

# 第5章

## 特征检测与匹配

本章主要介绍几种图像的特征,包括边缘、线和角点,以及它们的检测与匹配方法。5.1节主要介绍图像边缘的概念,并介绍常用的图像边缘检测方法。5.2节将关注图像中直线的这一重要特征,并在图像中拟合参数化的直线。5.3节将介绍如何用 Harris 角点检测寻找图像中的角点特征,并讲述其背后的数学思想。5.4节将介绍尺度不变特征转换(SIFT)、多尺度带方向的块描述符(MOPS)和定向梯度直方图(HOG)等特征描述子,这些描述子可以更好地表示角点的特征,提高图像特征点提取的鲁棒性。5.5节将介绍几种方法来使用这些特征点的描述子在图像之间进行高效率的匹配,包括基于局部敏感哈希(LSH)和基于KD树的方法。此外,还会介绍如何根据对齐匹配的特征点来对齐图像。

## 5.1 边缘检测

边缘检测是图像处理和计算机视觉中的基本问题,边缘检测的目的是标识数字图像中灰度值变化明显的点。这些边缘点在图像中大量存在,并且往往携带了重要的语义信息。例如,物体的边界或者三维空间中物体的相互遮挡往往在图像中以边缘的方式体现,还有一些边缘可能表示物体表面方向的变化或者是物体上的折痕,此外图像中的边缘还可能由场景光照的变化或者物体的阴影产生。我们可以通过简笔画勾勒出的物体轮廓轻松地识别出所熟知的物品,由此可见,图像边缘保留了图像中的重要结构属性,检测图像的边缘可以大幅减少图像中的数据量。边缘检测算法在图像分割和图像的特征提取等领域都有重要的应用,5.1.1节介绍几种边缘模型和边缘检测算法的基本步骤,5.1.2节介绍几种用于计算图像梯度的边缘检测算子,5.1.3节介绍具有广泛应用的Canny边缘检测算法。

### 5.1.1 基本原理

在检测边缘之前,本节首先介绍根据灰度值剖面进行分类的边缘模型,包括台阶边缘模型、斜坡边缘模型和屋顶边缘模型。台阶边缘是指在一个像素的距离上发生两个灰度级之间的理想突变。图5.1(a)展示的是一个垂直台阶边缘和通过该边缘的一个水平剖面。这种台阶边缘往往出现在由计算机生成的图像里。这种能够在一个像素的距离就能出现的清晰理想的边缘,只能在没有额外处理(如平滑处理)使其变得更“真实”的条件下见到。

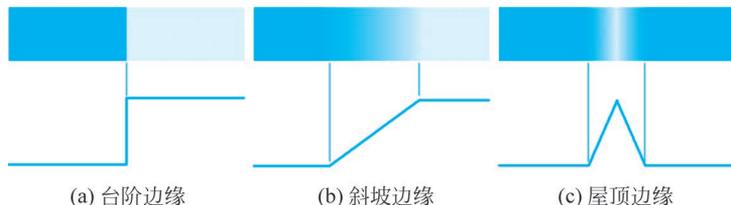


图 5.1 台阶边缘、斜坡边缘和屋顶边缘模型的理想表示以及它们的灰度剖面

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相机聚焦机理(如光学成像中的镜头)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,将边缘建模成斜坡形的灰度剖面更为贴切,如图 5.1(b)所示。斜坡的斜率与边缘的模糊程度成反比。在这一模型中,灰度剖面中不再有单一的“边缘点”。此时斜坡上的任意点都会被认为是边缘点,进而可以得到一个由这些边缘点连接而成的“边缘线段”。

图 5.1(c)展示了屋顶边缘的灰度剖面特征。屋顶边缘建模了深色背景区域的一条亮线,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,当其基底为一个像素宽时,屋顶边缘只不过是一条穿过图像中一个区域的一个像素宽的线。例如,在采集场景的深度图时,当细物体(如管子)比它的背景(如墙)更接近深度传感器时会出现屋顶边缘。此外,在卫星图像中道路这种较细的目标也可以被建模成此屋顶边缘。

下面从图像输入与输出的角度分析边缘检测算法。边缘检测算法接收一个灰度图像作为输入,将一个二值图像作为输出,其中用 1 标记出了灰度值变化明显的点,即边缘点。边缘检测算法通常有以下三个步骤:

- (1) 对图像进行平滑处理以消除噪声。噪声点也会造成局部灰度值明显的变化,产生虚假的边缘。
- (2) 边缘点的检测。通过局部信息找到灰度值变化明显的点作为候选边缘点。
- (3) 边缘定位。其主要目的是从上一步的候选边缘点中筛选出组成边缘集合的真实成员。

### 5.1.2 边缘检测算子

三种边缘模型可以表示为灰度值关于剖面位置的一元函数,用导数可以描述此函数在某一点附近的变化率。图 5.2(a)展示的是斜坡边缘,图 5.2(b)是此边缘对应的水平灰度剖面线以及它的一阶和二阶导数。沿着灰度剖面从左到右移动时,在斜坡开始处和在斜坡上的各个点处一阶导数为正,在恒定灰度区域处一阶导数为零。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导数的符号正好相反。零灰度轴和二阶导数极值间连线的交点称为该二阶导数的零交叉点。

由上面的观察可以得出结论:一阶导数的幅度可以用于检测图像中的某个点是否处在一个边缘。同样,二阶导数的符号可以用于确定一个边缘点是位于边缘的暗一侧还是边缘的亮一侧。围绕一条边缘的二阶导数的两个性质:一是对于图像中的每条边缘,二阶导数生成两个值;二是二阶导数的零交叉点可以用于定位粗边缘的中心。尽管上述分析都是限制在垂直边缘的一维水平剖面上,类似的结论同样适用于图像任何方向的边缘。

前面结合图 5.2 讨论了无噪声斜坡边缘和它的一阶和二阶导数的性质,下面结合图 5.3 对有噪声的情况进行分析。图 5.3 中第一列图像是四个斜坡边缘的特写,这些边缘从左边的黑色区域到右边的白色区域过渡。第一列的第一个图像没有噪声,而其他三

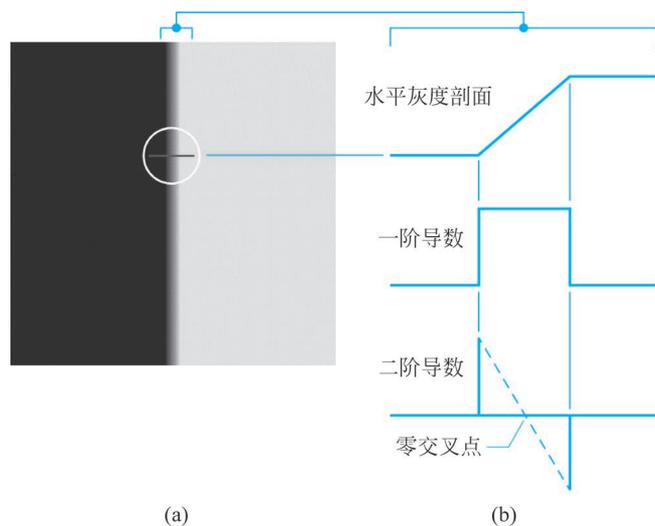


图 5.2 无噪声斜坡边缘及其一阶、二阶导数

注：图(a)是由一条垂直斜坡边缘分开的两个恒定灰度区域；图(b)为边缘附近的细节，显示了水平灰度剖面及其一阶导数和二阶导数。

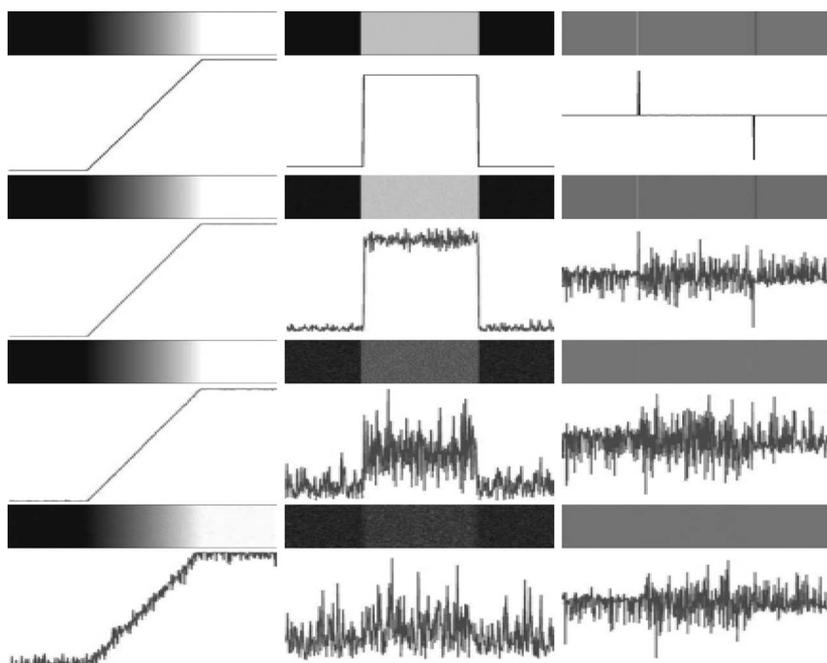


图 5.3 第一列：被均值为零，标准差分别为 0.1、1.0、10.0 三个级别随机高斯噪声污染的斜坡边缘图像和灰度剖面。第二列：一阶导数图像和灰度剖面线。第三列：二阶导数图像和灰度剖面线

个图像从上到下被均值为零,标准差分别为 0.1、1.0 和 10.0 灰度级的加性高斯噪声污染。每个图像段下面的图形是通过图像中心的水平灰度剖面线。

对第一列图像从上到下的每个水平灰度剖面求一阶导数,并用灰度表示导数值,可以得到第二列图像。在斜坡图像灰度值恒定的区域每个灰度剖面的每个点导数均为零,体现在导数图像上就是两端的黑色(灰度值为零)条带。在斜坡上各点导数恒定并且等于斜坡的斜率,在导数图像上体现为灰色。当继续观察中间一列下面三幅图像时,随着噪声的增加,导数图像与无噪声时的情况差别越来越大。

