到一个稀疏的像素集合, 如图 10.43(d) 所示。注意, 在该图像中, 很多点聚集在亮点的边缘附近, 这正如从前面的讨论所预料的那样。图 10.43(e) 是图 (a) 和图 (d) 之积的非零像素的直方图。最后, 图 10.43(f) 显示了以图 10.43(e) 中的直方图为基础, 采用 Otsu 方法全局地分割原图像后的结果。该结果与图像中亮点的位置一致。用 Otsu 方法计算出的阈值是 115, 可分性度量是 0.762, 这两个值都要高于使用原始直方图得到的值。

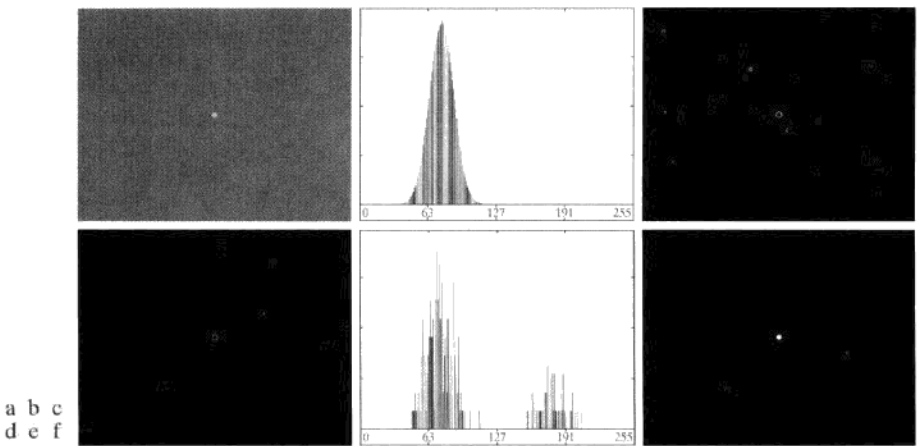


图 10.42 (a) 来自图 10.41 (a) 的带噪声图像; (b) 该图像的直方图; (c) 在 99.7% 处经阈值处理后的梯度幅度图像; (d) 图 (a) 和图 (c) 之积形成的图像; (e) 图 (d) 所示图像中非零像素的直方图; (f) 以直方图 (e) 为基础, 用 Otsu 阈值处理分割图像的结果。阈值为 134, 在该直方图中处在两个波峰间的近似中间的位置

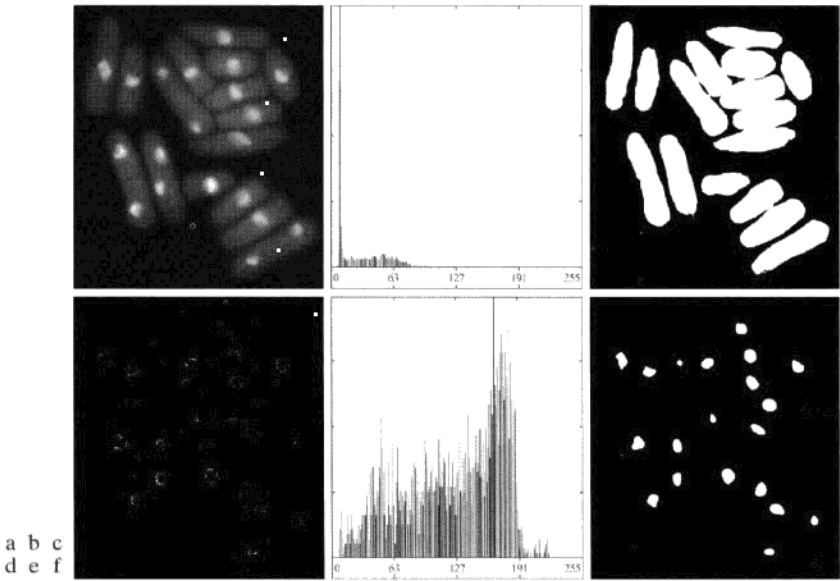


图 10.43 (a) 酵母细胞图像; (b) 图 (a) 的直方图; (c) 使用图 (b) 中的直方图, 用 Otsu 方法对图 (a) 进行分割; (d) 经阈值处理后的绝对拉普拉斯图像; (e) 图 (a) 和图 (d) 乘积中的非零像素的直方图; (f) 以 (e) 中的直方图为基础, 使用 Otsu 的方法对原图像进行阈值处理后的结果 (原图像由南加利福尼亚大学的 Susan L. Forsburg 教授提供)

通过在可变的百分比处设置阈值, 我们甚至还可以改善细胞区域的分割。例如, 图 10.44 显示了使用前段中的相同步骤得到的结果, 但阈值设置在 55 处, 它近似拉普拉斯图像最大绝对值的 5%。该值位于图像中的所有值的 53.9%。很明显, 这个结果好于图 10.43 (c) 所示对原始图像的直方图用 Otsu 的方法得到的结果。

774
~
775

10.3.6 多阈值处理

迄今为止,我们关注的是用单一全局阈值对图像进行分割。我们可将10.3.3节中介绍的阈值处理方法扩展到任意数量的阈值,因为以这种方法为基础的可分性度量也可以扩展到任意数量的分类(Fukunaga[1972])。在 K 个类 $C_1, C_2, \dots, C_K$ 的情况下,类间方差可归纳为下面的表达式:

$$\sigma_B^2 = \sum_{k=1}^K P_k (m_k - m_G)^2 \quad (10.3-21)$$

其中,

$$P_k = \sum_{i \in C_k} p_i \quad (10.3-22)$$

$$m_k = \frac{1}{P_k} \sum_{i \in C_k} ip_i \quad (10.3-23)$$

并且, m_G 是由式(10.3-9)给出的全局均值。 K 类由 $K-1$ 个阈值来分离,这些值 $k_1^*, k_2^*, \dots, k_{K-1}^*$ 是式(10.3-21)的最大值:

$$\sigma_B^2(k_1^*, k_2^*, \dots, k_{K-1}^*) = \max_{0 < k_1 < k_2 < \dots < k_{K-1} < L-1} \sigma_B^2(k_1, k_2, \dots, k_{K-1}) \quad (10.3-24)$$

虽然这个结果通常很完美,但当分类数量增加时它会开始失去意义,因为我们仅处理一个变量(灰度)。事实上,通常类间方差是依据以向量描述的多个变量来计算的(Fukunaga[1972])。实际中,当我们有理由相信使用两个阈值可以有效地解决问题时,那么使用多个全局阈值处理就被视为一种可行的方法。要求两个以上阈值的应用,通常使用更多的灰度值来解决。而该方法使用的是附加的描述子(譬如彩色),并且这种应用是以如10.3.8节描述的模式识别问题来筹划的。

对于由三个灰度间隔组成的三个类(这三个类由两个阈值分隔),类间方差由下式给出:

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + P_3(m_3 - m_G)^2 \quad (10.3-25)$$

其中,

$$P_1 = \sum_{i=0}^{k_1} p_i, \quad P_2 = \sum_{i=k_1+1}^{k_2} p_i, \quad P_3 = \sum_{i=k_2+1}^{L-1} p_i \quad (10.3-26)$$

和

$$m_1 = \frac{1}{P_1} \sum_{i=0}^{k_1} ip_i, \quad m_2 = \frac{1}{P_2} \sum_{i=k_1+1}^{k_2} ip_i, \quad m_3 = \frac{1}{P_3} \sum_{i=k_2+1}^{L-1} ip_i \quad (10.3-27)$$

如在式(10.3-10)和式(10.3-11)中那样,如下关系成立:

$$P_1 m_1 + P_2 m_2 + P_3 m_3 = m_G \quad (10.3-28)$$

和

$$P_1 + P_2 + P_3 = 1 \quad (10.3-29)$$

因此,我们看到, P 项、 m 项,以及 σ_B^2 都是 k_1 和 k_2 的函数。两个最佳阈值 k_1^* 和 k_2^* 是使得 $\sigma_B^2(k_1, k_2)$ 最



图 10.44 使用图 10.43 (d)~(f) 中解释的相同过程,对图 10.43 (a) 中的图像进行分割后的结果,但对绝对拉普拉斯图像进行阈值处理时,使用了一个较低的阈值

带有两个阈值的阈值处理有时也称为滞后阈值处理。

大的值。换句话说,如在 10.3.3 节讨论的单阈值情况那样,我们用下式寻找最佳阈值:

$$\sigma_B^2(k_1^*, k_2^*) = \max_{0 < k_1 < k_2 < L-1} \sigma_B^2(k_1, k_2) \quad (10.3-30)$$

该过程由选择第一个 k_1 值开始(该值是 1, 因为在 0 灰度处寻找阈值没有意义;还要记住,增量值为整数,因为我们处理的是灰度)。接着, k_2 的所有值在大于 k_1 和小于 $L-1$ 的范围内增加(即 $k_2 = k_1 + 1, \dots, L-2$)。然后,将 k_1 增大到其下一个值, k_2 的所有值再次在大于 k_1 的所有值范围内增加。重复该过程,直到 $k_1 = L-3$ 为止。该处理的结果是一个二维阵列 $\sigma_B^2(k_1, k_2)$, 最后一步是在该阵列中寻找最大值。对应于最大值的 k_1 值和 k_2 值就是最佳阈值 k_1^* 和 k_2^* 。如果存在几个最大值,则对应于 k_1 和 k_2 的值被平均以得到最终的阈值。然后,阈值处理后的图像由下式给出:

$$g(x, y) = \begin{cases} a, & f(x, y) \leq k_1^* \\ b, & k_1^* < f(x, y) \leq k_2^* \\ c, & f(x, y) > k_2^* \end{cases} \quad (10.3-31)$$

其中, a, b 和 c 是任意三个有效的灰度值。

最后,我们注意到,在 10.3.3 节中为单一阈值定义的可分性度量可直接扩展到多个阈值:

$$\eta(k_1^*, k_2^*) = \frac{\sigma_B^2(k_1^*, k_2^*)}{\sigma_G^2} \quad (10.3-32)$$

其中, σ_G^2 是来自式(10.3-13)的总图像方差。

例 10.19 多个全局阈值处理。

图 10.45(a)显示了一幅冰山的图像。这个例子的目的是把图像分割成三个区域:暗背景、冰山的明亮区域和阴影区域。由图 10.45(b)中的图像直方图可以看出,解决这一问题需要两个阈值是很明显的。按上面讨论的过程得到阈值 $k_1^* = 80$ 和 $k_2^* = 177$, 由图 10.45(b)注意到,它们靠近两个直方图波谷的中心。图 10.45(c)是在式(10.3-31)中使用这两个阈值得到的分割。可分性度量是 0.954。这个例子能做到这样好的主要原因在于能寻找到三个具有明显可分模式的直方图,该图有适度宽度和深度的波谷。

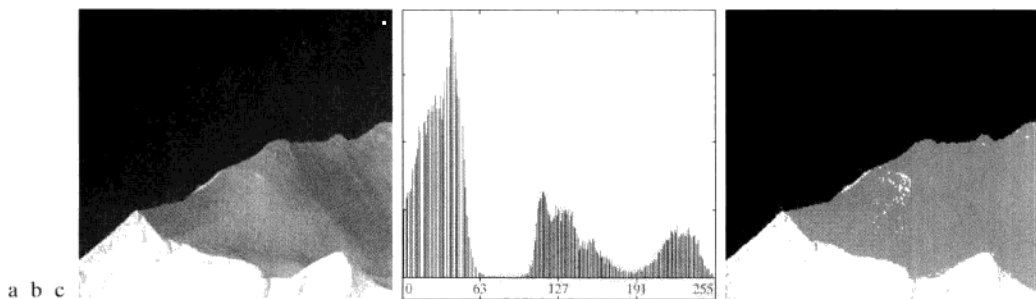


图 10.45 (a) 冰山图像; (b) 直方图; (c) 用双 Otsu 阈值将图像分割为三个区域(原图像由 NOAA 提供)

10.3.7 可变阈值处理

正如 10.3.1 节中讨论的那样,例如噪声和非均匀光照这样的因素对阈值处理算法的性能起着重要作用。我们在 10.3.4 节和 10.3.5 节说明过,图像平滑和使用边缘信息可以得到有益的帮助。然而,

常常会出现这种情况,即用迄今为止所讨论的任何方法解决问题时,要么这种类型的预处理不切实际,要么对这一情形的改进简单而无效。在这种情况下,更为高级的且复杂的阈值处理涉及可变阈值问题。在这一节中,我们讨论选择可变阈值的各种技术。

图像分块

可变阈值处理最简单的方法之一是把一幅图像分成不重叠的矩形。这种方法用于补偿光照和/或反射的不均匀性。选择的矩形要足够小,以便每个矩形的光照都近似是均匀的。我们用一个例子来说明这一方法。

例 10.20 通过图像分块的可变阈值处理。

图 10.46(a)显示了来自图 10.37(c)的图像,图 10.46(b)显示了其直方图。当讨论图 10.37(c)时,我们可得出这样的结论:使用一个全局阈值不能分割该图像,图 10.46(c)和(d)已确认了这一事实,图 10.46(c)和(d)分别显示了使用 10.3.2 节中讨论的迭代方案和 Otsu 方法分割图像后的结果。两种方法产生了可比较的结果,结果中大量的分割错误显而易见。

图 10.46(e)显示了已细分为 6 个矩形区域的原图像,图 10.46(f)是对每幅子图像应用 Otsu 全局方法得到的结果。尽管分割结果中的一些错误仍然是可见的,但在—幅分割起来十分困难的图像上,图像细分产生了合理的结果。分析每幅子图像的直方图,很容易就可解释这种改进的原因。如图 10.47 所示,每幅子图像都由一个双模式直方图来表征,两个模式之间有一个较深的波谷,这是我们所知的将导致有效全局阈值处理的一个事实。