第三列图像是第一列求二阶导数得来,从中可以看出,二阶导数对噪声更为敏感。第三列第一个图像是没有噪声时的二阶导数图像,其中白色和黑色的细垂直线是二阶导数的正负分量,通过比例缩放使得灰色的区域表示零。对于下面三个有噪声的图像,可以发现只有第一个图像能够清晰辨认出二阶导数的正、负两个分量,而其余两个则很难分辨。

因此可以发现,即便是很微弱的噪声对检测边缘所需的两个关键导数值都会产生严重的影响。当运用导数检测有着类似上述噪声级别图像的边缘时,应该优先对图像进行平滑处理。故而在 5.1.1 节中边缘检测算法的描述中需要将图像平滑放在第一步。

上面讨论了导数在检测边缘中起到的重要作用。如果把图像看成一个二元函数  $f(x, y)$ ,定义图像左上角为原点, $x$  轴方向竖直向下, $y$  轴方向水平向右。类似于一元函数的导数,可以用梯度来描述二元函数表面的斜率和方向。函数的梯度可以用  $\nabla f$  表示,并用向量来定义:

$$\nabla f(x, y) \equiv \begin{bmatrix} g_x(x, y) \\ g_y(x, y) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} \quad (5.1)$$

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。图像中每个有效位置可以计算  $\nabla f(x, y)$  并生成一个向量图像,即图像中的每个元素都是梯度向量。定义其幅值(或大小)为梯度向量  $\nabla f(x, y)$  的欧几里得范数  $M(x, y)$ :

$$M(x, y) = \|\nabla f(x, y)\| = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \quad (5.2)$$

这个值表示沿着梯度方向图像灰度的变化率。

梯度向量的方向角度由下式定义:

$$\alpha(x, y) = \arctan \left[ \frac{g_x(x, y)}{g_y(x, y)} \right] \quad (5.3)$$

任意点  $(x, y)$  处的边缘切线方向与该点处的梯度向量方向  $\alpha(x, y)$  正交。

前面的讨论都是将图像看成连续的二元函数,但实际上计算机存储图像时都是用离散的灰度矩阵。因此,常用前向差分来对两个方向上的偏导数进行近似:

$$g_x(x, y) = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y) \quad (5.4)$$

$$g_y(x, y) = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y) \quad (5.5)$$

这两个等式可以使用图 5.4 所示的一维模板(kernel)对图像进行滤波实现。

当检测对角线方向的边缘时,通常需要二维模板进行操作。Roberts 交叉梯度算子是最早尝试使用具有对角优势的二维模板之一。考虑图 5.5(a)中的  $3 \times 3$  区域,Roberts 算子以求对角像素之差为基础:

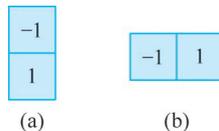


图 5.4 用于计算式(5.4)和式(5.5)的一维模板

$$g_y(x, y) = \frac{\partial f(x, y)}{\partial y} = z_9 - z_5 \quad (5.6)$$

$$g_x(x, y) = \frac{\partial f(x, y)}{\partial x} = z_8 - z_6 \quad (5.7)$$

式(5.6)和式(5.7)分别可以使用图 5.5(b)和(c)中的模板对图像进行滤波来实现。

$2 \times 2$  的模板很简单,在计算边缘的方向往往不像关于中心对称的模板那么有效<sup>[1]</sup>,而最小的中心对称的模板是  $3 \times 3$ 。这些模板考虑了关于中心对称的数据的性质,因而可以提取更多有关边缘方向的信息。下面的式子用  $3 \times 3$  的模板来对偏导数进行近似:

$$g_y(x, y) = \frac{\partial f(x, y)}{\partial y} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \quad (5.8)$$

$$g_x(x, y) = \frac{\partial f(x, y)}{\partial x} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \quad (5.9)$$

在上述公式中, $3 \times 3$  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。这两个偏导的近似式比 Roberts 算子更为精确。式(5.8)和式(5.9)可用图 5.5(d)和(e)中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。

对 Prewitt 算子的中心系数上乘一个权值 2,可得

$$g_y(x, y) = \frac{\partial f(x, y)}{\partial y} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (5.10)$$

$$g_x(x, y) = \frac{\partial f(x, y)}{\partial x} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (5.11)$$

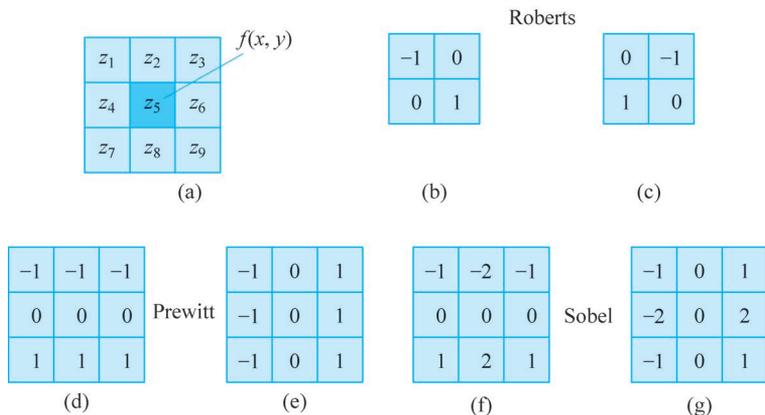


图 5.5 不同类型的边缘检测算子

注:图(b)和图(c)分别为 Roberts 垂直算子和水平算子;图(d)和图(e)为 Prewitt 算子;图(f)和图(g)为 Sobel 算子。

这个变化可以平滑图像。图 5.5(f)和(g)给出了用于实现式(5.10)和式(5.11)的模板,这两个模板称为 Sobel 算子。Sobel 算子根据像素与中心点的位置进行加权,起到降低边缘模糊程度的作用。

通过 Sobel 算子计算出图像梯度后,选择梯度幅值大于某一阈值的点作为边缘点。在实际应用中使用式(5.2)计算幅值时,平方和的平方根的计算导致时间开销太大,可以使用绝对值来近似梯度的幅值以减少计算量:

$$M(x,y) \approx |g_x| + |g_y| \quad (5.12)$$

Sobel 算子不但可以产生较好的检测效果,而且对噪声具有平滑抑制作用。但是,其得到的边缘较粗,且可能出现伪边缘<sup>[2]</sup>。因此,可以将 Sobel 算子检测出的边缘点视作候选边缘点,经过进一步的边缘定位处理后得到更好的边缘图像。

### 5.1.3 Canny 边缘检测算法

Canny 边缘检测算法是 John F. Canny 于 1986 年提出的一个多级边缘检测算法<sup>[3]</sup>。Canny 算法虽然年代久远,但现在的研究中仍广泛使用,而且已成为边缘检测的一种标准算法。Canny 算法基于以下三个基本目标来设计。

(1) 低错误率。所有边缘都应该被找到,并且没有伪响应,即检测到的边缘必须尽可能真实的边缘。

(2) 边缘点应被准确定位。已定位的边缘必须尽可能接近真实边缘。

(3) 单一边缘点响应。对于真实的边缘点,检测器应仅返回一个点。真实边缘周围要有尽可能少的局部最大值点。

Canny 算法的目标是在数学上尽可能实现上述三个准则,并且试图找到满足这些准则的最佳解。通常来说,找到一个满足前面目标的闭式解是很困难的(或不可能的)。Canny 算法使用高斯函数的一阶导数来对边缘进行近似,并用一些特殊处理来尽可能满足上述准则。

Canny 边缘检测算法可以分为下面五个步骤。

第一步,应用高斯滤波来平滑图像。这一步主要作用是去除噪声对后面边缘检测算子的影响。应用高斯模糊去除噪声,可以有效降低伪边缘的识别。高斯模糊的半径选择很重要,过大的半径很容易让一些弱边缘检测不到。图 5.6(a)和(b)分别展示了原图和高斯模糊后的图像。

第二步,计算图像梯度的幅值和方向。具体来说,用 Sobel 边缘检测算子计算图像梯度,并使用式(5.2)和式(5.3)计算梯度的幅值和方向。图 5.7 展示了这一步处理后的梯度幅值图像。

第三步,非极大值抑制。这是一种边缘细化的方法,通常真实边缘附近的几个像素都会产生较大的梯度幅值,正如在 5.1.2 节提到的那样,Sobel 算子检测出的边缘较粗。非极大值抑制的思想是保留梯度方向上的局部最大值,而将非最大值置为零。这里定义水平、垂直、+45°和-45°四种梯度方向,每个方向包含两个角度范围,如图 5.8 所示。若梯度的方向为-22.5°~22.5°或者-157.5°~157.5°,则将此梯度方向称为垂直方向。每



图 5.6 用于边缘检测的原始图像和原图高斯模糊之后的结果



图 5.7 使用 Sobel 算子计算出的梯度幅值图像

个方向对应两个邻居,例如,梯度方向为垂直时,对应的两个邻居为上、下两个像素;梯度方向为 $+45^\circ$ 时,对应的两个邻居为右下和左上两个像素。如图 5.9 所示,梯度方向与边缘的法线方向一致,即与边缘切线方向垂直。例如,梯度方向为垂直的边缘点处于水平边缘上,梯度方向为 $+45^\circ$ 的边缘点处于 $-45^\circ$ 边缘上。