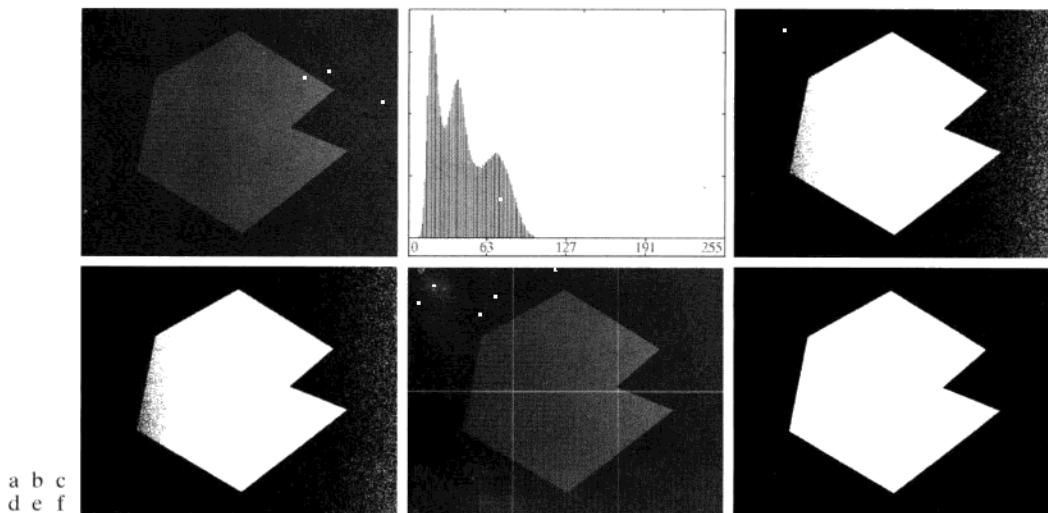


图 10.46 (a)带噪声阴影的图像;(b)该图像的直方图;(c)用来自 10.3.2 节的全局迭代算法对图(a)的分割;(d)使用 Otsu 方法得到的结果;(e)分为 6 幅子图像后的图像;(f)分别对每幅子图像应用 Otsu 方法的结果

当感兴趣物体和背景占据合理的可比大小的区域时,通常图像细分工作得较好,如图 10.46 所示。当不是这种情况时,典型地,该方法会失败,因为细分可能只包含物体或背景像素。虽然使用附加的技术来确定何时一个细分包含两种像素可以解决这一情况,但解决不同情况所要求的逻辑会变得很复杂。在这种情况下,使用这一节其余部分讨论的方法可能更好。

基于局部图像特性的可变阈值处理

与前一节中讨论的图像细分方法相比,更为一般的方法是在一幅图像中的每一点 (x, y) 计算阈值,该阈值是以一个或多个在 (x, y) 邻域计算的特性为基础的。虽然这可能是似乎有点费力的处理,但现代算法和硬件允许进行快速的邻域处理,特别是对于像逻辑操作和算术操作这样的普通函数。

我们用一幅图像中每个点的一个邻域内的像素的标准差和均值来说明局部阈值处理的基本方法。这两个量对于确定局部阈值非常有用,因为它们是局部对比度和平均灰度的描述子。令 σ_{xy} 和 m_{xy} 表示一幅图像中以坐标 (x, y) 为中心的邻域 S_{xy} 所包含的像素集合的标准差和均值(关于局部均值和标准差的计算,见 3.3.4 节)。下面是可变局部阈值的通用形式:

$$T_{xy} = a\sigma_{xy} + bm_{xy} \quad (10.3-33)$$

其中, a 和 b 是非负常数, 且

$$T_{xy} = a\sigma_{xy} + bm_G \quad (10.3-34)$$

其中, m_G 是全局图像均值。分割后的图像计算如下:

$$g(x, y) = \begin{cases} 1, & f(x, y) > T_{xy} \\ 0, & f(x, y) \leq T_{xy} \end{cases} \quad (10.3-35)$$

其中, $f(x, y)$ 是输入图像。该式对图像中的所有像素位置进行求值, 并在每个点 (x, y) 处使用邻域 S_{xy} 中的像素计算不同的阈值。

使用以 (x, y) 的邻域计算出的参数为基础的属性, 有效的权值可显著增强局部阈值处理(计算量会适度增加):

$$g(x, y) = \begin{cases} 1, & Q(\text{局部参数}) \text{ 为真} \\ 0, & Q(\text{局部参数}) \text{ 为假} \end{cases} \quad (10.3-36)$$

其中, Q 是以邻域 S_{xy} 中像素计算的参数为基础的一个属性。例如, 考虑如下基于局部均值和标准差的属性 $Q(\sigma_{xy}, m_{xy})$:

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} \text{真}, & f(x, y) > a\sigma_{xy} \text{ AND } f(x, y) > bm_{xy} \\ \text{假}, & \text{其他} \end{cases} \quad (10.3-37)$$

注意, 式(10.3-35)是式(10.3-36)的一种特殊情况, 它是在 $f(x, y) > T_{xy}$ 时令 Q 为真, 而在其他情况下令 Q 为假得到的。在这种情况下, 该属性仅简单地基于一个点处的灰度。

例 10.21 基于局部图像特性的可变阈值处理。

图 10.48(a)显示了来自例 10.18 的酵母图像。该图像有三个主要的灰度级, 因此也许双阈值处理是一种较好的分割方法的假设是合理的。图 10.48(b)是使用 10.3.6 节中说明的双阈值处理方法得到的结果。如图所示, 从背景中分离出明亮区域是可能的, 但是, 在图像右侧的中等灰度区域没有完全分割 [回忆

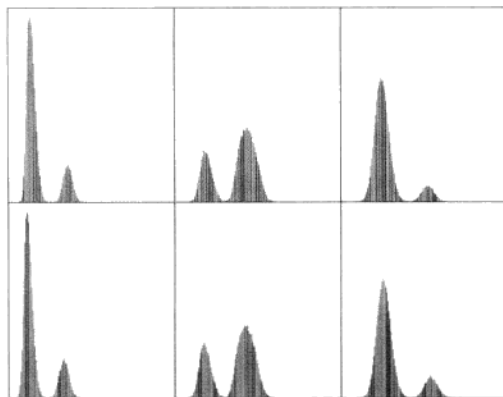


图 10.47 图 10.46(c)中 6 幅子图像的直方图

我们在例 10.18 中,遇到的与图 10.43(c)类似的问题]。为说明局部阈值处理的用法,我们用大小为 3×3 的邻域对输入图像中的所有 (x, y) 计算局部标准差 σ_{xy} 。图 10.48(c)显示了结果。注意,微弱的外部线是如何正确地刻画细胞的边界的。接着,我们使用全局均值替代 m_{xy} 来形成式(10.3-37)所示形式的一个属性。通常,当背景近似恒定,并且所有物体的灰度高于背景灰度或低于背景灰度时,选择全局均值会给出较好的结果。在完成的属性规范中,使用了值 $a = 30$ 和 $b = 1.5$ (在类似这样的应用中,这些值通常是通过试验确定的)。然后,用式(10.3-36)来分割图像。如图 10.48(d)所示,结果与输入图像中的两类灰度的大部分区域十分接近。尤其要注意所有的外部区域完全被正确分割了,且大多数较亮的内部区域也已被正确地分开了。

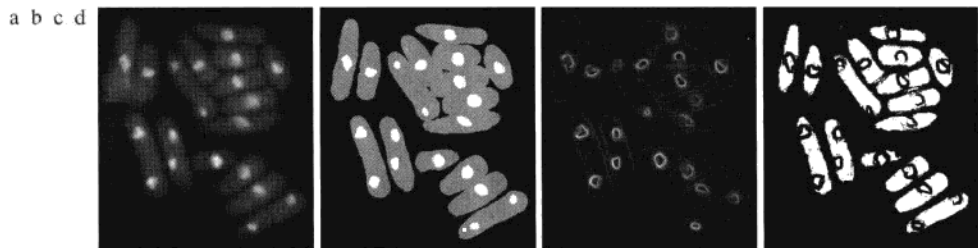


图 10.48 (a)来自图 10.43 的图像;(b)用 10.3.6 节中讨论的双阈值处理方法分割的图像;(c)局部标准差图像;(d)用局部阈值处理得到的结果

使用移动平均

刚才讨论的局部阈值处理方法的一种特殊情形是以一幅图像的扫描行计算移动平均为基础的。在速度是一种基本要求的文档处理中,这种实现相当有用。典型地,为减少光照偏差,扫描是以 Z 字形模式逐行执行的。令 z_{k+1} 表示步骤 $k+1$ 中扫描序列中遇到的点的灰度。这个新点处的移动平均(平均灰度)由下式给出:

$$m(k+1) = \frac{1}{n} \sum_{i=k+2-n}^{k+1} z_i = m(k) + \frac{1}{n} (z_{k+1} - z_{k-n}) \quad (10.3-38)$$

其中, n 表示用于计算平均的点数,且 $m(1) = z_1 / n$ 。这个初始值严格来说并不正确,因为单点的平均是该点自身的值。然而,我们使用 $m(1) = z_1 / n$, 因此,第一次开始计算式(10.3-38)时,并不要求特殊的计算。从另一个角度看,如果图像的边界填充了 $n-1$ 个 0, 它就是我们将得到的那个值。该算法仅初始化一次,不是在每行都初始化。因为对图像中的每个点都计算移动平均,因此,用式(10.3-35)实现分割,其中 $T_{xy} = bm_{xy}$, b 是常数, m_{xy} 是在输入图像中的点 (x, y) 处使用式(10.3-38)得到的移动平均。

第一个表达式在 $k \geq n-1$ 时有效。当 $k < n-1$ 时,平均是使用可用点形成的。类似地,第二个表达式在 $k \geq n+1$ 时有效。

例 10.22 用移动平均的文档阈值处理。

图 10.49(a)显示了一幅以斑点灰度模式遮蔽的手写文本图像。这种灰度遮蔽形式是使用摄影闪光灯得到的典型图像。图 10.49(b)是使用 Otsu 全局阈值处理方法分割的结果。全局阈值处理不能克服灰度变化并不意外。图 10.49(c)显示了采用移动平均的局部阈值处理的成功分割。一个经验是令 n 等于平均笔画宽度的 5 倍。在这种情况下,平均宽度是 4 个像素,因此我们在式(10.3-38)中令 $n = 20$ 并使用 $b = 0.5$ 。

作为这种分割方法有效性的另一个说明,我们使用了与前一段中相同的参数来分割图 10.50(a)中的图像,该图像被正弦型灰度变化污染了,当文本扫描仪的电源未适当接地时,会出现这种典型的变化。如图 10.50(b)和(c)所示,分割结果类似于图 10.49 中的那些结果。

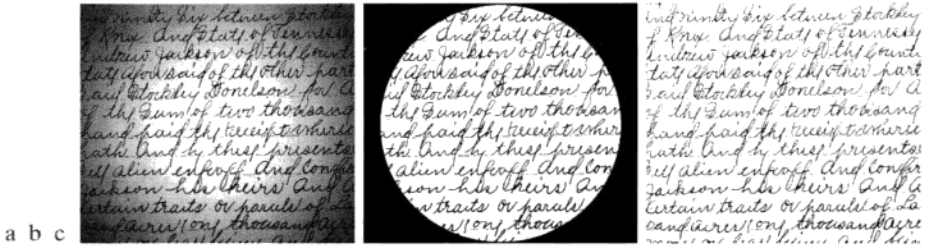


图 10.49 (a)被斑点遮蔽污染的文本图像; (b)使用 Otsu 方法的全局阈值处理的结果; (c)使用移动平均的局部阈值处理的结果



图 10.50 (a)被正弦亮度遮蔽污染的文本图像; (b)使用 Otsu 方法的全局阈值处理的结果; (c)使用移动平均的局部阈值处理的结果

有趣的是,使用相同的 n 值和 b 值,两种情形下均得到了成功的分割结果,这说明了这种方法的相对稳定性。通常,当感兴趣的物体与图像尺寸相比较小(或较细)时,基于移动平均的阈值处理会工作得很好。打印图像和手写文本图像满足这一条件。

10.3.8 多变量阈值处理

迄今为止,我们已涉及了基于单个变量(灰度)的阈值处理。在某些情况下,传感器可产生多个可利用的变量来表征图像中的每个像素,这样,就允许进行多变量阈值处理。一个显著的例子就是彩色成像,其中红(R)、绿(G)和蓝(B)分量用于形成一幅合成彩色图像(见第 6 章)。在这种情况下,每个“像素”由三个值来表征,并且可以表示为一个三维向量 $\mathbf{z} = (z_1, z_2, z_3)^T$, 其分量是一个点的 RGB 彩色。这些三维点通常称为体素,以便与图像元素相对应表示体积元素。

如 6.7 节中讨论的某些细节那样,多变量阈值处理可视为一种距离计算。假设我们想要从一幅彩色图像中提取具有指定颜色的所有区域,比如说,微红色。令 $\mathbf{a}$ 表示我们感兴趣的平均微红色。基于这一参数来分割彩色图像的一种方法是计算任意彩色点 $\mathbf{z}$ 和平均彩色 $\mathbf{a}$ 间的距离测度 $D(\mathbf{z}, \mathbf{a})$ 。然后,我们按如下方式分割输入图像:

$$g = \begin{cases} 1, & D(\mathbf{z}, \mathbf{a}) < T \\ 0, & \text{其他} \end{cases} \tag{10.3-39}$$

其中, T 是一个阈值。当然,要在输入图像中的所有坐标上执行距离计算,以便产生相应的分割值 g 。注意,该公式中的不等式与我们在式 (10.3-1) 中对于一个单变量阈值处理所用的不等式是相反的,原因是等式 $D(\mathbf{z}, \mathbf{a}) = T$ 定义了一个容积(见图 6.43), 并且它更直观地把分割后的像素值想象为包含在一个容积中,而使背景像素值位于该容积的表面或外部。令 $D(\mathbf{z}, \mathbf{a}) = -f(x, y)$, 式 (10.3-39) 便可简化为式 (10.3-1)。

783
~
784

很明显, 条件 $f(x, y) > T$ 基本上表示 f 的值和实线的原点间的欧氏距离超过了 T 值。这样, 阈值处理就是基于距离测度的计算, 并且, 式 (10.3-39) 的形式依赖于所用的测度。通常, 如果 z 是一个 n 维向量, 我们从 2.6.6 节可知 n 维欧几里得距离定义为

$$D(z, a) = \|z - a\| = \left[(z - a)^T (z - a) \right]^{\frac{1}{2}} \quad (10.3-40)$$

等式 $D(z, a) = T$ 描述了欧几里得空间中 (图 6.4.3 中显示了一个三维例子) 的一个球 (称为超球)。一种更有用的距离测度是所谓的马氏 (Mahalanobis) 距离, 定义为

$$D(z, a) = \left[(z - a)^T C^{-1} (z - a) \right]^{\frac{1}{2}} \quad (10.3-41)$$