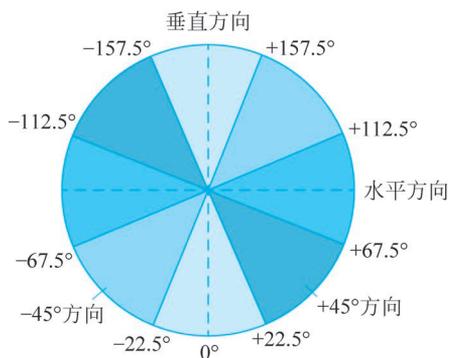


图 5.8 梯度的四个方向和对应的边缘

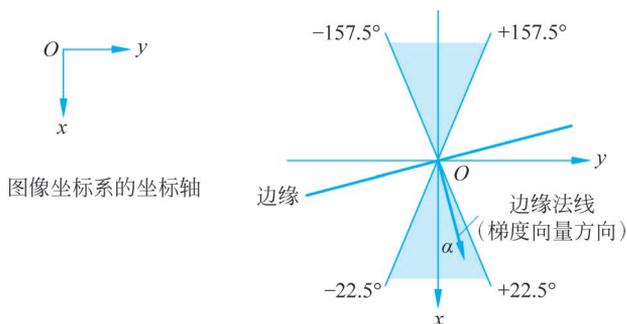


图 5.9 边缘与梯度方向对应关系

下面定义检测出的梯度幅值为  $M(x, y)$ , 梯度方向为  $\alpha(x, y)$ , 经过非极大值抑制处理后的图像为  $g_N(x, y)$ 。对于图像中每一个点, 首先根据  $\alpha(x, y)$  找到梯度对应的方

向,之后找到 $(x,y)$ 在该方向上的两个邻居 $(x_1,y_1)$ 和 $(x_2,y_2)$ ,如垂直方向上的两个邻居为 $(x-1,y)$ 和 $(x+1,y)$ 。如果这个点梯度的幅值小于两个邻居之一的幅值,即 $M(x,y) < M(x_1,y_1)$ 或 $M(x,y) < M(x_2,y_2)$ ,则这个点被抑制,即 $g_N(x,y) = 0$ ;否则, $g_N(x,y) = M(x,y)$ 。经过非极大值抑制后的图像如图 5.10 所示,可以发现边缘相较于梯度幅值图像得到细化。



图 5.10 非极大值抑制后的图像

第四步,双阈值处理。经过非极大抑制后图像中仍然有很多噪声点,称为伪边缘。当使用单阈值去除伪边缘时,若阈值设置过低,则无法将伪边缘完全去除;若阈值设置过高,则会删除实际的有效边缘点。因此,Canny 算法设置了高阈值  $T_H$  和低阈值  $T_L$  进行双阈值处理。 $g_N(x,y) \geq T_H$  的边缘称为强边缘, $T_H > g_N(x,y) \geq T_L$  的边缘称为弱边缘。Canny 算法建议,高阈值和低阈值的比应该为 2:1 或 3:1。图 5.11(a)的二值图像展示了双阈值处理后的强边缘,图 5.11(b)展示的是强边缘+弱边缘。

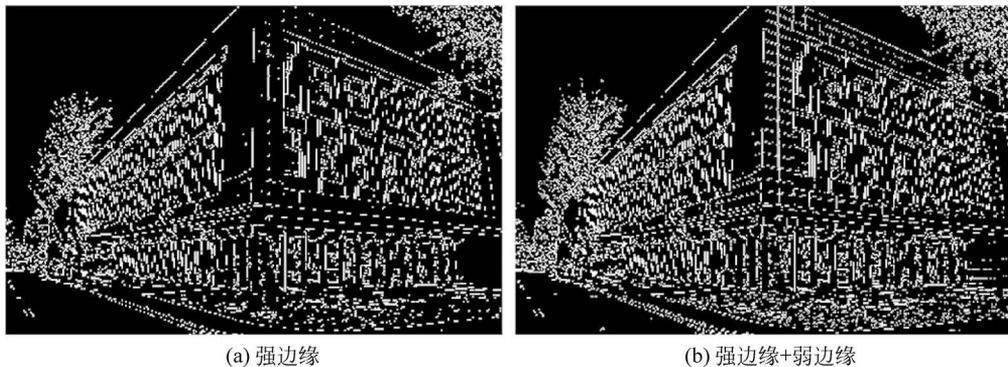


图 5.11 双阈值处理后的二值图像(右侧墙面有更多细节)

第五步,滞后边界跟踪。强边缘可认为是真实的边缘点,而弱边缘需要进一步判断。对每个弱边缘迭代地在其 8 邻域内搜索强边缘。若一个弱边缘直接与强边缘相连或者仅通过弱边缘连接到强边缘,则认为此弱边缘是真实的边缘点;否则,认为其是伪边缘。下面提供一种基于广度优先搜索的实现方式,其步骤如下。

- (1) 准备一个队列,将所有强边缘点放入队列,并标记其为真实边缘。
- (2) 从队列中取出一个边缘点,遍历其 8 邻域中的点,找到其中没有被标记为真实边缘的弱边缘。
- (3) 标记在(2)中满足条件的那些点为真实边缘,并将这些点放入队列。
- (4) 重复步骤(2)和(3)直至队列为空。
- (5) 所有标记为真实边缘的点组成了最终的边缘图像。

图 5.12 中的二值图像展示了图 5.6(a)中的原始图像经过 Canny 边缘检测算法生成的边缘图像。



图 5.12 Canny 边缘检测算法生成的边缘图像

## 5.2 线的检测

除了边缘外,直线在图像中也经常出现,尤其是在有人造物品的场景中,如建筑、道路等,因此检测图像中的直线是一个很有价值的问题。5.2.1 节主要介绍了基于图像卷积的直线检测方法。5.2.2 节和 5.2.3 节分别介绍两个参数化直线检测方法,具体地,5.2.2 节介绍了 Hough 直线检测,5.2.3 节介绍了用随机采样一致(RANSAC)算法检测直线。其中 Hough 直线可以应用于图像中多条直线的检测,用 RANSAC 算法检测直线则对噪声鲁棒性更强,一般只能用来检测一条直线。

### 5.2.1 基本原理

本节首先介绍基于图像卷积的直线检测方式,这是一种简单直接的方式。一些可以选用的模板如图 5.13 所示,这四个模板对单像素宽、特定方向的直线有最大的响应。

-1	-1	-1	2	-1	-1	-1	2
2	2	2	-1	2	-1	-1	2
-1	-1	-1	-1	-1	2	-1	-1
(a) 水平		(b) +45°		(c) 垂直		(d) -45°	

图 5.13 线检测模板,分别对四个对应方向的单像素宽直线有最大响应

上面的四个模板会对暗背景下的亮直线产生很大的正响应,而对于亮背景下的暗直线有很大的负响应。如果只关心前一种,那么只需要设定阈值筛选出响应较大的点;如果只关心后一种,那么可以将模板取反并筛选出具有较大响应的点;如果两种都需要检测,那么可以根据响应的绝对值进行筛选。令  $R_1$ 、 $R_2$ 、 $R_3$  和  $R_4$  分别表示图 5.13 中从左到右的各个模板的响应,假设在图像中的某一点处对于所有的  $j \neq k$  有  $R_k > R_j$ ,则该点最可能位于模板  $k$  所检测的直线上。

虽然这种基于卷积的方法实现简单,但是也存在着一一定的局限性。首先是检测的直线方向比较有限,其次经过阈值处理后的同一条直线往往会形成若干片段,需要进行连接处理。在实际的应用中常用参数化的方法来表示图像中的直线。

正如 5.1 节提到的那样,边缘图像剔除了很多图像中的冗余信息,而保留了重要的结构信息。因此,可以在边缘图像的基础上提取直线的参数化模型,即找到边缘点连成的直线。然而这并不是一个简单的问题,首先能够穿过图像的直线有很多,而对于每一条可能的直线又需要知道哪些边缘点在此直线上。其次,为了准确地检测直线,算法还需要合理地处理噪声的影响。Hough 直线检测算法和 RANSAC 直线检测算法可用在有一定噪声的边缘图像中检测直线,并且用参数化的方式表示检测出的直线。

### 5.2.2 Hough 变换

在边缘图像中拟合直线面临着三个主要问题:第一,给定了属于某条直线的点,那么直线是什么?第二,这些点属于哪条直线?第三,有多少条直线? Hough 变换解决了上面三个问题。

考虑  $xy$  平面上的一个点  $(x_i, y_i)$  和形式为  $y_i = ax_i + b$  的一条直线。通过点  $(x_i, y_i)$  的直线有无数条,且对不同的  $a$  和  $b$  值它们都满足方程  $y_i = ax_i + b$ 。可以将该等式写为  $b = -x_i a + y_i$ ,考虑  $ab$  平面(也称为参数空间),将得到固定点  $(x_i, y_i)$  的单一直线方程。此外,第二个点  $(x_j, y_j)$  在参数空间  $ab$  平面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  $(x_i, y_i)$  相关联的直线相交于一点,令其为  $(a', b')$ ,如图 5.14(b)所示。其中  $a'$  和  $b'$  是同时通过  $xy$  平面中点  $(x_i, y_i)$  和点  $(x_j, y_j)$  的直线的斜率和截距,如图 5.14(a)所示。在  $xy$  平面上考虑,任意在直线  $y = a'x + b'$  上的点,在  $ab$  平面上都相交于  $(a', b')$ 。