其中, 如 12.2.2 节讨论的那样, C 是 z 的协方差矩阵。 $D(z, a) = T$ 描述了一个 n 维超球 (图 6.4.3 显示了一个三维例子)。当 $C = I$ 时, 即为么阵时, 该表达式就简化为式 (10.3-40)。

我们在 6.7 节中已给出了一个关于使用这些表达式的详细例子。我们在 12.2 节中还要讨论使用基于决策函数的模式识别技术来分割图像的问题, 这种问题可看成是多类、多变量阈值处理问题。

10.4 基于区域的分割

如 10.1 节中讨论的那样, 分割的目的是将一幅图像划分为多个区域。在 10.2 节中, 我们其于灰度级的不连续性尝试寻找区域间的边界来解决这一问题; 而在 10.3 节中, 分割是通过以像素特性分布为基础的阈值处理来完成的, 如灰度值或彩色。在本节中, 我们讨论以直接寻找区域为基础的分割技术。

在继续阅读下文前, 建议读者回顾一下 10.1 节中介绍的术语。

10.4.1 区域生长

正如它的名字所暗示的那样, 区域生长是根据预先定义的生长准则将像素或子区域组合为更大区域的过程。基本方法是从一组“种子”点开始, 将与种子预先定义的性质相似的那些邻域像素添加到每个种子上来形成这些生长区域 (如特定范围的灰度或颜色)。

如稍后在例 10.23 中显示的那样, 通常我们可基于问题的性质来选择一组或多组开始点。当先验知识不可用时, 这一过程是在每个像素处计算相同特性的集合, 最终, 在生长过程期间, 根据这些特性集合把像素分配给各个区域。如果这些计算的结果显示了一族值, 则那些特性靠近这些族的中心的像素可以作为种子使用。

相似性准则的选择不仅取决于所面对的问题, 还取决于现有的图像数据类型。例如, 土地利用卫星成像分析非常倚重彩色的使用。如果彩色图像中没有可用的固有信息, 那么解决这个问题将非常困难, 甚至无法解决。如果图像是单色的, 就必须使用一组基于灰度级和空间特性的描绘子 (如矩或纹理) 对区域进行分析。在第 11 章中, 我们将讨论那些对区域特征有用的描绘子。

如果在区域生长过程中没有使用连通属性, 那么单独的描绘子会产生错误的结果。例如, 设想仅有三个不同灰度值的像素的随机排列。将具有相同灰度级的像素组合形成一个“区域”, 而不考虑连通性, 会产生对当前讨论的内容毫无意义的分割结果。

区域生长的另一个问题是终止规则的表示法。当不再有像素满足加入某个区域的准则时, 区域生长就会停止。像灰度值、纹理和彩色准则本质上都是局部的, 都没有考虑区域生长的“历史”。增强区域生长算法能力的其他准则利用了候选像素和已加入生长区域的像素间的大小、相似性等概念 (如

候选像素灰度和生长区域的平均灰度的比较), 以及正在生长的区域的形状。这类描绘子的使用是以期望结果的模型至少部分可用这一假设为基础的。

令 $f(x, y)$ 表示一个输入图像阵列; $S(x, y)$ 表示一个种子阵列, 阵列中种子点位置处为 1, 其他位置处为 0; Q 表示在每个位置 (x, y) 处所用的属性。假设阵列 f 和 S 的尺寸相同。基于 8 连接的一个基本区域生长算法可说明如下。

关于连通分量, 请参阅 2.5.2 节和 9.5.3 节; 关于腐蚀, 请参阅 9.2.1 节。

1. 在 $S(x, y)$ 中寻找所有连通分量, 并把每个连通分量腐蚀为一个像素; 把找到的所有这种像素标记为 1, 把 S 中的所有其他像素标记为 0。
2. 在坐标对 (x, y) 处形成图像 f_Q : 如果输入图像在该坐标处满足给定的属性 Q , 则令 $f_Q(x, y) = 1$, 否则令 $f_Q(x, y) = 0$ 。
3. 令 g 是这样形成的图像: 即把 f_Q 中为 8 连通种子点的所有 1 值点, 添加到 S 中的每个种子点。
4. 用不同的区域标记(如 1, 2, 3, ...) 标出 g 中的每个连通分量。这就是由区域生长得到的分割图像。

我们通过一个例子来说明这种算法的机理。

例 10.23 使用区域生长的分割。

图 10.51(a) 显示了一条焊缝(水平深色区域)的 8 比特 X 射线图像, 该图像中含有几条裂缝和孔隙(水平横穿图像中心的明亮区域)。我们用分割有缺陷的焊缝区域来说明区域生长的应用。这些区域可用于焊缝检测, 包括有关历史学习的数据库或控制自动焊接系统。

工作的第一步是确定种子点。从问题的物理性质看, 我们知道裂缝和孔隙与实的焊缝相比, 对 X 射线的衰减要小, 因此, 我们预料包含这种类型缺陷的区域要比 X 射线图像中的其他部分明显亮一些。我们可用在高百分比处设置阈值对原始图像进行阈值处理来提取种子点。图 10.51(b) 显示了图像的直方图, 图 10.51(c) 显示了用等于图像中灰度值的 99.5% 的阈值对图像进行阈值处理后的结果, 在这种情况下, 阈值为 254(见 10.3.5 节关于百分比的说明)。图 10.51(d) 显示了将图 10.51(c) 中的每个连通分量用形态学腐蚀为一个单点后的结果。

接下来, 我们必须确定一个属性。在这个例子中, 我们的兴趣是把满足如下要求的所有像素添加到每个种子中: (a) 对于种子是 8 连接的, 并且 (b) 与种子相似。若使用灰度差作为一种相似性度量, 则应用于每一个位置 (x, y) 处的属性是

$$Q = \begin{cases} \text{TRUE}, & \text{种子和点}(x, y)\text{处的像素间灰度差的绝对值} \leq T \\ \text{FALSE}, & \text{其他} \end{cases}$$

其中, T 是一个指定的阈值。虽然这个属性是基于灰度差的, 并且使用单个阈值, 但我们可指定更复杂的方案, 即对每个像素应用一个不同的阈值, 并使用不同于差值的其他属性。在这种情况下, 前述的属性足以解决该问题, 如这个例子的其余部分说明的那样。

从前面的章节我们知道, 最小的种子值是 255, 因此, 使用 254 的阈值对图像进行阈值处理。图 10.51(e) 显示了图 10.51(a) 和 (c) 所示图像的差的绝对值。图 10.51(e) 中的图像包含了用于计算每个位置 (x, y) 处的属性的所需要的差值。图 10.51(f) 显示了相应的直方图。我们需要一个阈值, 以便用于建立相似性属性。该直方图有三个主要模式, 因此我们首先把 10.3.6 节中讨论的双阈值处理技术用于差值图像。在这种情况下, 导致的两个阈值是 $T_1 = 68$ 和 $T_2 = 126$, 我们看到, 它们相当靠近直方图的波谷[短暂地离开一下话题, 我们用这两个阈值分割了图像。图 10.51(g) 显示了使用双阈值不能解决缺陷的分割问题, 尽管阈值处在主要波谷的中]。

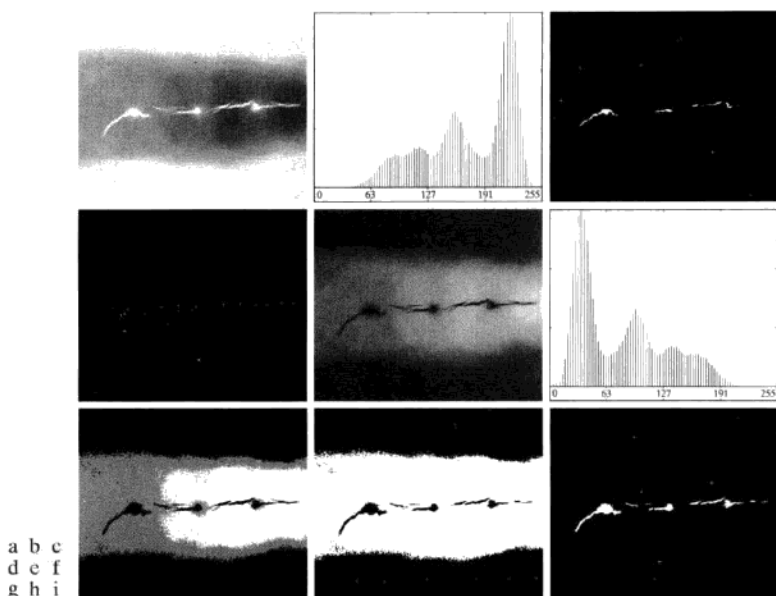


图 10.51 (a) 有缺陷焊缝的 X 射线图像; (b) 直方图; (c) 初始种子图像; (d) 最终种子图像 (为清楚起见, 点已被放大); (e) 图(a)和图(c)之差的绝对值; (f) 图(e)的直方图; (g) 使用双阈值进行阈值处理后的差值图像; (h) 使用最小双阈值进行阈值处理后的差值图像。 (i) 用区域生长得到的分割 (原图像由 X-TEK Systems 有限公司提供)

图 10.51 (h) 显示了仅使用 T_1 对差值图像进行阈值处理的结果。黑点是其属性为“真”(TRUE)的像素, 其他点是属性为“假”(FALSE)的像素。这里, 重要结果是焊缝的良好区域中断定为失败的点, 因此, 这些点将不包含在最终结果中。在外部区域的点将由区域生长算法作为候选点来考虑。然而, 步骤 3 将拒绝外部点, 因为它们与种子不是 8 连接的。事实上, 如图 10.51 (i) 所示, 这一步导致了正确的分割, 这表明连接性的使用是这种情况下的一个基本需求。最后, 注意, 在步骤 4 中, 我们对由该算法找到的所有区域使用了相同的值。在这种情况下, 它确实做得很好。

10.4.2 区域分裂与聚合

上一节中讨论的过程是从一组种子点来生长区域。另一种可供选择的方法是首先将一幅图像细分为一组任意的不相交区域, 然后聚合和/或分裂这些区域, 并力图满足 10.1 节中阐述的分割条件。下面讨论分裂和聚合的基础知识。

令 R 表示整幅图像区域, 并选择一个属性 Q 。对 R 进行分割的一种方法是依次将它细分为越来越小的四象限区域, 以便对于任何区域 R_i 有 $Q(R_i) = \text{TRUE}$ 。我们从整个区域开始。如果 $Q(R) = \text{FALSE}$, 那么将该图像分割为 4 个象限区域。如果对于每个象限区域 Q 为 FALSE, 则将该象限区域再次细分为四个子象限区域, 依此类推。这种特殊的分裂技术有一个方便的表示方法, 即用所谓的四叉树形式表示, 即每个节点都正好有 4 个后代, 如图 10.52 所示 (对应一个四叉树的节点的图像有时称为四分区域或四分图像)。注意, 树根对应于整幅图像, 而每个节点对应于该节点的 4 个细分子节点。在这种情况下, 仅 R_4 被进一步再细分。

如果只使用分裂, 那么最后的分区通常包含具有相同性质的邻接区域。这种缺陷可以通过允许聚合和分裂得到补救。要满足 10.1 节

关于区域相邻的讨论, 请参阅 2.5.2 节。

中提出的分割约束条件仅要求聚合满足属性 Q 的组合像素的邻接区域。也就是说，只有在 $Q(R_j \cup R_k) = \text{TRUE}$ 时，两个邻接区域 R_j 和 R_k 才能聚合。

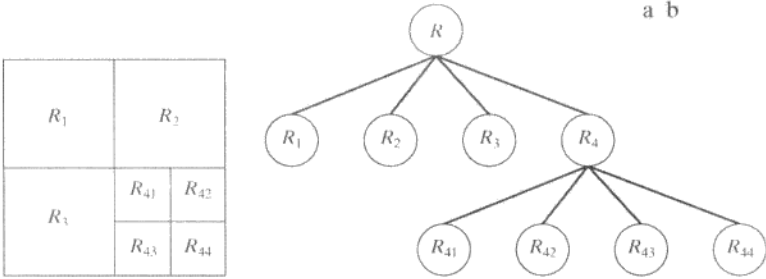


图 10.52 (a) 被分割的图像; (b) 对应的四叉树。 R 表示整个图像区域

前述讨论可以小结为如下过程，在该过程的任何一步中，我们

- 1. 对满足 $Q(R_i) = \text{FALSE}$ 的任何区域 R_i 分裂为 4 个不相交的象限区域。
- 2. 当不可能进一步分裂时，对满足条件 $Q(R_j \cup R_k) = \text{TRUE}$ 的任意两个邻接区域 R_j 和 R_k 进行聚合。
- 3. 当无法进一步聚合时，停止操作。

习惯上要规定一个不能再进一步执行分裂的最小四象限区的尺寸。

前述的基本原理可能有多种变化。例如，在步骤 2 中，如果两个邻接的区域 R_i 和 R_j 各自都满足属性，则我们允许聚合这两个区域会得到有意义的简化。这将导致简单得多(且快得多)的算法，因为属性的测试被限制在各个四分区域。正如下例所示，这种简化仍能产生较好的分割结果。

例 10.24 用区域分裂和聚合进行分割。

图 10.53(a) 显示了天鹅星座环的一幅大小为 566×566 的 X 射线波段的图像。这个例子的目的是把围绕致密中心的不太密集的环状物质的图像分割出来。感兴趣区域有一些有助于分割的明显特征。首先，我们注意到该区域中的数据具有随机性质，这表明其标准差应大于背景和较大的相当平滑的中心区的标准差(背景的标准差近于 0)。类似地，来自外环的包含数据的一个区域的均值(平均灰度)应大于暗背景的均值而小于较大且亮中心区域的均值。从而，我们应该能够用下面的属性分割感兴趣的区域：

$$Q = \begin{cases} \text{TRUE}, & \sigma > a \text{ 且 } 0 < m < b \\ \text{FALSE}, & \text{其他} \end{cases}$$

其中， m 和 σ 是一个四象限区域中的像素的均值和标准差， a 和 b 为常数。

对感兴趣外部区域中的几个区域的分析表明，这些区域中的像素的平均灰度未超过 125，而标准差总是大于 10。图 10.53(b) 到 (d) 显示了对 a 和 b 使用这些值和允许的四象限区域的最小尺寸从 32 变到 8 得到的结果。一个四象限区域中满足属性的像素置为白色，而该区域中的其他像素则置为黑色。根据截获的外部区域的形状，最好的结果是使用大小为 16×16 的四象限区域得到的。图 10.53(d) 中的黑色方块是大小为 8×8 的四象限区域，该区域中的像素不满足属性。使用较小的四象限区域将导致此类黑色区域数量的增加。使用比此处说明的大区域会导致更多“类似块”的分割。注意，在所有情况下，分割后的区域(白色像素)从背景中完全分隔了平滑的内部区域。这样，分割就有效地把图像分成了三个明显的区域，这些区域对应于图像中的三个主要特征：背景、密集区域和稀疏区域。若使用图 10.53 中的任何一个白色区域作为模板，将使从原图像中提取这些区域成为一项相对简单的任务(见习题 10.40)。正如例 10.23 那样，使用基于边缘或阈值的分割不可能得到这些结果。



图 10.53 (a) 由 NASA 的哈勃望远镜在 X 射线波段拍摄的天鹅星座环超新星的图像; (b)~(d) 分别将允许的最小四象限区域尺寸限制为 32×32 , 16×16 和 8×8 像素时得到的结果(原图像由 NASA 提供)

如前面的例子所使用的那样, 基于一个区域内像素灰度的均值和标准差的特性试图量化该区域的纹理(关于纹理的讨论, 见 11.3.3 节)。纹理分割的概念是以在属性上使用纹理测度为基础的。换句话说, 我们可以使用本节讨论的任何一种方法, 用指定基于纹理内容的属性简单地执行纹理分割。

10.5 用形态学分水岭的分割

迄今为止, 我们已经讨论了基于三种主要概念的分割: (a) 边缘检测, (b) 阈值处理和 (c) 区域生长。每种方法都有其优点(例如全局阈值处理具有速度优势)和缺点(例如, 在基于边缘的分割中, 需要进行后处理, 如边缘连接)。在本节中, 我们将讨论基于所谓的形态学分水岭概念的方法。在下面的讨论中, 将更为明显, 形态学分水岭分割将其他三种方法中的许多概念进行了具体化, 因此通常会产生更稳定的分割结果, 包括连接的分割边界。这种方法为在分割中结合基于知识的约束提供了一个简单的框架。

10.5.1 背景知识

分水岭的概念是以三维形象化一幅图像为基础的: 两个空间坐标作为灰度的函数, 如图 2.18(a) 所示。在这种“地形学”解释中, 我们考虑三种类型的点: (a) 属于一个区域最小值的点; (b) 把一点看成是一个水滴, 如果把这些点放在任意位置上, 水滴一定会下落到一个单一的最小值点; (c) 处在该点的水会等可能性地流向不止一个这样的最小值点。对于一个特定的区域最小值, 满足条件 (b) 的点的集合称为该最小值的汇水盆地或分水岭。满足条件 (c) 的点形成地表面的峰线, 这种峰线称为分割线或分水线。

基于这些概念的分割算法的主要目标是找出分水线, 其基本思想非常简单, 如下面类似的说明那样。假设在每个区域的最小值上打一个洞, 并且让水通过洞以均匀的速率上升, 从低到高淹没整个地形。当不同的汇水盆地中上升的水聚集时, 修建一个水坝来阻止这种聚合。水将达到在水线上只能见到各个水坝的顶部这样一个程度。这些大坝的边界对应于分水岭的分割线。因此, 它们是由分水岭算法提取出来的(连接的)边界。

借助于图 10.54, 我们可以进一步解释这些概念。图 10.54(a) 显示了一幅灰度图像, 图 10.54(b) 是一幅地形俯视图, 其中“山峰”的高度与输入图像中的灰度值成正比。为了便于解释, 这些结构的背面已被遮蔽, 以便不会与灰度值混淆, 因为我们只对三维表示的普通地形感兴趣。为了阻止上升的水从图像的边缘溢出, 我们可以想象整个地形(图像)的四周已被比最高山峰更高的水坝包围起来, 最高的山峰的值是由输入图像灰度级可能具有的最大值决定的。

假设在每个区域的最小值处打一个洞 [如图 10.54(b) 中的深色区域所示], 并且让水以均匀的速率上升, 从低到高淹没整个地形。图 10.54(c) 说明了被水淹没的第一个阶段, 其中显示为浅灰色的“水”仅覆盖了图像中相对较暗的背景区域。在图 10.54(d) 和 (e) 中, 我们看到水现在已分别上升至第一个和第二个汇水盆地。当水继续上升时, 最终会从一个汇水盆地溢出到另一个汇水盆地。第一个征兆如图 10.54(f) 所

示。这里,水实际上从左侧盆地溢出到了右侧的盆地,因此构筑了一个较短的“水坝”(由几个单一的像素组成)来阻止水在这一水位汇合(构筑水坝的细节将在下一节讨论)。随着水位的不断抬升,更加明显的效果如图 10.54(g) 所示。这幅图在两个汇水盆地之间显示了一条更长的坝,另一条水坝在右侧盆地的顶部。构筑后一水坝的目的是阻止来自盆地中的水与来自对应的背景中水的汇聚。该过程一直持续,直到达到水的最高水位(对应于图像中的最高灰度值)。最终的水坝对应于分水线,这些分水线就是我们希望的分割结果。在图 10.54(h) 中,该例的结果显示为叠加到原图像之上的一条 1 像素宽的深色路径。注意,一条重要的性质就是分水线组成一条连通的路径,这样,在两个区域间就给出了连续的边界。

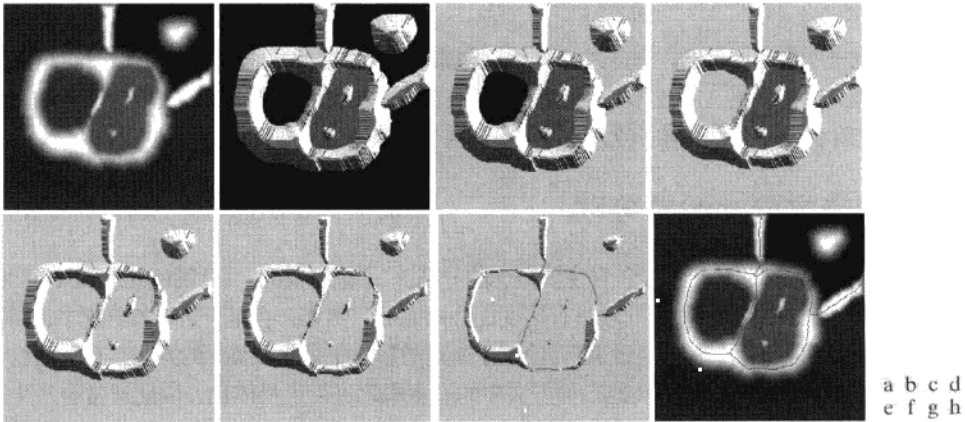


图 10.54 (a)原图像;(b)地形俯视图;(c)~(d)被水淹没的两个阶段;(e)进一步淹没的结果;(f)来自两个汇水盆地的水开始汇聚(两个汇水盆地之间构筑了一个较短的水坝);(g)较长的水坝;(h)最终的分水(分割)线(原图像由 CMM/Ecole des Mines de Paris 公司的 S. Beucher 博士提供)

分水岭分割的主要应用之一是从背景中提取近乎一致(类似于水滴)的物体。由变化较小的灰度表征的区域有较小的梯度值。因此,我们经常见到分水岭分割方法用于一幅图像的梯度,而不是图像本身。有了这一表示法,汇水盆地的区域最小值就可很好地与对应于感兴趣物体的较小梯度值联系起来。

10.5.2 水坝构建

在讨论之前,让我们考虑一下如何构建分水岭分割算法所需的水坝或分水线。水坝的构造是以二值图像为基础的,这种图像是二维整数空间 Z^2 的成员(见 2.4.2 节)。构建分离二元点集的水坝的最简方法是使用形态学膨胀(见 9.2.2 节)。

图 10.55 说明了如何使用形态学膨胀来构建水坝的基础。图 10.55(a) 显示了第 $n-1$ 步淹没的两个汇水盆地。图 10.55(b) 显示了第 n 步淹没的结果。水已从一个盆地溢出到了另一个盆地,所以,必须构建水坝来阻止这种情况的发生。为了与很快将要引入的符号一致,令 M_1 和 M_2 表示两个区域中的最小值点坐标的集合。然后,将汇水盆地中点的坐标集合与这两个在溢出的第 $n-1$ 步处的最小值点的坐标集合联系起来,在第 $n-1$ 步溢出处,两个最小值分别表示为 $C_{n-1}(M_1)$ 和 $C_{n-1}(M_2)$ 。它们是图 10.55(a) 中的两个灰色区域。

令 $C[n-1]$ 表示这两个集合的并集。图 10.55(a) 中有两个连通分量(关于连通分量的讨论,见 2.5.2 节),而图 10.55(b) 中只有一个连通分量,该连通分量包含了如虚线所示的前两个分量。两个连通分量变成一个连通分量的事实表明两个汇水盆地在第 n 步淹没中已被聚合了。令 q 表示这个连通分量。注意,来自 $n-1$ 步的两个连通分量可以通过使用简单的“与”操作($q \cap C[n-1]$)从 q 中提取出来。我们还注意到,属于单个汇水盆地的所有点形成了一个单一的连通分量。

793
~
794

假设图 10.55 (a) 中的每个连通分量被图 10.55 (c) 中所示的结构元膨胀服从两个条件: (1) 膨胀必须受到 q 约束 (这意味着在膨胀的过程中, 结构元的中心只能位于 q 中的点处), (2) 不能对会引起正被膨胀集合聚合 (变成一个单一的连通分量) 的那些点执行膨胀。图 10.55 (d) 表明, 第一轮膨胀 (显示为浅灰色) 展开了每个原始连通分量的边界。注意, 膨胀过程中, 每个点都满足条件 (1), 而在膨胀处理中, 条件 (2) 没有应用于任何点; 这样, 每个区域的边界就被均匀地扩展了。

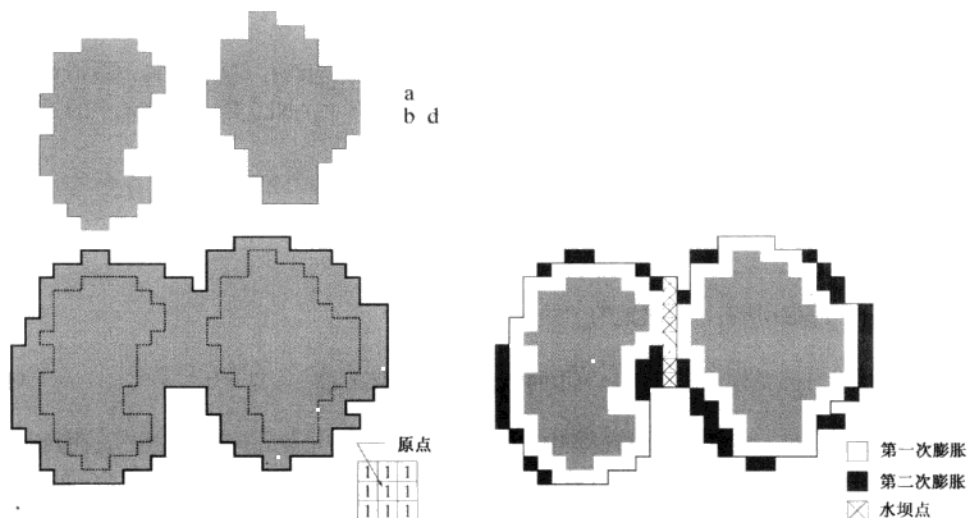


图 10.55 (a) 第 $n-1$ 步淹没处, 两个部分淹没的汇水盆地; (b) 在淹没的第 n 步处, 显示水已在两个盆地间溢出; (c) 用于膨胀的结构元; (d) 膨胀结果和水坝构建

在第二轮膨胀中 (显示为黑色), 几个不符合条件 (1) 的点却符合条件 (2), 这导致了图中所示的断开的周界。很明显, q 中只有满足上述两个条件的点, 才能描绘出图 10.55 (d) 中由打叉线所示的 1 像素宽的连通路。在淹没的第 n 步, 这条路径构成了所期望的分隔水坝。在这个淹没水平上的水坝构建, 是在刚刚确定的这条路径上的所有点设置一个大于图像最大灰度值的值来完成的。所有水坝的高度通常设置为 1 加上图像中允许的最大灰度值。当水位升高时, 这样设置可以阻止洪水漫过所完成的水坝。注意, 通过这一过程构建的水坝是连通分量是很重要的, 它就是我们希望得到的分割边界。换句话说, 这种方法消除了分割线断裂的问题。

尽管刚刚描述的过程只是基于一个简单的例子, 但该方法同样适用于更为复杂的情形, 包括使用图 10.55 (c) 中所示的 3×3 对称结构元。

10.5.3 分水岭分割算法

令 $M_1, M_2, \dots, M_R$ 是表示图像 $g(x, y)$ 的区域最小值点的坐标的集合。如同在 10.5.1 节末尾说明的那样, 这是一幅典型的梯度图像。令 $C(M_i)$ 是表示与区域最小值 M_i 相联系的汇水盆地中的点的坐标集合 (回忆任何汇水盆地中的点都形成一个连通分量)。符号 $\min$ 和 $\max$ 将用于表示 $g(x, y)$ 的最小值和最大值。最后, 令 $T[n]$ 表示满足 $g(s, t) < n$ 的坐标 (s, t) 的集合, 即

$$T[n] = \{(s, t) \mid g(s, t) < n\} \quad (10.5-1)$$

几何上, $T[n]$ 是 $g(x, y)$ 中位于平面 $g(x, y) = n$ 下方的点的坐标的集合。

随着水位以整数从 $n = \min + 1$ 到 $n = \max + 1$ 不断上升, 地形将被水淹没。在淹没过程的任意步 n , 算法都需要知道处在淹没深度之下的点的数量。从概念上来说, 假设处在 $g(x, y) = n$ 平面之下的 $T[n]$

中的坐标被“标记”为黑色,所有其他坐标被标记为白色。然后,当我们以任何淹没增量 n 处向下观察 xy 平面时,我们将会看到一幅二值图像,图像中的黑点对应于函数中平面 $g(x, y) = n$ 之下的点。这种解释对于理解下面的讨论很有帮助。