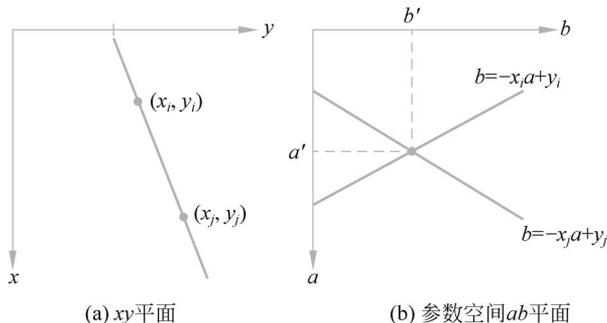


图 5.14

原理上可以对  $xy$  平面中的每一个边缘点  $(x_k, y_k)$  画出其在参数空间对应的直线, 那么  $xy$  空间中的主要直线可以在参数空间中寻找交点来确定。在参数空间上有大量直线通过的点对应在  $xy$  平面有穿过大量边缘点的直线。然而在实际中, 当直线逼近垂直方向时, 直线的斜率  $a$  会趋于无穷大。为了解决这一问题, 可以使用直线的极坐标表示:

$$\rho = x \cos \theta + y \sin \theta \quad (5.13)$$

图 5.15(a)展示了参数  $\rho$  和  $\theta$  的几何解释。

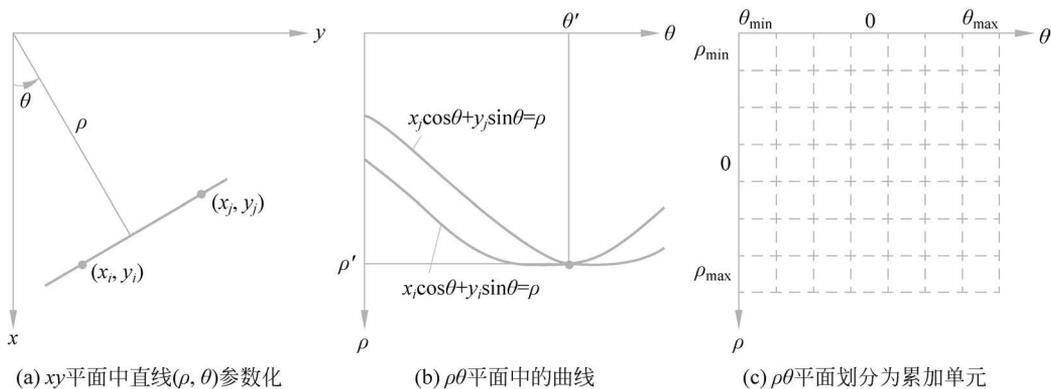


图 5.15

图 5.15(b)中的曲线表示通过  $xy$  平面中一个点  $(x_k, y_k)$  的一组直线, 两个曲线的交点  $(\rho', \theta')$  对应图 5.15(a)中通过点  $(x_i, y_i)$  和点  $(x_j, y_j)$  的直线。如图 5.15(c)所示, 可以将  $\rho\theta$  参数空间划分为累加单元。参数  $\theta$  和  $\rho$  的取值范围分别限定在  $-90^\circ \leq \theta \leq 90^\circ$  和  $-D \leq \rho \leq D$ , 其中  $D$  是图像对角线的距离, 这两个范围可以涵盖所有经过图像的直线。

即便可以将  $\rho\theta$  参数空间划分成了有限的累加空间, 但是遍历累加空间所有的参数模型来检验是否有足够多的边缘点在此直线上仍然需要很多时间。为了提高效率, 可以采用投票的思想进行处理。首先遍历所有的边缘点, 每个边缘点为所有可能的参数模型投票(所有通过该点的直线对应的累加单元加一); 之后遍历每个累加单元的所有参数模型, 找到票数多的几个单元来构造主要直线。在这个过程中虽然噪声点也会投票, 但是这些投票基本没有一致性, 故不会产生主要直线。

Hough 直线检测算法的流程如下。

- (1) 将  $\rho\theta$  参数空间根据所需精度划分为累加单元  $A(i, j)$ , 并将累加单元清空。
- (2) 对于每个边缘点  $(x_k, y_k)$ , 遍历所有的  $\theta$ , 并根据  $\rho = x_k \cos \theta + y_k \sin \theta$  计算出对应的  $(\rho, \theta)$ 。令  $A(\rho, \theta) = A(\rho, \theta) + 1$ 。
- (3) 选择投票大于一定阈值  $T$  的累加单元, 这些累加单元的参数  $\rho$  和  $\theta$  对应图像中的主要直线。
- (4) 在检测到主要直线后, 再检查每条直线通过哪些点, 使用这些点进一步拟合出更精确的直线表示(如使用最小二乘法)。

回忆在 5.1 节介绍的边缘检测, 边缘点往往具有较大的梯度幅值, 此外每个边缘点

还有梯度方向的信息。下面利用方向信息对上述算法步骤(2)进行优化。对每个边缘点遍历投票时,只关注其梯度方向附近一定范围的 $\theta$ ,而非遍历所有 $\theta$ 。这样做一方面可以节省算法的运算时间,另一方面可以减轻噪声点的影响。这一优化在图像中不以散点状出现的直线会更为有效。下面解释这个优化方法的合理性。注意到参数 $\theta$ 代表了直线的法线方向,如图 5.15(a)所示。此方向与梯度方向基本一致,所以用梯度方向来近似参数 $\theta$ 是合理的。此外,Hough 变换的思路也可以用于圆的检测,方法是在 $(a, b, r)$ 空间上划分累积单元并进行投票,其中 $(a, b)$ 代表圆心坐标, $r$ 表示圆的半径。

### 5.2.3 RANSAC 直线检测

RANSAC 算法是一种从含有外点的观测数据集中估计数学模型参数的迭代方法,外点是指偏离正常范围很远无法适应数学模型的数据。因此,RANSAC 算法也可以理解为一种外点剔除算法。该算法需要用到随机采样,因此它是一种非确定的算法。它会有一定概率输出一个合理的解,这个概率会随着迭代次数的增加而逐渐变大。

RANSAC 算法的一个假设是观测数据中既包含可以被数学模型描述的内点,也包含代表着噪声的外点;另一个假设是给定一个少量的内点集合,存在一种可以用它们估计出数学模型参数的方法。

RANSAC 算法执行之前需要确定一系列参数:

- $n$ ——最少需要来估计数学模型参数的点数。根据之前的假设, $n$  不应该太大。
- $k$ ——算法迭代的次数。
- $t$ ——用于判断一个点是否符合模型参数的阈值,即内点。
- $d$ ——最少需要几个内点才认为该模型有效。

RANSAC 算法的流程如下:

- (1) 从所有数据中随机选择  $n$  个点。
- (2) 使用这  $n$  个点估计出数学模型的参数,令估计出的模型为  $M$ 。
- (3) 对于未被采样到的点,根据阈值  $t$  来判断该点是否符合模型  $M$ ,并统计出符合模型的内点数目  $d'$ 。
- (4) 若  $d' \geq d$ ,则认为  $M$  是一个有效模型,重新用  $d' + n$  个内点来估计出更精确的模型  $M'$  并计算出拟合误差  $e'$ 。如果  $d' < d$ ,那么认为  $M$  是无效的,不做进一步处理。
- (5) 执行步骤(1)~(4)直至达到迭代次数  $k$ 。
- (6) 在所有有效模型中选择拟合误差  $e'$  最小的模型  $M'$  作为最终结果。

在了解 RANSAC 算法的基本流程之后,下面讨论如何使用此算法在边缘图像中拟合出一条参数化的直线。首先确定参数  $n=2$ ,因为使用两个点就能确定一条直线。每次迭代随机选择两个点并确定出一条直线方程。在步骤(3),利用点到直线的距离和阈值  $t$  比较来判断其是否为内点。在步骤(4),如果两点确定的直线有足够多的内点,那么通过最小二乘法来拟合直线,并采用平均最小二乘误差作为拟合误差。在步骤(6),选择拟合误差最小的直线作为结果。

注意到 RANSAC 算法中迭代次数  $k$  是一个很重要的参数,它决定了算法的时间,同

时也影响着算法找到合适模型的概率。下面从两个角度来讨论参数  $k$  的确定。

定义  $w$  为所有数据点中内点的比例(只需要合理估计即可),并且定义采样的  $n$  个点都是内点的情况为一次好的采样(算法步骤(1))。那么易得出一次好的采样概率为  $w^n$ 。第一个角度考虑首次出现好的采样时期望采样几次。令  $k$  为出现第一次好的采样时的采样次数,那么可以得到  $k$  的概率分布为

$$P(k=t) = (1-w^n)^{t-1}w^n \quad (t=1,2,\dots) \quad (5.14)$$

注意到  $k$  满足几何分布,根据几何分布的公式  $k$  的期望为

$$E(k) = \frac{1}{w^n} \quad (5.15)$$

此期望的含义理解为希望能够通过  $k$  次采样得到一次好采样。同样根据几何分布的公式  $k$  的标准差为

$$SD(k) = \sqrt{\frac{1-w^n}{w^{2n}}} = \frac{\sqrt{1-w^n}}{w^n} \quad (5.16)$$

下面换一个角度来看待这个问题,如果希望  $k$  次采样后能够有一次好采样的概率大于或等于一个定值  $p$ ,那么可以得到

$$1 - (1-w^n)^k \geq p \quad (5.17)$$

式中:  $(1-w^n)^k$  表示  $k$  次采样全部为坏采样的概率。

式(5.17)两边取对数化简后,令  $k$  为满足上述条件的最小值,得到

$$k = \frac{\log(1-p)}{\log(1-w^n)} \quad (5.18)$$

表 5.1 展示了  $p=0.99$  的情况下,  $n$  和  $w$  在不同取值时  $k$  的根据式(5.18)得到的值。

表 5.1  $p=0.99$  的情况下,  $n, w$  和  $k$  取值的对应关系

	$w=95\%$	$w=90\%$	$w=80\%$	$w=75\%$
$n=2$	$k=2$	$k=3$	$k=5$	$k=6$
$n=3$	$k=3$	$k=4$	$k=7$	$k=9$
$n=4$	$k=3$	$k=5$	$k=9$	$k=13$
$n=5$	$k=4$	$k=6$	$k=12$	$k=17$
$n=6$	$k=4$	$k=7$	$k=16$	$k=24$

RANSAC 算法的优点是估计模型的鲁棒性强,对于有大量噪声的数据,它仍可以给出现较为精准参数估计。但是,该算法的时间没有上限(除非穷举所有采样),如果限制迭代次数,就会一定程度限制其得到更好的模型参数。从直线检测的应用上来讲, RANSAC 直线检测在图像中只有一条主要直线时非常有效。如果有两条或两条以上主要直线,可能会导致其检验失败。而 Hough 直线检测算法有检测多条直线的能力。