令 $C_n(M_i)$ 表示汇水盆地中与淹没阶段 n 的最小值 M_i 相关联的点的坐标集。参照前一段的讨论可知, $C_n(M_i)$ 可看成是由下式给出的一幅二值图像:

$$C_n(M_i) = C(M_i) \cap T[n] \quad (10.5-2)$$

换句话说,如果 $(x, y) \in C(M_i)$ 和 $(x, y) \in T[n]$ 相“与”,则在位置 (x, y) 处有 $C_n(M_i) = 1$, 否则 $C_n(M_i) = 0$ 。这一结果的几何解释很简单,我们在淹没阶段 n 只需使用“与”(AND)算子将 $T[n]$ 中的二值图像分离出来即可。

接下来,我们令 $C[n]$ 表示在阶段 n 中已被水淹没的汇水盆地的“并”:

$$C[n] = \bigcup_{i=1}^R C_n(M_i) \quad (10.5-3)$$

然后,令 $C[\max + 1]$ 表示所有汇水盆地的“并”:

$$C[\max + 1] = \bigcup_{i=1}^R C(M_i) \quad (10.5-4)$$

可以证明(见习题 10.41),在算法的执行过程中, $C_n(M_i)$ 和 $T[n]$ 中的元素是不会被替换的,而且当 n 增大时,这两个集合中的元素的数量不是增加,就是保持相同。这样,可理解为 $C[n-1]$ 就是 $C[n]$ 的一个子集。根据式(10.5-2)和式(10.5-3)可知, $C[n]$ 是 $T[n]$ 的一个子集,所以, $C[n-1]$ 可以理解为 $T[n]$ 的一个子集。由此,我们得出一个重要的结果: $C[n-1]$ 中的每个连通分量都恰好包含在 $T[n]$ 的一个连通分量中。

寻找分水线的算法使用 $C[\min + 1] = T[\min + 1]$ 来初始化。然后,该算法进行递归处理,由 $C[n-1]$ 计算 $C[n]$ 。由 $C[n-1]$ 求得 $C[n]$ 的过程如下:令 Q 表示 $T[n]$ 中的连通分量的集合。然后,对于每个连通分量 $q \in Q[n]$,有如下三种可能性:

1. $q \cap C[n-1]$ 为空集。
2. $q \cap C[n-1]$ 包含 $C[n-1]$ 的一个连通分量。
3. $q \cap C[n-1]$ 包含 $C[n-1]$ 的一个以上的连通分量。

由 $C[n-1]$ 构建 $C[n]$ 取决于这三个条件中的哪一个成立。当遇到一个新的最小值时,条件 1 发生,这种情况下,连通分量 q 并入 $C[n-1]$ 中形成 $C[n]$ 。当 q 位于某些局部最小值的汇水盆地内时,条件 2 发生,这种情况下, q 并入 $C[n-1]$ 中形成 $C[n]$ 。当遇到全部或部分分隔两个或多个汇水盆地的山脊线时,条件 3 发生。进一步淹没会导致这些汇水盆地中的水位聚合。因此,必须在 q 内构筑一个水坝(如果涉及两个以上的汇水盆地,就要构筑多个水坝)以阻止汇水盆地间的水溢出。正如前节中解释的那样,当我们使用元素为 1、大小为 3×3 的一个结构元来膨胀 $q \cap C[n-1]$ 且膨胀被限制到 q 时,可以构建一座 1 个像素宽的水坝。

仅用 $g(x, y)$ 中对应于现有灰度值的 n 值就可改善算法的效率;根据 $g(x, y)$ 的直方图,我们可以确定这些值,以及最小值和最大值。

例 10.25 分水岭分割算法的说明。

分别考虑图 10.56(a) 和 (b) 中的图像和它的梯度。应用刚才讨论的分水岭算法得到了图 10.56(c) 中的梯度图像的分水线(白色路径)。这些叠加到原图像上的分割边界示于图 10.56(d) 中。正如在本节开始时所注意到的那样,分割边界具有连通路径的重要特性。

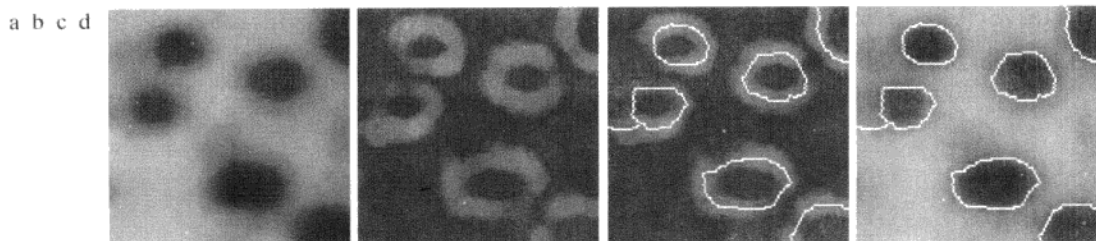


图 10.56 (a)水滴的图像；(b)图像的梯度；(c)分水线；(d)叠加在原图上的分水线(原图像由 CMM/Ecole des Mines de Paris 公司的 S. Beucher 博士提供)

10.5.4 标记的使用

直接应用前一节讨论的形式的分水岭分割算法通常会由于噪声和梯度的其他局部不规则性造成过度分割，如图 10.57 所示，过度分割的严重性足以令算法得到的结果变得毫无用处。在这种情况下，这意味着存在大量的分割区域。一个实际解决该问题的方案是通过融入预处理步骤来限制允许存在的区域的数目，设计的预处理步骤是将附加知识应用于分割过程。

用于控制过度分割的一种方法是基于标记这一概念的。一个标记是属于一幅图像的一个连通分量。与感兴趣物体相联系的标记称为内部标记，与背景相关联的标记称为外部标记。选择标记的典型过程由两个主要步骤组成：(1) 预处理；(2) 定义标记必须满足的一个准则集合。为了对此进行说明，再次考虑图 10.57 (a)。导致图 10.57 (b) 中过度分割结果的部分原因是存在大量潜在的最小值。由于它们的尺寸，许多最小值是不相关的细节。就像先前的讨论中多次指出的那样，将很小的空间细节的影响降至最低的有效方法是用一个平滑滤波器对图像进行过滤。在这种特殊情况下，这是一种合适的预处理方案。

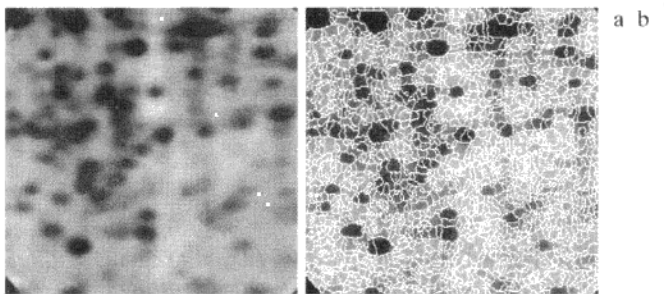


图 10.57 (a)电泳现象图像；(b)对梯度图像应用分水岭分割算法得到的结果。过度分割现象很明显(原图像由 CMM/Ecole des Mines de Paris 公司的 S. Beucher 博士提供)

假设我们将内部标记定义为：(1) 被更高“海拔”点包围的区域；(2) 区域中形成一个连通分量的那些点；(3) 连通分量中所有的点有相同灰度值。在图像经过平滑处理之后，由该定义导致的内部标记在图 10.58 (a) 中以浅灰色、斑点状区域显示。接着，在这些内部标记只能是在允许的区域最小值的限制下对平滑后的图像应用分水岭算法。图 10.58 (a) 显示了得到的分水线。将这些分水线定义为外部标记。请注意，沿分水线的点经过相邻标记间的最高点。

图 10.58 (a) 中的外部标记有效地将图像分割成了不同的区域，每个区域都包含一个单一的内部标记和部分背景。这样，问题就简化为将这些区域的每一个划分为两部分：单个物体及其背景。我们可以根据这一简化过的问题应用多种在本章先前讨论过的分割技术。另一种方法是对各个区域简单地应用分水岭分割算法。换句话说，我们简单地求得平滑后图像的梯度 [见如图 10.56 (b)]，然后，将算

法限制在只对在该特殊区域中包含这一标记的单一分水岭进行操作。使用这种方法得到的结果如图 10.58 (b) 所示。相对于图 10.57 (b) 的改善是明显的。

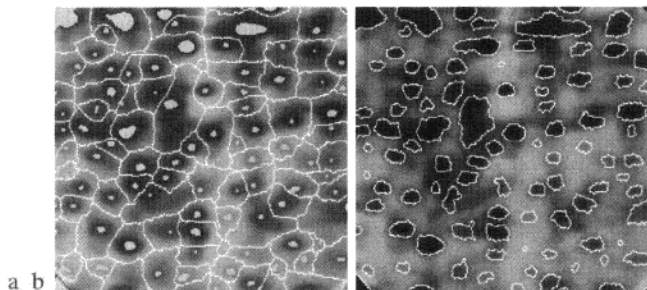


图 10.58 (a) 显示有内部标记(浅灰色区域)和外部标记(分水线)的图像; (b) 分割的结果。注意, 注意其相对于图 10.47 (b) 的改进。(原图像由 CMM/Ecole des Mines de Paris 公司的 S. Beucher 博士提供)

标记的选择可以用基于灰度值和连通性的简单过程归类, 如刚才所说明的那样, 更复杂的描述涉及尺寸、形状、位置、相对距离、纹理内容等(见第 11 章中关于描绘子的内容)。关键是使用标记带来的与分割问题有关的先验知识。应该提醒读者注意的是: 人类经常以先验知识用日常的视觉辅助进行各种分割和更高级的工作, 最熟悉的例子之一就是上下文的使用。因此, 分水岭分割方法提供了一种能有效使用这类知识的框架, 这是这种方法的一个突出的优点。

10.6 分割中运动的应用

运动是人类和其他任何动物从无细节背景中提取感兴趣物体或区域的强有力的提示。在成像应用中, 运动来自感觉系统和正被观看的场景间的相对位移, 比如机器人应用、自主导航和动态场景分析中。在下面的几节中, 我们将考虑运动在空间域和频率域分割中的运用。

10.6.1 空间域技术

基本方法

分别检测在时刻 t_i 和 t_j 获取的两帧图像帧 $f(x, y, t_i)$ 和 $f(x, y, t_j)$ 之间的变化的最简方法之一是逐像素地比较这两幅图像。这样做的一个过程是形成一幅差值图像。假设我们有一幅仅包含固定分量的参考图像。将这幅图像和一幅后续的同场景但包含一个运动物体的图像进行比较, 去掉两幅图像中的固定元素, 仅保留对应于非固定图像分量的非零项, 就可得到两幅图像的差。

在时刻 t_i 和 t_j 获取的两幅图像间的差值图像可以定义为

$$d_{ij}(x, y) = \begin{cases} 1, & |f(x, y, t_i) - f(x, y, t_j)| > T \\ 0, & \text{其他} \end{cases} \quad (10.6-1)$$

其中, T 是一个指定的阈值。注意, 就像由该指定阈值确定的那样, 仅当两幅图像间的灰度差在空间坐标 (x, y) 处明显不同时, $d_{ij}(x, y)$ 在该坐标处的值才为 1。假设所有图像的大小相同。最后, 我们注意到, 式 (10.6-1) 中坐标 (x, y) 的值跨越了这些图像的维度, 所以差值图像 $d_{ij}(x, y)$ 与序列中的其他图像的大小相同。

在动态图像处理中, $d_{ij}(x, y)$ 中值为 1 的所有像素被认为是物体运动的结果。这种方法只有在两幅图像都做了空间配准且在 T 值设定的范围内光照相对恒定的情况下才是可用的。实际上, $d_{ij}(x, y)$ 中的 1 值

项可能是由噪声造成的。通常,在差值图像中这些项是孤立的点,去除它们的一种简单方法是在 $d_g(x, y)$ 中构成由 1 组成的 4 连通或 8 连通区域,然后忽略 1 值项少于预定元素数量的任何区域。尽管这种方法会忽略掉小的和/或慢速运动的物体,但它改善了差值图像中剩余项实际是运动结果的机会。

累积差值

考虑一个图像帧序列 $f(x, y, t_1), f(x, y, t_2), \dots, f(x, y, t_n)$, 令 $f(x, y, t_1)$ 为参考图像。一幅累积差值图像 (ADI) 是通过将参考图像与序列中每幅后续图像进行比较形成的。当参考图像和序列中一幅图像之间在某个像素位置上出现一次差异时,累积图像中每个像素位置的计数器计数一次。因此,当第 k 帧图像与参考图像进行比较时,累积图像中一个给定像素的输入项会给出在此位置上灰度与参考图像相应的像素值不同的次数 [正如由式 (10.6-1) 中的 T 确定的那样]。

考虑如下累积差值图像的三种类型:绝对 ADI、正 ADI 和负 ADI。假设运动物体的灰度值大于背景的灰度值,这三种 ADI 定义如下。令 $R(x, y)$ 表示参考图像,为简化符号,令 k 表示 t_k , 则有 $f(x, y, k) = f(x, y, t_k)$ 。我们假设 $R(x, y) = f(x, y, 1)$ 。然后,对于任何 $k > 1$, 且记住 ADI 的值是计数,我们对 (x, y) 的所有相关值定义于下:

$$A_k(x, y) = \begin{cases} A_{k-1}(x, y) + 1, & |R(x, y) - f(x, y, k)| > T \\ A_{k-1}(x, y), & \text{其他} \end{cases} \quad (10.6-2)$$

$$P_k(x, y) = \begin{cases} P_{k-1}(x, y) + 1, & [R(x, y) - f(x, y, k)] > T \\ P_{k-1}(x, y), & \text{其他} \end{cases} \quad (10.6-3)$$

和

$$N_k(x, y) = \begin{cases} N_{k-1}(x, y) + 1, & [R(x, y) - f(x, y, k)] < -T \\ N_{k-1}(x, y), & \text{其他} \end{cases} \quad (10.6-4)$$

其中, $A_k(x, y)$, $P_k(x, y)$ 和 $N_k(x, y)$ 分别为遇到序列中的第 k 幅图像后的绝对 ADI、正 ADI 和负 ADI。

不难理解,这些 ADI 均从零值开始(计数)。还要注意这些 ADI 与序列中的图像大小相同。最后,我们注意到,如果背景像素的灰度值大于运动物体的灰度值,则式 (10.6-3) 和式 (10.6-4) 中的不等式的顺序和阈值的符号是相反的。

例 10.26 绝对、正和负的累积差异图像的计算。

图 10.59 显示了以灰度图像形式显示的三种 ADI, 图中矩形物体的大小为 75×50 像素, 该物体以每帧 $5\sqrt{2}$ 个像素的速度向东南方向运动。图像的大小为 256×256 像素。我们注意到下列几点: (1) 正 ADI 的非零区域等于运动物体的大小。(2) 正 ADI 的位置对应于参考帧中运动物体的位置。(3) 当运动物体与参考帧中的同一物体完全被移动时,正 ADI 中的计数停止增加。(4) 绝对 ADI 包含正 ADI 和负 ADI 的区域。(5) 运动物体的方向和速度可以由绝对 ADI 和负 ADI 中的输入决定。

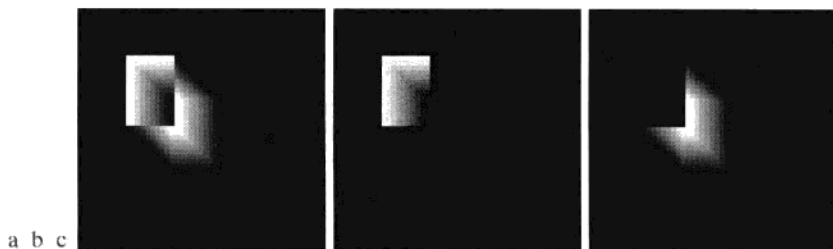


图 10.59 一个向东南方移动的矩形物体的 ADI: (a) 绝对 ADI; (b) 正 ADI; (c) 负 ADI

建立参考图像

前两节讨论的技术的成功关键在于找到一幅可以与后续图像进行比较的参考图像。在动态成像问题中,两幅图像的差值倾向于消除图像中的所有固定分量,而只留下对应于噪声和运动物体的图像元素。

实际上,仅用固定元素得到一幅参考图像并不总是可行的,所以有必要根据一组包含一个或多个运动物体的图像来构建一幅参考图像。这尤其适用于描述变化频繁的场景或需要频繁更新的场景的情况。生成参考图像的过程如下所述。考虑将图像序列中的第一幅图像作为参考图像。当一个非固定分量完全移出其参考帧的位置时,在当前帧中对应的背景可复制到最初被参考帧中的物体所占据的位置。当所有的移动物体均已完全移出其原来的位置时,就创建了一幅仅包含固定分量的参考图像。像前节中说明的那样,通过监视正 ADI 中的变化,可确定物体的位移。

例 10.27 构造一幅参考图像。

图 10.60(a)和(b)显示了一个交通十字路口的两幅图像帧。第一幅图像作为参考图像,第二幅图像描述了一段时间后的同一场景。为生成一幅静止图像,我们的目的是去掉参考图像中的主要运动物体。尽管存在其他较小的运动物体,但主要的运动特征是在十字路口从左到右运动的汽车。为了说明这一目的,我们将注意力放在这个物体上。通过监视正 ADI 的变化,可以确定一个移动物体的初始位置,如先前解释的那样。一旦确定了被这个物体占据的区域后,使用减法可从图像中移除该物体。通过观察图像序列中正 ADI 停止变化的那一帧图像,我们可以从该幅图像中复制先前被初始帧中的移动物体所占据的区域。然后,将该区域粘贴到被挖去物体的图像中,从而恢复该区域的背景。如果对所有运动的物体都进行这种操作,结果将是一幅只包含静止分量的参考图像,针对该参考图像,我们可在运动检测中与后续帧进行比较。在这种情况下,图 10.60(c)显示了去掉向东运动的汽车后得到的结果。



图 10.60 构造一幅静止参考图像: (a) 和 (b) 序列中的两帧图像; (c) 从图(a)中减去向东运动的汽车并使用图(b)中的对应区域恢复背景后的结果(原图像由 Jain 和 Jain 提供)

10.6.2 频率域技术

在这一节中,我们考虑通过傅里叶变换公式来确定运动的问题。考虑一个序列 $f(x, y, t), t = 1, 2, \dots, K-1$, 该序列是由一台固定的照相机所拍摄的大小为 $M \times N$ 的 K 帧数字图像。我们从假设所有帧都有零灰度的均匀背景开始讨论。例外的是一个单一的具有单位灰度的一个 1 像素大小的物体,该物体以恒定速度运动。假设对于第一帧($t=0$),物体在位置 (x', y') 处,并且将图像平面投影到 x 轴上;也就是说,对图像中的一列像素的灰度相加。这一操作产生一个带有 M 个零值项的一维阵列,除了 x' 处,它是单点物体的 x 坐标。现在,如果我们用 $\exp[j2\pi a_1 x \Delta t]$ 乘以一维阵列中的每个分量,其中 $x = 1, 2, \dots, M-1$, 并把结果相加,我们会得到单项 $\exp[j2\pi a_1 x' \Delta t]$ 。在这种表示中, a_1 是一个正整数, Δt 是不同帧之间的时间间隔。

假设在第二帧中($t=1$),物体已运动到坐标 $(x'+1, y')$ 处;也就是说,物体在平行于 x 轴的方向移动了 1 个像素的位置。然后,重复上一段中提到的投影过程,得到和 $\exp[j2\pi a_1 (x'+1) \Delta t]$ 。如果物体

每帧继续移动一个像素, 则在任何整数瞬间 t , 结果为 $\exp[j2\pi a_1(x'+t)\Delta t]$, 使用欧拉公式, 可将它表示为

$$e^{j2\pi a_1(x'+t)\Delta t} = \cos[2\pi a_1(x'+t)\Delta t] + j\sin[2\pi a_1(x'+t)\Delta t] \quad (10.6-5)$$

其中, $t = 0, 1, \dots, K-1$ 。换句话说, 这一过程得到一个频率为 a_1 的复正弦曲线。如果物体在帧间移动 V_1 个像素(在 x 方向), 那么正弦曲线的频率为 $V_1 a_1$ 。因为 t 在 0 和 $K-1$ 之间以整数增量变化, 限制 a_1 只取整数值会使复正弦曲线的离散傅里叶变换具有两个波峰——一个位于频率 $V_1 a_1$ 处, 另一个位于 $K - V_1 a_1$ 处。如 4.6.4 节中讨论的那样, 后一个波峰是由离散傅里叶变换的对称性产生的, 故可以忽略掉。因此, 在傅里叶频谱中找到一个波峰, 就得到 $V_1 a_1$ 。用 a_1 除该量得到 V_1 , 它就是 x 方向的速度分量, 这里假设帧速率是已知的。采用类似的方法可以得到 y 方向的速度分量 V_2 。

没有运动发生的帧序列会产生相同的指数项, 其傅里叶变换在频率 0(一个单一的直流项)处有一个单一的波峰。所以, 由于迄今为止我们讨论的都是线性操作, 故在任意静止背景中涉及一个或多个运动物体的一般情况下, 对应于静止图像分量的直流处傅里叶变换有一个波峰, 且这些位置的波峰与物体的速度成正比。

上述概念可以总结如下: 对于一个由 K 幅大小为 $M \times N$ 的数字图像序列, 在任何整数的瞬时点上, 图像投影到 x 轴上的加权求和为

$$g_x(t, a_1) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y, t) e^{j2\pi a_1 x \Delta t} \quad t = 0, 1, \dots, K-1 \quad (10.6-6)$$

类似地, y 轴上的投影和为

$$g_y(t, a_2) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x, y, t) e^{j2\pi a_2 y \Delta t} \quad t = 0, 1, \dots, K-1 \quad (10.6-7)$$

其中, 如已经解释过的那样, a_1 和 a_2 都是正整数。

式(10.6-6)和式(10.6-7)的一维傅里叶变换分别为

$$G_x(u_1, a_1) = \sum_{t=0}^{K-1} g_x(t, a_1) e^{-j2\pi u_1 t / K} \quad u_1 = 0, 1, \dots, K-1 \quad (10.6-8)$$

和

$$G_y(u_2, a_2) = \sum_{t=0}^{K-1} g_y(t, a_2) e^{-j2\pi u_2 t / K} \quad u_2 = 0, 1, \dots, K-1 \quad (10.6-9)$$

实际上, 如 4.11 节中讨论的那样, 这些变换的计算是使用 FFT 算法执行的。

频率和速度的关系是

$$u_1 = a_1 V_1 \quad (10.6-10)$$

和

$$u_2 = a_2 V_2 \quad (10.6-11)$$

在该式中, 速度的单位是像素数/总帧时间。例如, $V_1 = 10$ 可以解释为在 K 帧中物体运动了 10 个像素。对于匀速摄取的图像帧, 实际的物理速度取决于帧速率和像素之间的距离。因此, 如果 $V_1 = 10$, $K = 30$, 帧率是两幅图像每秒, 像素之间的距离是 0.5 m, 则 x 方向的实际物理速度是

$$V_1 = (10 \text{ 个像素}) (0.5 \text{ m/像素}) (2 \text{ 帧/s}) / (30 \text{ 帧}) = 1/3 \text{ m/s}$$

速度的 x 分量的符号可通过计算下式得到:

$$S_{1x} = \frac{d^2 \operatorname{Re}[g_x(t, a_1)]}{dt^2} \Big|_{t=n} \tag{10.6-12}$$

和

$$S_{2x} = \frac{d^2 \operatorname{Im}[g_x(t, a_1)]}{dt^2} \Big|_{t=n} \tag{10.6-13}$$

因为 g_x 是正弦函数, 故可以证明(见习题 10.47), 如果速度分量 V_1 为正, 则在时间 n , S_{1x} 和 S_{2x} 在任意点将有相同的符号。反之, S_{1x} 和 S_{2x} 的符号相反则表明速度分量 V_1 为负。如果 S_{1x} 或 S_{2x} 为零, 那么我们考虑下一个最近的时间点 $t = n \pm \Delta t$ 。类似的说明可用于计算 V_2 的符号。

例 10.28 通过频率域检测较小的运动物体。

图 10.61 到图 10.64 说明了刚才推导方法的有效性。图 10.61 显示了一组 32 帧 LANDSAT 图像序列中的一幅, 该图像是通过将白噪声添加到一幅参考图像中产生的。图像序列中包含一个叠加的运动物体, 物体在 x 方向以每帧 0.5 个像素的速度运动, 在 y 方向以每帧 1 个像素的速度运动。在图 10.62 中用圆圈标出的物体在展开的较小(9 像素)的区域内具有高斯灰度分布, 该分布用肉眼不易辨别。图 10.63 和图 10.64 显示了分别令 $a_1 = 6$ 和 $a_2 = 4$ 计算式(10.6-8)和式(10.6-9)时得到的结果。根据图 10.63 中 $u_1 = 3$ 处的波峰, 由式(10.6-10)得到 $V_1 = 0.5$ 。同样, 根据图 10.64 中 $u_2 = 4$ 处的波峰, 式(10.6-11)得到 $V_2 = 1.0$ 。

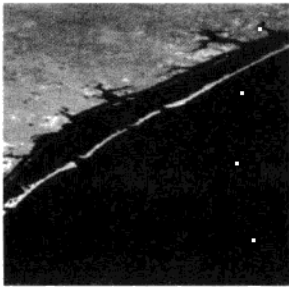


图 10.61 LANDSAT 的图像帧(原图像由 Cowart, Snyder 和 Ruedger 提供)

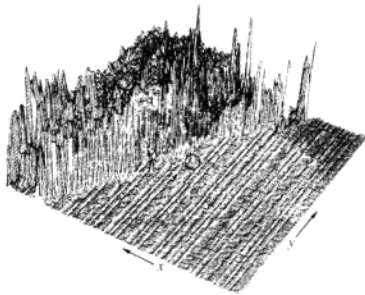


图 10.62 图 10.61 中图像的灰度图, 目标已用圆圈标出(原图像由 Rajala, Riddle 和 Snyder 提供)

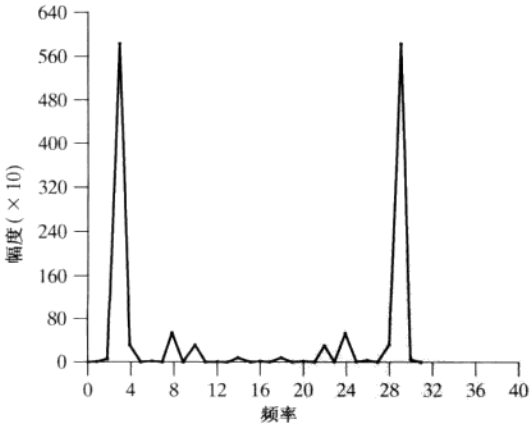


图 10.63 式(10.6-8)的谱, 在 $u_1 = 3$ 处显示了一个波峰(原图像由 Rajala, Riddle 和 Snyder 提供)

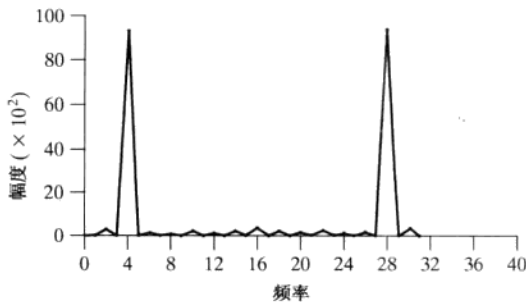


图 10.64 式(10.6-9)的谱, 在 $u_2 = 4$ 处显示了一个波峰(原图像由 Rajala, Riddle 和 Snyder 提供)

借助于图 10.63 和图 10.64, 我们可以解释选择 a_1 和 a_2 的原则。例如, 假设我们用 $a_2 = 15$ 取代 $a_2 = 4$ 。在这种情况下, 图 10.64 中的波峰将位于 $u_2 = 15$ 和 17 处, 因为 $V_2 = 1.0$, 这将是一种严重的混淆的结果。正如在 2.5.4 节中讨论过的那样, 混淆是由欠采样造成的(当 u 的范围由 K 决定时, 当前讨论中, 图像帧数太少)。因为 $u = aV$, 故一种可能是将 a 选择为最接近 $a = u_{\max}/V_{\max}$ 的整数, 其中 $u_{\max}$ 是由 K 建立的混淆频率限制, 而 $V_{\max}$ 是期望的物体的最大速度。