### 5.3 Harris 角点检测



直观来讲,角点是指边缘曲线的方向快速变化的位置。由于角点的独特性和在不同视角下有一定的不变性,它们是常用的图像特征匹配手段。角点检测广泛应用于全景图

像拼接、立体视觉、三维重建、相机运动识别等领域。为了使角点在经过变换后的不同图像中都能够被准确定位,好的角点应该具有一系列不变性。不变性基本分为几何不变性和光照不变性。几何不变性是指角点能够应对图像的平移、旋转以及尺度缩放等变换。光照不变性是指角点能够应对图像亮度和对比度的变换。此外,好的角点还应应对仿射、投影的复杂的几何变换以及图像的噪声、模糊具有鲁棒性。为了检测出此种具有不变性的角点,这一部分主要介绍 Harris 角点检测算法,5.3.1 节介绍了 Harris 角点检测的基本思想以及数学公式的推导,5.3.2 节介绍了 Harris 角点检测算法的流程。

### 5.3.1 基本原理

下面介绍由 Harris 和 Stephens 于 1988 年提出的 Harris 角点检测算法<sup>[4]</sup>。该算法的基本方法是 将一个窗口在图像上扫描一遍,就像使用模板对图像滤波一样。滤波的模板用于计算模板中心位置滤波后的灰度值,而窗口用于计算其朝不同方向微小移动后窗口内各像素灰度值的变化。接下来讨论三种场景:第一种,窗口朝不同方向微小移动后各像素灰度值几乎不变,当窗口在平坦或接近平坦的图像区域时会出现这种情况,如图 5.16 中的窗口 A 所示;第二种,窗口朝某一个方向微小移动后灰度变化明显,而朝其垂直方向微小移动后灰度几乎不变,如图 5.16 中的窗口 B 所示;第三种,向不同方向移动后灰度都会发生明显变化,当窗口中包含角点时常出现这种情况,如图 5.16 中的窗口 C 所示。

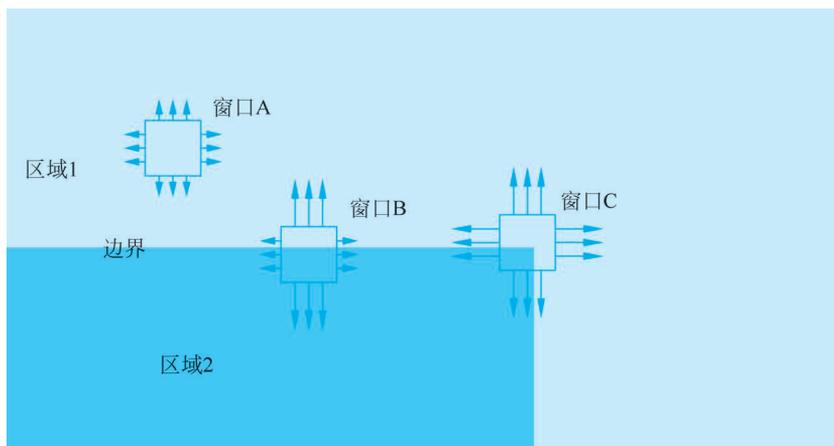


图 5.16 探测窗口出现的三种图像区域 A(平坦)、B(平直边缘)、C(角点)

注:箭头长度表示向该方向移动后窗口灰度值变化大小

设  $f$  表示一张图像,  $f(s, t)$  表示由  $(s, t)$  定义的一个窗口, 其中  $(s, t)$  为窗口的中心点。朝  $(x, y)$  方向移动后相同大小的窗口可以表示为  $f(s+x, t+y)$ 。这两个窗口内的各像素灰度加权平方差可以由下式表示:

$$C(x, y) = \sum_s \sum_t w(s, t) [f(s+x, t+y) - f(s, t)]^2 \quad (5.19)$$

式中:  $w(s, t)$  是一个加权函数。一般来讲该函数是下面两种形式之一: 一是在窗口里

面函数值取 1, 窗口外面函数值取 0; 二是类似于高斯滤波的模板, 表示为指数函数形式, 即

$$w(s, t) = e^{-(s^2+t^2)/2\sigma^2} \quad (5.20)$$

两种加权函数如图 5.17 所示。朝  $(x, y)$  方向微小移动后图像块的灰度值可以用泰勒公式的一阶展开, 近似为

$$f(s+x, t+y) \approx f(s, t) + xf_x(s, t) + yf_y(s, t) \quad (5.21)$$

式中:  $f_x(s, t) = \partial f / \partial x$ ,  $f_y(s, t) = \partial f / \partial y$ , 表示  $(s, t)$  位置处的在  $x$  方向和  $y$  方向偏导数。

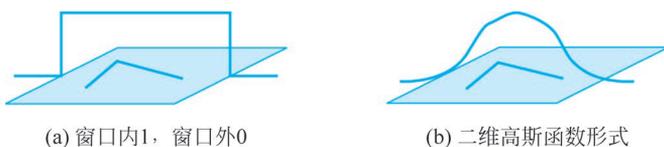


图 5.17 两种加权函数

式(5.19)可以写为

$$C(x, y) = \sum_s \sum_t w(s, t) [xf_x(s, t) + yf_y(s, t)]^2 \quad (5.22)$$

写成矩阵形式

$$C(x, y) = \sum_s \sum_t [x \ y] \left( w(s, t) \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \right) \begin{bmatrix} x \\ y \end{bmatrix} \quad (5.23)$$

其中  $x$  和  $y$  与求和无关, 可以将其提到外面

$$C(x, y) = [x \ y] \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix} \quad (5.24)$$

式中:  $\mathbf{M}$  为 Harris 矩阵, 可表示为

$$\mathbf{M} = \left( \sum_s \sum_t w(s, t) \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \right) \quad (5.25)$$

上面的讨论将窗口灰度值变化用数学方式表达为  $C(x, y)$ , 并写成了矩阵的形式。可以发现,  $\mathbf{M}$  是一个实对称矩阵, 根据实对称矩阵的性质(不同特征值对应的特征向量正交), 它存在两个相互正交的特征向量, 它们的方向为  $\mathbf{v}_1$  和  $\mathbf{v}_2$ 。接下来考虑窗口朝不同方向移动时  $C(x, y)$  的大小, 即  $(x, y)$  为方向不同的单位向量。它们都可以被两个单位长度的特征向量表示

$$\mathbf{v} = a\mathbf{v}_1 + b\mathbf{v}_2 \quad (a^2 + b^2 = 1) \quad (5.26)$$

根据特征向量的性质:

$$\mathbf{v}_1^T \mathbf{M} \mathbf{v}_1 = \mathbf{v}_1^T (\lambda_1 \mathbf{v}_1) = \lambda_1, \quad \mathbf{v}_2^T \mathbf{M} \mathbf{v}_2 = \lambda_2$$

又根据两个向量的正交性:

$$\mathbf{v}_2^T \mathbf{M} \mathbf{v}_1 = \mathbf{v}_2^T (\lambda_1 \mathbf{v}_1) = 0, \quad \mathbf{v}_1^T \mathbf{M} \mathbf{v}_2 = 0$$

那么图像块朝  $\mathbf{v}$  表示的方向移动后灰度变化为

$$C(\boldsymbol{v}) = \boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v} = (a\boldsymbol{v}_1 + b\boldsymbol{v}_2)^T \boldsymbol{M} (a\boldsymbol{v}_1 + b\boldsymbol{v}_2) = a^2 \lambda_1 + b^2 \lambda_2 \quad (5.27)$$

可以发现,此响应在  $\lambda_1$  和  $\lambda_2$  之间,因此两个特征向量可以表示  $C$  函数取值最大和最小的方向。如图 5.18 所示,可以根据两个特征值的大小来区分上面讨论的三种情况:两个特征值都很小,表示无论朝什么方向移动  $C$  都很小,对应平坦区域;一个特征值远大于另一个,表示沿着一个方向移动图像块  $C$  会很大,但是朝其垂直方向移动  $C$  会很小,对应平直边缘的区域;两个特征值都很大,说明无论朝哪个方向移动图像块  $C$  都比较大,对应的就是角点。

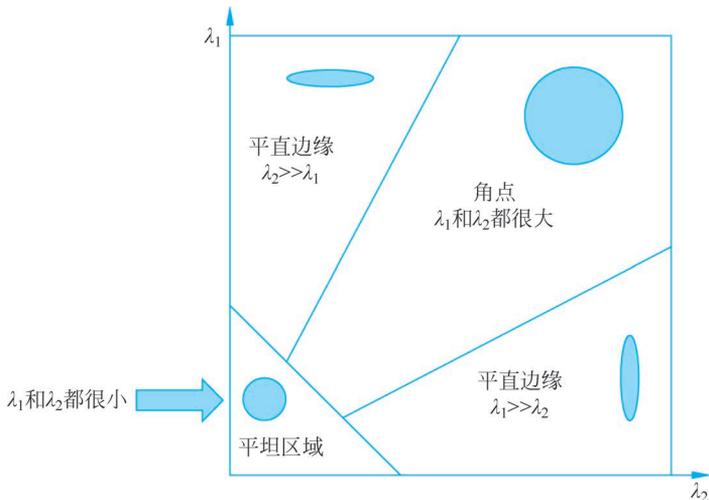


图 5.18 两个特征值大小与图像区域的对应

上面提到可以用两个特征值来区分三种场景,出于计算效率的考虑,Harris 角点检测使用了矩阵  $\boldsymbol{M}$  的行列式代替了特征值的乘积,使用矩阵  $\boldsymbol{M}$  的迹代替了行列式的和,这两个等式在数学上是严格成立的。因此,定义如下公式来计算 Harris 角点响应<sup>[4]</sup>:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(\boldsymbol{M}) - k \operatorname{tr}^2(\boldsymbol{M}) \quad (5.28)$$

式中:  $k$  是一个根据经验确定的常数,它的值取决于具体的实现,一般为  $0.04 \sim 0.06$ 。

响应  $R$  在两个特征值都很大的情况下是一个大的正数,表示出现了角点;响应  $R$  在一个特征值远大于另一个的情况下是一个绝对值很大的负数,表示平直边缘;响应  $R$  在两个特征值都很小的情况下是一个绝对值很小的数,表示了平坦的区域。 $k$  可以解释为一个敏感因子, $k$  越小,Harris 角点检测越容易发现角点。一般设定一个阈值  $T$ ,把图像窗口响应  $R > T$  且为局部最大值的窗口中心点称为角点。

### 5.3.2 计算流程

前面阐述了 Harris 角点检测的基本思想以及数学的表达,本节介绍算法的具体实现。Harris 角点检测算法基本步骤如下:

(1) 计算图像的梯度用于近似公式中的  $f_x$  和  $f_y$ ,梯度的计算可以选择 Sobel 算子,计算图像梯度前一般需要平滑图像以去除噪声。

(2) 令步骤(1)计算的两个方向的梯度为  $I_x$  和  $I_y$  (是两个矩阵), 计算三个梯度的乘积矩阵,  $I_x^2 = I_x \odot I_x$ ,  $I_x I_y = I_x \odot I_y$ ,  $I_y^2 = I_y \odot I_y$ 。

(3) 使用加权函数模板分别对  $I_x^2$ ,  $I_x I_y$ ,  $I_y^2$  进行加权, 模板大小为窗口大小, 图 5.17(a) 所示加权函数对应模板值全为 1, 图 5.17(b) 所示加权函数对应二维高斯函数模板。

(4) 令加权后生成  $M$  矩阵的三个元素为  $A = w \otimes I_x^2$ ,  $B = w \otimes I_x I_y$ ,  $C = w \otimes I_y^2$ ,  $w$  为加权函数。根据式(5.28)计算 Harris 的角点响应  $R = AC - B^2 - k(A + C)^2$ 。

(5) 在邻域内进行非极大值抑制, 选取响应高于阈值  $T$  的局部最大值作为角点。

上述过程中,  $\odot$  代表矩阵对应元素乘运算, 即对于矩阵  $A$  和  $B$ ,  $(A \odot B)_{ij} = A_{ij} B_{ij}$ ;  $\otimes$  代表卷积运算。

### 5.3.3 多尺度 Harris 角点

本节首先讨论 Harris 角点检测的不变性。首先讨论几何不变性, 图像平移后窗口可以在平移后的位置检测到角点, 图像旋转后 Harris 矩阵的特征向量方向改变但特征值不变, 因此不影响角点的检测。由此可见, Harris 角点对于平移和旋转具有不变性。下面讨论光照不变性。由于 Harris 角点采用微分运算, 对于图像灰度(强度)的增加和下降不敏感, 对于对比度的变换则不会改变 Harris 响应极值的位置, 但是可能会影响角点检测的数量。对于图像尺度的缩放, Harris 角点检测不具有不变性。图 5.19(a) 展示了当尺度较小时可以检测到的角点, 图 5.19(b) 展示了同样的角点在经过放大后每个探测窗口检测到的都是平直边缘。

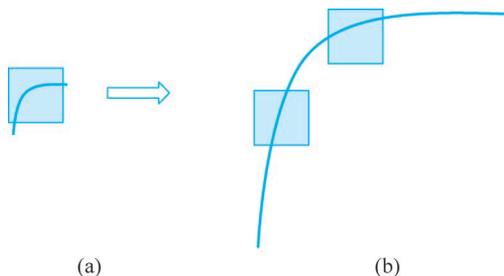


图 5.19 尺度放大后导致的 Harris 角点丢失

为了解决上述问题, 下面介绍尺度不变的 Harris 角点检测<sup>[5]</sup>。类似于 Harris 矩阵, 提出尺度自适应二阶矩阵, 定义如下:

$$\mu(\mathbf{x}, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) \otimes \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix} \quad (5.29)$$

式中:  $\mathbf{x}$  为窗口中的所有像素;  $\sigma_I$  为积分尺度;  $\sigma_D$  为微分尺度;  $L_x$  为  $x$  方向的一阶导数,  $L_y$  为  $y$  方向的一阶导数, 求导之前图像使用标准差为  $\sigma_D$  的高斯函数平滑。积分尺度  $\sigma_I$  则控制了二维高斯函数形式的窗口函数  $g$  的标准差。

有了上述尺度自适应 Harris 矩阵, 就可以在多个尺度上提取 Harris 角点, 构造多尺度图像的方法为尺度空间<sup>[6]</sup>。尺度空间基本思想是用一组平滑参数生成一组逐渐平滑、

细节也逐渐消失的图像。这一组图像逐渐模糊的过程模拟了图像在尺度缩小时细节的丢失,而图像的尺寸保持不变。因此,尺度空间用一致的方式来处理不同尺度下的图像结构。

多尺度 Harris 角点生成的具体过程:首先用一组  $K$  个平滑参数来构造尺度空间。其中第  $n$  个平滑参数  $\sigma_n = \xi^n \sigma_0$ ,其中  $n$  取  $0, 1, \dots, K-1$ ,  $\xi$  和  $\sigma_0$  是用于生成尺度空间的常数。这组平滑参数将作为高斯函数的标准差用于后续步骤生成平滑图像。接下来使用尺度自适应二阶矩矩阵在尺度空间中提取多尺度的 Harris 角点。具体来说,对于第  $n$  个尺度,计算该尺度下的 Harris 矩阵  $\mu(x, \sigma_1, \sigma_D)$ ,其中积分尺度  $\sigma_1 = \sigma_n$ ,微分尺度  $\sigma_D = \sigma_n$  用于平滑图像(文献[5]中取  $s=0.7$ )。使用式(5.28)计算 Harris 响应,对于每个尺度选取 8 邻域的极值点且大于一定阈值的位置作为候选角点。

上述步骤选择了每一尺度上图像空间域的极值,下面在尺度域上进一步筛选候选角点,采用的方法为高斯拉普拉斯(Laplacian of Gaussian, LoG)算子。高斯拉普拉斯算子是对高斯函数求二阶导数,再与图像进行卷积。其等价于先用高斯函数对图像进行平滑,再使用拉普拉斯算子对平滑图像求二阶导数。尺度为  $\sigma$  的高斯平滑函数为

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

对其求二阶导数得到尺度为  $\sigma$  高斯拉普拉斯函数,即

$$\text{LoG}_\sigma(x, y) = \frac{\partial^2 G_\sigma(x, y)}{\partial x^2} + \frac{\partial^2 G_\sigma(x, y)}{\partial y^2} = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-(x^2+y^2)/(2\sigma^2)} \quad (5.30)$$

用不同尺度参数的归一化 LoG 算子绝对值与图像卷积来在尺度域上寻找极值。对于第  $n$  个尺度,计算出  $L_n = |\sigma_n^2 \text{LoG}_{\sigma_n}| \otimes I$ ,其中  $I$  表示原图,  $L_n$  是原图与归一化 LoG 算子绝对值做卷积得到的 LoG 响应,LoG 的归一化即乘以  $\sigma_n^2$  的系数。对于上一步得到的每一个候选点,  $(x, y, n)$  表示在候选点位于尺度  $n$  的  $(x, y)$  位置。若其在尺度域上的 LoG 响应不为极值,即  $L_n(x, y) \leq L_{n+1}(x, y)$  或  $L_n(x, y) \leq L_{n-1}(x, y)$ ,则将其删除。若 LoG 响应  $L_n(x, y)$  小于一定阈值,则也将此候选点删除。经过在尺度域上的筛选剩下的点即为多尺度 Harris 角点,这些角点不仅拥有在图像空间域的位置信息,而且增加了一维在尺度域上的信息,因此具有尺度不变性。

## 5.4 特征描述子



角点检测可以在图像中找到包含较多信息的关键点,为了在不同的图像中匹配角点,还需要描述角点周围的图像特征。这一部分将介绍几种用于描述角点周围特征的描述子,描述子之间相似性的比较可以用于匹配角点。这些描述子也需要对于尺度缩放、旋转和光照变化等具有不变性。5.4.1 节介绍 SIFT 描述子,5.4.2 节介绍 MOPS 描述子,5.4.3 节介绍 HOG 描述子。

### 5.4.1 SIFT 描述子

SIFT 尺度不变特征转换是由 Lowe 等提出的在图像中提取不变性特征的算法<sup>[7]</sup>。

之所以称为转换,是因为其将原图像转换成相对于局部图像特征具有尺度不变性的坐标,并从中提取若干具有不变性的 SIFT 关键点和特征描述子。其对图像尺度变化和旋转具有不变性,对一系列仿射畸变、视角变化、噪声以及光照变化具有鲁棒性。同时, SIFT 也是一种启发式的算法,其中有大量需要通过实验确定的参数。

SIFT 算法的第一步是找到对尺度变化有不变性的图像位置,即关键点。这一步是通过构造尺度空间,并在所有可能的尺度上搜索稳定的特征来实现的。在尺度空间上寻找到的关键点可以表示为  $(x, y, \sigma)$ , 其中  $x$  和  $y$  表示关键点在图像平面中的位置,  $\sigma$  表示其在尺度域上的尺度。