小结

在大多数自动图像模式识别和场景分析应用中, 图像分割是一个基本的预备步骤。正如在前面各节中介绍的各种例子所说明的那样, 选择何种分割技术主要取决于所考虑问题的特殊性。本章中讨论的各种方法尽管并不十分详尽, 但却是在实际应用中普遍使用的具有代表性的技术。下面的参考文献可作为读者进一步研究该主题的基础。

参考文献

由于分割在自动图像处理领域中所起的核心作用, 大多数关于图像处理、图像分析和计算机视觉的著述中都涵盖了这一主题。以下图书为这一主题提供了补充和/或辅助阅读材料: Umbaugh[2005]; Davies[2005]; Gonzalez, Woods 和 Eddins[2004]; Shapiro 和 Stockman[2001]; Sonka 等人[1999]; Petrou 和 Bosdogianni[1999]。

使用模板检测灰度的不连续性(见 10.2 节)已有较长的历史。这些年来, 人们提出了无数的模板: Roberts[1965], Prewitt[1970], Kirsh[1971], Robinson[1976], Frei 和 Chen[1977], 以及 Canny[1986]。Fram 和 Deutsch[1975]发表的回顾性文章中包含了大量的模板, 以及对它们的性能的评价。模板性能存在的问题, 特别是对边缘检测来说, 仍是相当重要的研究领域, 如由 Qian 和 Huang[1996], Wang 等人[1996], Heath 等人[1997, 1998]和 Ando[2000]给出的例子。彩色图像边缘检测在许多感知的应用中正逐渐普及, 例如, 可见 Salians, Abidi 和 Gonzalez[1996]的著述; Zugaj 和 Lattuati[1998]; Mirmehdi 和 Petrou[2000]; 以及 Plataniotis 和 Venetsanopoulos[2000]的著述。图像特征和模板性能之间的相互影响也是当前人们感兴趣的课题, 如 Ziou[2001]给出的例子。我们关于拉普拉斯算子的零交叉性质的介绍基于 Marr 和 Hildreth[1980]的论文和 Marr[1982]的专著。也可以见 Clark[1989]关于使用零交叉算法进行边缘鉴别的论文(Clark 的论文的部分修正由 Piech[1990]给出)。如 10.2 节中提到的那样, 通过高斯拉普拉斯的零交叉方法是一种重要的方法, 其相对性能仍是一个活跃的研究主题(Gunn[1998, 1999])。如其名字暗示的那样, 在 10.2.6 节讨论的坎尼检测器基于坎尼的论文[1986]。这一主题 20 年后的例子见 Zhang 和 Rockett[2006]。

霍夫变换(Hough[1962])是进行全局像素连接和曲线检测的一种实用方法。本章讨论的这种基本变换的大量推广已经提出很多年了。例如, Lo 和 Tsai[1995]讨论了检测细线的一种方法, Guil 等人[1995, 1997]解决了霍夫变换的快速实现和简单曲线的检测问题, Daul 等人[1998]将霍夫变换进一步推广到了检测椭圆形弧线, Shapiro[1996]则解决了对灰度图像实现霍夫变换的问题。

如 10.3 节开始时提到的那样, 阈值处理技术享有重要的普及地位, 因为这种技术实现简单。关于该主题的许多文献中报告了大量的工作并不奇怪。在 Sahoo 等人[1988]和 Lee 等人[1990]的综述性的文章中, 可以看到对这类文献的较好评价。除了本章中讨论的技术外, 处理光照和反射影响(10.3.1 节)所使用的其他方法, 由 Perez 和 Gonzalez [1987], Parker[1991], Murase 和 Nayar[1994], Bischsel[1998], Drew 等人[1999]和 Toro 和 Funt[2007]的工作做了说明。关于 10.3.2 节中的其他读物, 见 Jain 等人[1995]的著述。

最优全局阈值处理(10.3.3 节)的早期工作已在 Chow 和 Kaneko[1972]的经典著述中举例说明(我们将在 12.2.2 节在目标识别的更一般的行文来讨论这个方法)。虽然在理论上它是最佳的, 但这种方法在灰度阈值处理中的应用却受到了限制, 因为需要估计概率密度函数。10.3.3 节中给出的出自 Otsu[1979]的最佳方法

得到了人们的广泛接受,因为它使优秀的性能和实现简单结合起来,且仅要求估计图像的直方图。使用预处理(10.3.4节和10.3.5节)的基本概念可回溯至 White 和 Rohrer[1983]的早期论文,该论文在解决困难的分割问题中,综合使用了阈值处理、梯度和拉普拉斯算子。前面三篇文章中讨论的方法,与由 Cheriet 等人[1998], Sauvola 和 Pietikainen[2000], Liang 等人[2000]和 Chan 等人[2000]在阈值处理方面所做的工作几乎相差了近 20 年,因此在图像分割能力方面比较这些方法的基本相似性是很有趣的。关于多阈值处理(见 10.3.6 节)的其他读物,见 Yin 和 Chen[1997], Liao 等人[2001]和 Zahara 等人[2005]的著述。关于可变阈值处理(见 10.3.7 节)的其他读物,见 Parker[1997]和 Delon 等人[2007]的著述。

面向区域分割这一主题的早期的著述见 Fu 和 Mui[1981]的著述。为分割目的而综合区域和边界信息的早期努力见 Haddon 和 Boyce[1990]和 Pavlids 和 Liow[1990]的工作。由 Hojjatoleslami 和 Kittler[1998]提出的一种较新区域生长方法也很重要。对于最近一段时间基本涵盖面向区域分割概念的文献,见 Shapiro 和 Stockman[2001]以及 Sonka 等人[1999]的著述。

10.5 节所示的分水岭分割是一个强大的概念。涉及分水岭分割的早期文献见 Serra[1988], Beucher[1990]和 Beucher 和 Meyer[1992]的著述。Baccar 等人[1996]的论文讨论了基于数据融合和形态学分水岭的分割方法。完全致力于分水岭分割的专刊 *Pattern Recognition*[2000]表明,10 年来该领域的研究进展相当明显。如我们在 10.5 节中讨论的那样,使用分水岭方法的关键问题之一是过度分割问题。Najmanand 和 Schmitt[1996], Haris 等人[1998]和 Bleau 和 Leon[2000]的论文是处理这种问题的例证。Bieniek 和 Moga[2000]讨论了一种基于连通分量的分水岭分割算法。

10.6.1 节中的材料来自 Jain, R.[1981]的著述,也可参阅 Jain, Kasturi 和 Schunck[1995]的著述。10.6.2 节中的材料来自 Rajala, Riddle 和 Snyder[1983]的著述,也可参阅 Shariat 和 Price[1990]及 Cumani 等人[1991]的论文。关于运动估计的其他读物,见 Sonka 等人[1999], Shapiro 和 Stockman[2001], Snyder 和 Qi[2004]及 Davies[2005]的书籍,也可参阅 Alexiadis 和 Sergiadis[2007]的著述。

习题

- ★10.1 证明式(10.2-2)的正确性。(提示:用泰勒级数展开,并仅保留线性项。)
- ★10.2 设有一幅包含有水平直线、垂直直线、 45° 方向直线和 -45° 方向直线的二值图像。给出一组大小为 5×5 的模板,要求这些模板可用于检测这些直线中的 1 像素的间断。假设直线和背景的灰度分别是 1 和 0。
- 10.3 提出一种用于检测二值图像的线段中长度从 1 到 K 个像素的裂缝的技术。假设线的宽度为 1 个像素。您的技术要以 8 邻域连通性分析为基础,而不能试图构造检测裂缝的模板。
- 10.4 参考图 10.7, 回答下列问题。
 - ★(a) 图 10.7(c)中某些连接焊点和中心元素的线是单线,而其他线是双线。解释一下为什么。
 - (b) 提出一种消除图 10.7(f)中那些 -45° 方向线的部分分量的方法。
- 10.5 参考图 10.8 中的边缘模型。
 - ★(a) 假设我们使用图 10.14 中的 Prewitt 算子来计算每个模型的梯度幅度。请画出通过梯度图像中心的水平剖面线。
 - (b) 画出每个对应角度图像的水平剖面。
 (注意:在回答该问题时不要生成梯度图像和角度图像。简单地提供剖面线的图,该图显示像您预期的幅度和角度的剖面线即可。)
- 10.6 考虑穿过一幅二值图像中部的一个水平灰度剖面线,二值图像中包含一个垂直通过图像中心的台阶边缘。画出已被一个大小为 $n \times n$ 且系数等于 $1/(n^2)$ 的平均模板模糊后图像的剖面线。为简单起

见, 假设图像已被标定, 即其灰度级在边缘的左侧为 0, 在右侧为 1。此外, 假设模板的尺寸远小于图像的尺寸, 以便在靠近水平灰度剖面的中心处不必关注图像的边界效应。

★10.7 假设我们已使用示于右图中的边缘模型代替了图 10.10 中的斜坡模型。请写出每个剖面的梯度和拉普拉斯算子。

10.8 参照图 10.14, 回答下列问题。

(a) 假设使用 Sobel 模板得到了 g_x 和 g_y 。证明在这种情况下, 使用式 (10.2-10) 和式 (10.2-20) 计算得到的梯度幅度可得出相同的结果。

(b) 证明对于 Prewitt 模板也是正确的。

★10.9 证明图 10.14 和图 10.15 中的 Sobel 模板和 Prewitt 模板仅对水平边缘、垂直边缘和 $\pm 45^\circ$ 方向边缘给出各向同性的结果。

10.10 某些使用二维模板处理图像一遍得到的结果, 也可使用一维模板处理图像两遍来得到。例如, 使用系数为 $1/9$ 、大小 3×3 的平滑模板得到的结果, 可以通过对图像使用一次模板 $[1, 1, 1]$, 然后对结果应用模板

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

然后, 最终结果乘以 $1/9$ 。证明 Prewitt 模板 (见图 10.14) 的响应可通过使用一次形如 $[-1, 0, 1]$ 的差分模板 (或其对应的正交阵), 然后使用一次形如 $[1, 2, 1]$ 的平滑处理模板 (或其对应的正交阵) 得到。

10.11 所谓的大小为 3×3 的罗盘梯度算子设计用来度量 8 个方向上的梯度: E, NE, N, NW, W, SW, S 和 SE。

★(a) 使用系数值 0, 1, -1, -2 或 2, 给出这 8 个算子的形式。

(b) 指定每个模板的梯度向量方向, 记住梯度方向与边缘方向是正交的。

10.12 右图所示二值图像中, 矩形的大小为 $m \times n$ 像素。

(a) 以式 (10.2-20) 给出的近似为基础, 这幅图像的梯度幅度像什么?

假设 g_x 和 g_y 是使用 Prewitt 算子得到的。说明梯度图像中的所有相应的不同像素值。

(b) 画出用式 (10.2-11) 计算得到的边缘方向的直方图。精确标出直方图中每个分量的高度。

(c) 以式 (10.2-7) 给出的近似为基础, 该图像的拉普拉斯图像是什么? 说明拉普拉斯图像中所有相应的不同像素值。

10.13 假设图像 $f(x, y)$ 与大小为 $n \times n$ 的模板 (系数为 $1/n^2$) 卷积后, 产生了一幅平滑后的图像 $\bar{f}(x, y)$ 。

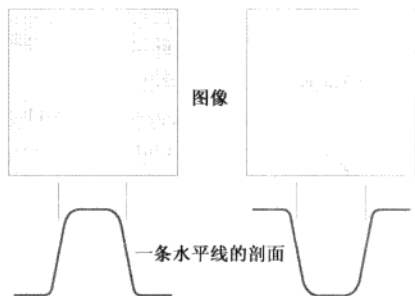
★(a) 以模板尺寸的函数的形式推导平滑后图像的边缘强度 (边缘幅度) 的一个表达式。为简单起见, 假设 n 是奇数, 且边缘是使用如下偏导数得到的:

$$\frac{\partial \bar{f}}{\partial x} = \bar{f}(x+1, y) - \bar{f}(x, y) \quad \text{和} \quad \frac{\partial \bar{f}}{\partial y} = \bar{f}(x, y+1) - \bar{f}(x, y)$$

(b) 证明平滑后的图像的最大边缘强度与原图像的最大边缘强度之比为 $1/n$ 。换句话说, 边缘强度与平滑模板的尺寸成反比。

10.14 参考式 (10.2-23):

★(a) 证明一个高斯算子的拉普拉斯 $\nabla^2 G(x, y)$ 为零。



图像

一条水平线的剖面



- (b) 证明用这个算子卷积的任何图像的平均值也为零。(提示: 考虑在频率域中解决该问题, 使用卷积定理, 以及一个函数的平均值与其在原点处计算的傅里叶变换成正比这一事实)。
- (c) 在一般情况下, 如果我们(1)使用图 10.4(a)中的模板来计算使用大小为 3×3 的拉普拉斯模板的高斯低通滤波器的拉普拉斯, 并(2)将得到的结果与任何图像卷积, 那么请问(b)还正确吗? 请解释。(提示: 参见习题 3.16。)

10.15 参考图 10.22(c)。

- (a) 解释为什么边缘构成闭合的轮廓线。

★(b) 寻找边缘位置的零交叉方法总会导致闭合轮廓吗? 请解释。

10.16 我们时常发现文献中推导高斯拉普拉斯(LoG)时, 都用如下表达式开始:

$$G(r) = e^{-r^2/2\sigma^2}$$

式中 $r^2 = x^2 + y^2$ 。然后, 用二阶偏导数得到 LoG: $\nabla^2 G(r) = \partial^2 G / \partial r^2$ 。最后, 使用 $x^2 + y^2$ 替代 r^2 得到了(不正确的)结果:

$$\nabla^2 G(x, y) = [(x^2 + y^2 - \sigma^2) / \sigma^4] \exp[-(x^2 + y^2) / 2\sigma^2]$$