与 5.3.3 节中构造尺度空间的方法类似, SIFT 用一组平滑参数确定尺度空间,并将其作为高斯模板的标准差生成一组平滑图像。SIFT 将尺度空间的一组平滑图像称为音阶组。参考了乐理中的音阶, 这组图像在尺度域上的范围为  $\sigma \sim 2\sigma$ 。SIFT 首先用  $s+1$  个图像将此范围划分成  $s$  个间隔, 这  $s+1$  个图像进行高斯平滑的标准差(平滑参数)为  $\sigma, k\sigma, k^2\sigma, \dots, k^s\sigma$ , 其中  $k$  为固定常数, 且  $k^s\sigma = 2\sigma$ 。尺度空间将在后续过程中用于计算高斯差分, 为了能够覆盖整个尺度空间, 需要额外计算两个图像, 分别使用标准差为  $k^{s+1}\sigma$  和  $k^{s+2}\sigma$  的高斯模板进行平滑, 因而每个组共  $s+3$  个图像。与 5.3.3 节多层 Harris 角点不同, SIFT 通过在图像平面域中下采样并在尺度域采用不同的范围来构造尺度空间金字塔, 如图 5.20 所示。图中展示了三层尺度空间金字塔, 每一层都是一组图像组成的尺度空间, 从下层到上层每组图像尺寸由于下采样而缩小。图 5.20 中每个组将尺度域划分  $s=2$  个间隔, 且每组由  $s+3=5$  个逐渐平滑的图像组成。一个组的划分相当于在  $(x, y, \sigma)$  尺度空间中连续变化的尺度  $\sigma$  轴上采样出  $s+3$  个图像平面  $(x, y)$ , 可以理解成组是尺度空间的离散化表示。

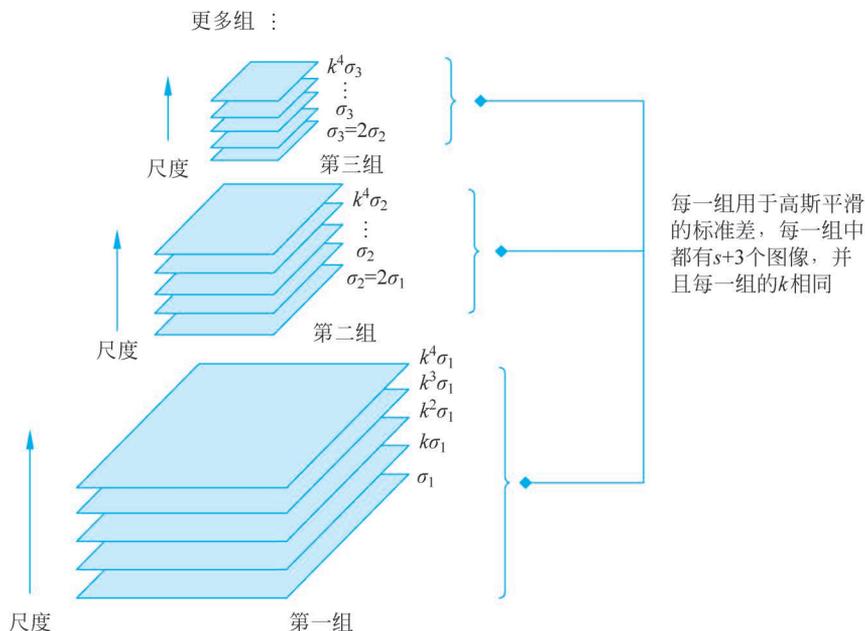


图 5.20 尺度空间构成的三层图像金字塔 ( $s=2$ )

下面介绍尺度空间金字塔的构造方法。根据划分尺度空间的方法,可以用原图像和  $\sigma_1$  来生成第一个组,其中第一组的尺度域范围为  $\sigma_1 \sim 2\sigma_1$ 。用  $\sigma_1$  和常数  $k$  确定的  $s+3$  个平滑参数作为高斯平滑的标准差生成第一组图像。第二个组的第一个图像是由原图像下采样(行和列分别采样到原来的一半),再用第一组所用标准差的 2 倍,即  $\sigma_2=2\sigma_1$  进行平滑。第二组用与第一组相同的  $k$  进行平滑参数的划分,并生成第二组接下来的图像。生成尺度空间图像金字塔的流程如下。

- (1) 确定  $\sigma_1$ ,由原图  $I_1$  和标准差  $\sigma_1$  生成第一组。
- (2) 由上一组初始图像  $I_{i-1}$  下采样生成当前组初始图像  $I_i$ ,确定当前组的标准差  $\sigma_i=2\sigma_{i-1}$ 。
- (3) 使用标准差  $\sigma_i, k\sigma_i, k^2\sigma_i, \dots, k^{s+2}\sigma_i$  平滑  $I_i$  生成当前组的多尺度图像。
- (4) 重复步骤(2)和(3)直至达到金字塔指定层数。

从图 5.21 中可以观察到,图像在同一组中从下到上以及在金字塔中从下到上变得越来越模糊,这也丢失了更多细节,但是在大体外观上可以保持相似的结构。尺度空间金字塔中的每一层对应了一个组,而每组表示了一个尺度空间。

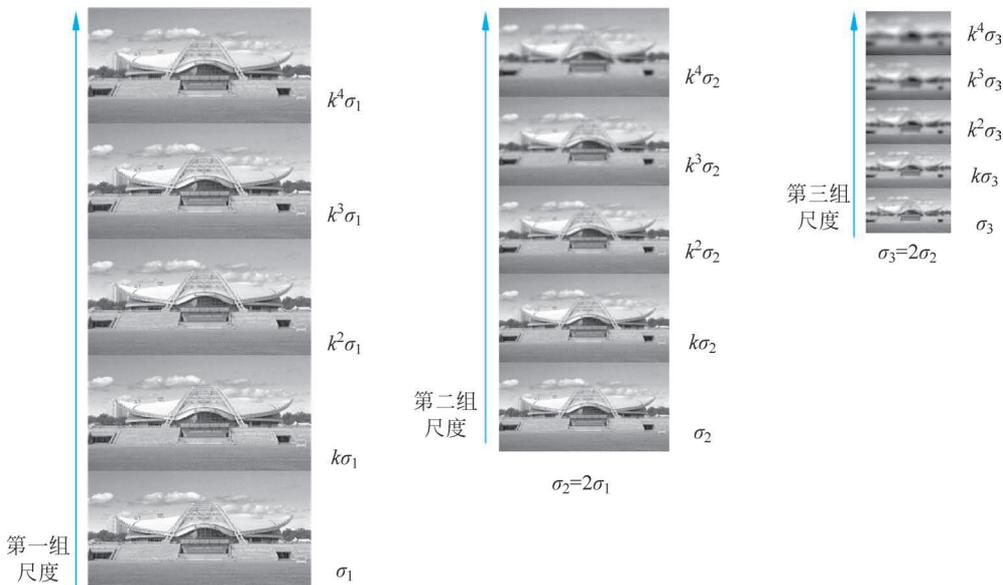


图 5.21 尺度空间构成的图像金字塔的实例

第二步是定位局部极值点作为关键点,先用尺度空间的平滑图像得到高斯差分(Difference of Gaussians, DoG)图像,再通过两个步骤细化关键点位置并验证其合理性。SIFT 中高斯差分其实就是同一组的相邻两个平滑图像做差,前面提到的每组中有  $s+3$  个图像,做差后得到  $s+2$  个高斯差分图像,如图 5.22 所示。事实上,高斯差分是高斯拉普拉斯算子近似,不仅如此,高斯差分还将尺度信息考虑在内,两者关系为

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G \tag{5.31}$$

式中:  $G$  为高斯函数,式子左边表示的就是高斯差分;  $k\sigma$  和  $\sigma$  分别为同组中相邻的上下

两层的标准差,  $\nabla^2$  表示拉普拉斯算子;  $\sigma^2 \nabla^2 G$  为归一化高斯拉普拉斯算子;  $k-1$  在金字塔的每组的每个图像上都是一个常数, 不会影响极值的判断。

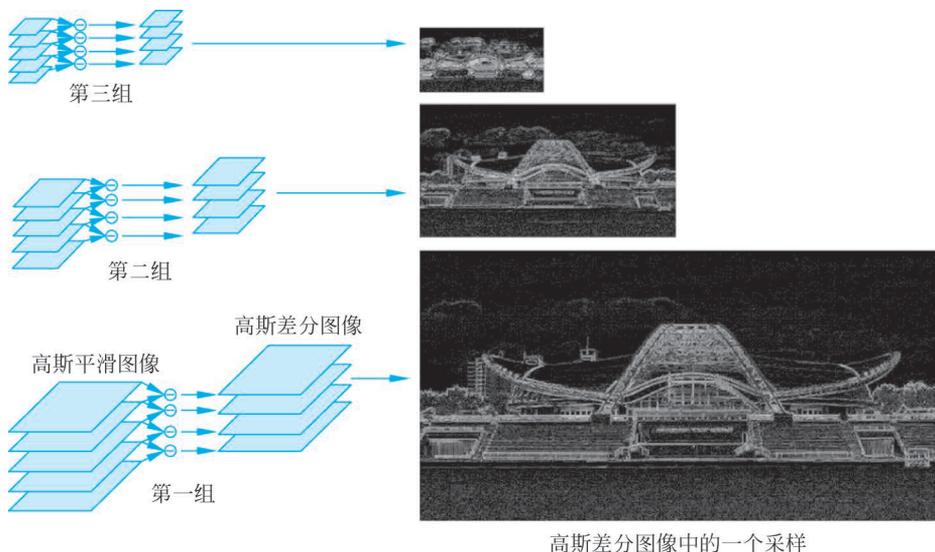


图 5.22 尺度空间  $s+3$  个高斯平滑图像生成的  $s+2$  个高斯差分图像

令  $L$  表示高斯平滑后的图像, 例如第一组的第一张图像可以表示为

$$L_1(x, y, \sigma_1) = G(x, y, \sigma_1) \otimes f(x, y) \quad (5.32)$$

类似地, 第一组的第二张图像为  $L_1(x, y, k\sigma_1)$ 。令  $D$  表示高斯差分图像, 根据上面的讨论第一组的第一张差分图像为

$$D_1(x, y, \sigma_1) = L_1(x, y, k\sigma_1) - L_1(x, y, \sigma_1) \quad (5.33)$$

即

$$D_1(x, y, \sigma_1) = [G(x, y, k\sigma_1) - G(x, y, \sigma_1)] \otimes f(x, y) \quad (5.34)$$