请推导该结果并解释该表达式与式(10.2-23)不同的原因。

10.17 (a) 推导式(10.2-27)。

- (b) 令 $k = \sigma_1 / \sigma_2$ 表示 DoG 函数的标准差比率。请使用 k 和 σ_2 来表示式(10.2-27)。

10.18 在下文中, 假设 G 和 f 分别是大小为 $n \times n$ 和 $M \times N$ 的离散阵列。

- ★(a) 证明式(10.2-21)中的高斯函数与图像 $f(x, y)$ 的二维卷积可以表示为先沿 $f(x, y)$ 的行(列)进行一维卷积, 然后沿所得结果的列(行)进行一维卷积(关于离散卷积的讨论, 见 3.4.2 节)。
- (b) 推导出使用(a)中的一维卷积方法相对于直接执行二维卷积的优势的表达式。假设 $G(x, y)$ 被取样产生一个大小为 $n \times n$ 的阵列, 而 $f(x, y)$ 是一个大小为 $M \times N$ 的阵列。计算优势是二维卷积要求的乘法次数与一维卷积要求的乘法次数之比。

10.19 (a) 证明 Marr-Hildreth 算法的步骤 1 和步骤 2 可以通过使用 4 次一维卷积来实现。(提示: 参考习题 10.18(a), 将拉普拉斯算子表示为由式(10.2-5)和式(10.2-6)给出的两个偏导数之和, 如习题 10.10 那样, 并使用一个一维模板实现每个导数。)

- (b) 推导出使用(a)中的一维卷积方法相对于直接执行二维卷积的优势的表达式。假设 $G(x, y)$ 被取样产生一个大小为 $n \times n$ 的阵列, 而 $f(x, y)$ 是一个大小为 $M \times N$ 的阵列。计算优势是二维卷积要求的乘法次数与一维卷积要求的乘法次数之比(见习题 10.18)。

10.20 (a) 用一维卷积代替二维卷积, 用公式表示坎尼算法的步骤 1 和第二步中梯度幅度图像的计算。

- (b) 相对于使用二维卷积实现, 使用一维卷积方法的计算优势是什么? 假设步骤 1 中的二维高斯滤波器被取样为一个大小为 $n \times n$ 的阵列, 且输入图像的大小为 $M \times N$ 。将计算优势表示为每种方法所要求的乘法次数之比。

10.21 参考图 10.8 中的三个垂直边缘模型和对应的剖面。

- ★(a) 假设我们使用 Sobel 模板来计算三个边缘模型的梯度幅度。画出三幅梯度图像的水平灰度剖面线。
- ★(b) 画出三幅拉普拉斯图像的水平灰度剖面线, 假设拉普拉斯是使用图 10.4(a)中的 3×3 模板计算的。
- ★(c) 仅使用 Marr-Hildreth 边缘检测器的前两步产生的图像, 重复上一问题。
- (d) 对于坎尼边缘检测器的前两步重复上述问题。您可以忽略角度图像。

(e) 针对坎尼边缘检测器, 画出角度图像的水平剖面线。(注意: 回答这一问题时不必生成图像, 只须简单地画出剖面线。并说明您所期望的图像剖面线像什么。)

10.22 参考 10.2.7 节中讨论的霍夫变换。

(a) 从 $y = ax + b$ 的斜截式形式, 导出一条线的法线表示的一般过程。

★(b) 求直线 $y = -3x + 2$ 的法线表示。

★**10.23** 参考 10.2.7 节中讨论的霍夫变换。

(a) 解释为什么图 10.33 (a) 中的点 1 的霍夫映射在图 10.33 (b) 中为一条直线。

(b) 这是仅能产生这样结果的点吗? 请解释。

(c) 以图 10.33 (b) 中标为 Q 的曲线为例, 解释反射的邻接关系。

10.24 证明实现 10.2.7 节所述累加单元方法所需的操作次数与图像平面 (即 xy 平面) 中的背景点数量 n 成线性关系。

10.25 图像分割技术的重要应用领域是处理由所谓的泡沫室活动导致的图像。这些图像是在高能物理实验中得到的, 在这种实验中, 一束已知性质的粒子直接打到一个已知核子的目标上。一个典型的活动由任何一次碰撞活动中的输入轨迹以及粒子由撞击点散发出的二次轨迹组成。请提出一种分割方法来检测所有的轨迹, 这些轨迹至少包含 100 个像素并与水平方向有如下 6 个方向的夹角: $\pm 20^\circ$, $\pm 40^\circ$ 和 $\pm 60^\circ$ 。在这 6 个方向上的每一个, 允许的估计误差为 $\pm 5^\circ$ 。一条有效的轨迹必须至少要有 100 个像素长, 间隙不多于 3 个, 并且每个间隙不超过 10 个像素的长度。您可以假设图像已进行了预处理以便形成二值图, 并且除了发出轨迹的撞击点以外, 所有轨迹均为 1 个像素宽。您的过程应能分辨原点不同但方向相同的轨迹。(提示: 以霍夫变换作为您的解决方案的基础。)

★**10.26** 重新说明 10.3.2 节中的全局阈值处理算法, 以便它使用一幅图像的直方图来代替该图像本身。

★**10.27** 证明 10.3.2 节中的基本全局阈值处理算法在有限几步内收敛。(提示: 使用来自习题 10.26 的直方图表达式。)

10.28 给出一种解释, 为什么 10.3.2 节的基本全局阈值处理算法的初始阈值必须在图像的最小值和最大值之间。(提示: 构建一个例子来说明阈值选在该范围之外算法会失败。)

★**10.29** 使用 10.3.2 节的基本全局阈值处理算法得到的阈值与起始点无关吗? 如果您的回答是“是”, 证明它, 如果回答是“不”, 给出一个例子。

10.30 您可以假设下列两种情况, 在迭代中阈值限制在开区间 $(0, L-1)$ 内。

★(a) 如果图像的直方图在所有可能的灰度级上都是均匀的, 请证明 10.3.2 节的基本全局阈值处理算法将收敛到图像的平均灰度 $(L-1)/2$ 处。

(b) 如果一幅图像的直方图是双模式的, 且具有关于其均值对称的相同模式, 请证明基本全局阈值处理算法将收敛到这两个模式的均值之间的中点。

10.31 参考 10.3.2 节中的阈值处理算法。假设在给定的问题中, 直方图是双模式的, 且这两个模式是形如 $A_1 \exp[-(z-m_1)^2/2\sigma_1^2]$ 和 $A_2 \exp[-(z-m_2)^2/2\sigma_2^2]$ 的高斯曲线。假设 $m_1 > m_2$, 且初始的 T 在图像的最大灰度值和最小灰度值之间。当算法收敛时, 给出以下陈述为真的条件 (用这些曲线的参数表示):

(a) 阈值等于 $(m_1 + m_2)/2$ 。

(b) 阈值在 m_2 的左边。

(c) 阈值位于区间 $(m_1 + m_2)/2 < T < m_1$ 内。

如果这些条件的任意一条不存在, 请说明原因。

★**10.32** (a) 证明式 (10.3-15) 的第一行来自式 (10.3-14)、式 (10.3-10) 和式 (10.3-11)。

(b) 证明式 (10.3-15) 的第二行来自第一行。

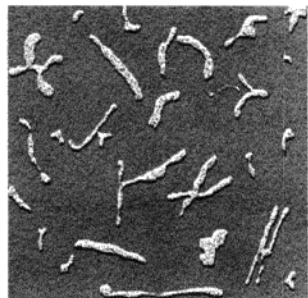
10.33 证明对于范围 $0 \leq k \leq L-1$ 内的 k 值, 式 (10.3-18) 总存在一个最大值。

10.34 参考式 (10.3-20), 预设一个参量, 可确定 $0 \leq \eta(k) \leq 1$, k 在 $0 \leq k \leq L-1$ 范围内, 仅在图像具有恒定灰度时可达最小值, 并且仅在图像有两个值 0 和 $L-1$ 时最大值才发生。

★ **10.35** (a) 假设一幅图像 $f(x, y)$ 的灰度在范围 $[0, 1]$ 内, 并且阈值 T 成功地将该图像分割为物体和背景。证明阈值 $T' = 1 - T$ 将成功地把 $f(x, y)$ 的负图像分割为相同的区域。这里, “负”这一术语被用于 3.2.1 节定义的场景。

(b) 在 (a) 中, 把一幅图像映射为它的负图像的灰度变换函数是一个具有负斜向的线性函数。说明一个任意的灰度变换函数必须满足关于将被保留的阈值 T 对原始图像可分割性的条件。灰度变换后的阈值将是什么值?

10.36 右图所示图像中的物体和背景, 在标度范围 $[0, 255]$ 内具有的平均灰度分别为 180 和 70。该图像被均值为 0、标准差为 10 个灰度级的高斯噪声污染了。请提出一种正确分割率为 90% 或更高百分比的阈值处理方法。(回忆一下, 高斯曲线下 99.7% 的面积位于均值的 $\pm 3\sigma$ 区间内, 其中 σ 是标准差。)

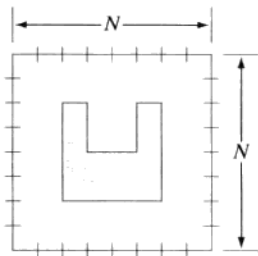


10.37 参考图 10.37 (b) 中的灰度斜坡图像和 10.3.7 节中讨论的移动平均算法。假设图像的尺寸为 400×700 像素, 其最小值和最大值分别为 0 和 1, 这里的 0 项仅包含在第一列中。

- ★ (a) 若 $b=0$ 而 n 为任意值, 那么使用移动平均算法分割该图像后的结果是什么? 解释该图像像什么。
- (b) 现在, 斜坡的方向反转, 因此, 最左侧的值为 1 而最右侧的值为 0。重复问题 (a)。
- (c) 重复问题 (a), 但 $n=4, b=1$ 。
- (d) 重复问题 (a), 但 $n=100, b=1$ 。

10.38 提出一个区域生长算法来分割习题 10.36 中的图像。

★ **10.39** 使用 10.4.2 节中讨论过的分裂和聚合过程来分割右图所示的图像。如果 R_i 中的所有像素都有相同的灰度, 则令 $Q(R_i) = \text{TRUE}$ 。给出对应于您的分割的四叉树。



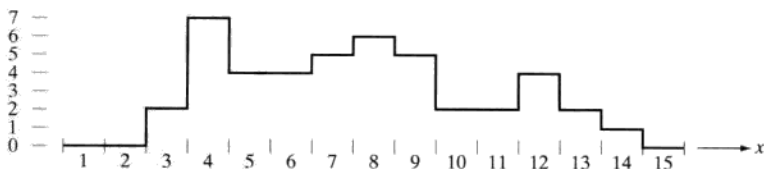
10.40 考虑例 10.24 中对天鹅星座环图像的稀疏区域进行分割得到的由 1 构成的区域。提出一种将该区域作为模板来隔离图像中的三个主要分量的技术: (1) 背景, (2) 致密内部区域, (3) 稀疏外部区域。

10.41 参考 10.5.3 节中的讨论。

- ★ (a) 说明在分水岭算法执行的过程中, $C_n(M_i)$ 和 $T[n]$ 的元素从不被替换。
- (b) 说明当 n 增加时, 集合 $C_n(M_i)$ 和集合 $T[n]$ 中的元素数量随着 n 的增加, 不是增加, 就是保持不变。

10.42 在 10.5 节说明过的使用分水岭分割算法得到的边界形成了一个闭合环 (例如, 见图 10.56 和图 10.58)。请预设一个参量, 使得应用这个算法无论如何总能得到闭合形式的边界。

★ **10.43** 对于下图所示的一维灰度横截面, 给出逐步构建水坝的实现过程。在每一步骤画出显示水位和已构建大坝的横截面。



10.44 如果我们在式(10.6-4)中是针对 T (而不是针对 $-T$) 进行测试, 那么图 10.59(c) 中的负 ADI 图像会是什么样子?

10.45 判断下列说法是否正确。对您的答案进行解释。

- ★ (a) 只要物体正在运动, 那么绝对 ADI 中的非零项的维数就会持续增加。
- (b) 不管物体是否运动, 正 ADI 中的非零项总是占据相同的面积。
- (c) 只要物体正在运动, 负 ADI 的非零项的维数就会持续增加。

10.46 假设在例 10.28 中将 x 轴方向的运动设为零。现在物体仅沿 y 轴以每帧 1 个像素的速度运动 28 帧, 然后瞬间转向, 向相反方向运动 28 帧。在这些条件下, 图 10.63 和图 10.64 会是什么样子?

★ **10.47** 预置一个参量, 以证明当式(10.6-12)和式(10.6-13)中的 S_{1x} 和 S_{2x} 的符号相同时, 速度分量 V_1 为正。

10.48 一家自动制药厂为质量控制的目的用图像处理方法来测量药片的形状。该系统的分割步骤是以 Otsu 方法为基础的。由于检测线的速度很高, 因此需要速率非常高的闪光来“停止”运动。当照明灯为新的时, 照明灯投影一个均匀的光模式。但当灯泡老化时, 照明模式的退化根据下面的公式是时间和空间坐标的函数:

$$i(x, y) = A(t) - t^2 e^{-[(x-M)^2 + (y-N)^2]}$$

其中, (M, N) 是观察区域的中心, t 是以月为单位增加的测试时间。灯泡是试验性的, 且制造商并不了解灯光的 $A(t)$ 特性。所知道的全部是在灯泡的寿命期, $A(t)$ 总是大于上式中的负值, 因为光照不能为负。观察表明, 当灯泡是新的时, Otsu 算法工作得很好, 其光照模式在整个图像上近似于常数。然而, 分割性能却随时间恶化。由于试验性灯泡格外昂贵, 因此请您作为顾问在计算上帮助解决该问题, 从而延长灯光的可用寿命。给您很大的灵活性, 可以在成像摄像机的取景区域的边缘附近安装任何特殊标记或其他可视提示。请提出一个能让工程计划经理理解的详细解决方案。(提示: 回顾 2.3.4 节中讨论的图像模型并考虑使用一个其反射率已知的小目标。)

10.49 飞行中的子弹的速度可以用成像技术来估计。选择一种包括使用一台 TV 摄像机和可以曝光场景 k 秒的闪光灯的方法。子弹长 2.5 cm, 宽 1 cm, 速度范围为 700 ± 200 m/s。光学摄像机拍摄了一幅水平分辨率为 256×256 的数字图像, 其中子弹占据了 10%。

- ★ (a) 为保证运动对图像造成的模糊不超过 1 像素, 确定最大的 K 值。
- (b) 确定每秒必须得到的帧数的最小值, 以便保证在子弹穿过摄像机的视场时至少得到两幅子弹的完整图像。
- (c) 提出一种能自动地从帧序列中提取子弹的分割过程。
- (d) 提出一种能自动地确定子弹速度的方法。