至此在金字塔中每个尺度空间上都构造了两组图像, 一组是高斯平滑图像  $L$ , 另一组是高斯差分图像  $D$ 。

下面介绍 SIFT 在每一组高斯差分图像中寻找极值的方法, 如图 5.23 所示。对于第  $i$  组中一个高斯差分图像  $D_i$  中的每个位置 (图 5.23 中深色位置), 此位置的  $D_i(x, y, \sigma)$  与其同一图像的 8 个邻居比较, 再与其上下两个同组图像的 9 个邻居比较。若  $(x, y, \sigma)$  位置的值大于或小于所有上述邻居, 则称为极值。每一组的最下层和最上层两个图像不会提取极值。因此, 每一组  $s+2$  张高斯差分图像中, 会在中间  $s$  个图像中寻找极值点。

经过上面的步骤, 金字塔中的每一个组都能得到极值点, 这些极值点的坐标定义在这一组所

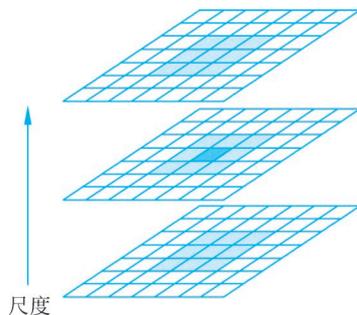


图 5.23 高斯差分图像中深色位置像素与灰色邻居比较寻找极值

代表的尺度空间上,并以离散的形式表示。例如,某组中的一个极值点坐标为 $(x, y, \sigma)$ , $\sigma$ 表示极值点位于该组的第几个图像上, $(x, y)$ 表示在图像中的像素坐标,其范围是该组图像的尺寸。但是,前面在离散空间的极值点并不是真正的极值点,需要细化关键点来提高关键点精确度。采用的方法是子像素插值,利用已知的离散空间点插值得到连续空间的极值点。高斯差分 $D(x, y, \sigma)$ 可视为三维尺度空间中的连续三元函数,SIFT使用泰勒展开式的一次和二次项来进行插值,用向量的形式表示为

$$\begin{aligned} D(x) &= D + \left(\frac{\partial D}{\partial \mathbf{x}}\right)^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial}{\partial \mathbf{x}} \left(\frac{\partial D}{\partial \mathbf{x}}\right) \mathbf{x} \\ &= D + (\nabla D)^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} \end{aligned} \quad (5.35)$$

式中: $D$ 及其导数都在采样点处计算; $\mathbf{x} = (x, y, \sigma)^T$ 表示从采样点处的偏移, $\nabla$ 为梯度运算符,且有

$$\nabla D = \frac{\partial D}{\partial \mathbf{x}} = \begin{bmatrix} \partial D / \partial x \\ \partial D / \partial y \\ \partial D / \partial \sigma \end{bmatrix} \quad (5.36)$$

$\mathbf{H}$ 为海森矩阵,且有

$$\mathbf{H} = \begin{bmatrix} \partial^2 D / \partial x^2 & \partial^2 D / \partial x \partial y & \partial^2 D / \partial x \partial \sigma \\ \partial^2 D / \partial y \partial x & \partial^2 D / \partial y^2 & \partial^2 D / \partial y \partial \sigma \\ \partial^2 D / \partial \sigma \partial x & \partial^2 D / \partial \sigma \partial y & \partial^2 D / \partial \sigma^2 \end{bmatrix} \quad (5.37)$$

式(5.35)对偏移量 $\mathbf{x}$ 求导,并令其为零,得到极值时的偏移量

$$\hat{\mathbf{x}} = -\mathbf{H}^{-1} (\nabla D) \quad (5.38)$$

式中: $\hat{\mathbf{x}}$ 代表相对插值中心的偏移量,当它在任一维度( $x$ 或 $y$ 或 $\sigma$ )上的偏移量大于0.5时,意味着插值中心已经偏移到了它的邻近点上,所以必须改变当前关键点的位置。同时新的位置上反复插值直到收敛;也有可能超出所设定的迭代次数或者超出图像边界的范围,此时这样的点应该删除。Lowe<sup>[7]</sup>提出做5次迭代,最终将偏移 $\hat{\mathbf{x}}$ 加到收敛后的位置上,可以达到亚像素精度。为了得到极值点上 $D$ 的值,将式(5.38)代入式(5.35)得到

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} (\nabla D)^T \hat{\mathbf{x}} \quad (5.39)$$

Lowe<sup>[7]</sup>的实验结果表示,当 $D(\hat{\mathbf{x}})$ 的绝对值小于0.03时(图像灰度值范围为 $[0, 1]$ )拒绝该极值。细化后的关键点可以表示为 $\mathbf{X} + \hat{\mathbf{x}}$ ,其中 $\mathbf{X}$ 为收敛到的离散坐标, $\hat{\mathbf{x}}$ 表示细化到亚像素精度的偏移。

高斯差分的极值点可能位于平直边缘,但是SIFT算法的关键点对角点更感兴趣,这些位于平直边缘的极值点需要被消除。下面用局部曲率量化表示平直边缘和角点的区别。平直边缘可以刻画为在一个方向上有很高的曲率,而在其垂直方向有较低曲率。在关键点 $\mathbf{X}$ 所在高斯差分图像 $D$ 的对应位置计算海森矩阵来得到该点的曲率:

$$\mathbf{H} = \begin{bmatrix} \partial^2 D / \partial x^2 & \partial^2 D / \partial x \partial y \\ \partial^2 D / \partial y \partial x & \partial^2 D / \partial y^2 \end{bmatrix} \quad (5.40)$$

其中  $\mathbf{H}$  的两个特征值与两个方向的曲率成正比。令海森矩阵大小两个特征值为  $\alpha$  和  $\beta$ ，用矩阵的行列式和矩阵的迹代替两个特征值的积与和，并且令  $r$  为  $\alpha$  和  $\beta$  两个特征值的比，可得

$$\frac{\text{tr}^2(\mathbf{H})}{\det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (5.41)$$

式(5.41)只取决于两个特征值的比例，当两个特征值相等时， $(r + 1)^2/r$  取得最小值，且随着  $r$  增大而增大。对  $r$  设置一个阈值  $T_r$  并使用

$$\frac{\text{tr}^2(\mathbf{H})}{\det(\mathbf{H})} < \frac{(T_r + 1)^2}{T_r} \quad (5.42)$$

来验证  $\mathbf{H}$  对应的特征值比例(两个方向曲率的比例)是否小于阈值  $T_r$ 。

第三步为每一个关键点根据局部图像性质分配一个方向，这样可以在此方向来计算描述子，实现描述子对图像旋转的不变性。在平滑图像  $L$  上可以用下式计算  $(x, y)$  位置梯度的幅值和方向：

$$M(x, y) = \sqrt{[L(x + 1, y) - L(x - 1, y)]^2 + [L(x, y + 1) - L(x, y - 1)]^2} \quad (5.43)$$

$$\theta(x, y) = \arctan[(L(x, y + 1) - L(x, y - 1)) / (L(x + 1, y) - L(x - 1, y))] \quad (5.44)$$

每个关键点  $\mathbf{X}$  根据其所在组和尺度确定平滑图像和邻域大小，在邻域中采样来形成方向直方图，并用 36 个柱的直方图来覆盖图像平面的  $360^\circ$  角度的范围。每一个采样根据其相对于关键点的距离，将梯度幅值经过高斯函数加权后累加到对应方向的直方图柱上。

直方图的最高峰值对应的是局部梯度的主方向。除了最高的峰值以外，对其他高于 80% 最高峰值的局部峰值(高于相邻两个柱)也会创建对应方向的关键点。这些关键点在尺度空间的位置(图像平面的位置及其尺度)都是相同的。一般来讲，只有约 15% 的关键点会被 SIFT 算法分裂出多个不同方向相同位置的关键点，这可以明显地提高关键点匹配的稳定性。最后，为了进一步提高角度定位的精准度，用峰值及其相邻的两个直方图值进行抛物线插值，设插值的抛物线方程为  $h(t) = at^2 + bt + c$ ，其中  $a$ 、 $b$  和  $c$  为抛物线的系数， $t$  为自变量， $t \in [-1, 1]$ ，表示相对于峰值的偏移。因为已知  $h(-1)$ 、 $h(0)$  和  $h(1)$  的值分别对应峰值左边、峰值和峰值右边三个直方图的值，将它们代入方程解得三个参数：

$$\begin{cases} a = \frac{h(1) + h(-1)}{2} - h(0) \\ b = \frac{h(1) - h(-1)}{2} \\ c = h(0) \end{cases} \quad (5.45)$$

通过抛物线插值得到更精确的峰值位置，得到其对应的偏移为

$$\hat{t} = \frac{-b}{2a} = \frac{h(-1) - h(1)}{2[h(-1) + h(1) - 2h(0)]} \quad (5.46)$$

前面的步骤确定了极值点在尺度空间中的位置来实现了尺度不变性,并且通过细化关键点和筛除平直边缘点,得到了亚像素精度的关键点。此外,通过直方图统计出关键点的方向。最后这些位于尺度空间金字塔不同组的关键点,需要乘以对应下采样倍率,将图像平面的位置统一到第一组图像的坐标上。

第四步是在关键点周围区域计算描述子,描述子不仅要有区分性,而且要有尺度、方向和光照的不变性。可以用这些描述子确定在多个图像中局部区域的匹配。

图 5.24 总结了 SIFT 生成每个特征点描述子的方法。以关键点为中心取出周围  $16 \times 16$  的像素区域,并根据式(5.43)和式(5.44)计算出区域中每个位置的梯度幅值和方向,在左上角图中显示为随机的箭头。为了实现描述子在方向上的不变性,描述子的坐标及梯度方向都相对于关键点的方向进行旋转。之后用高斯加权函数对每个点的梯度幅值进行加权,在图中显示为虚线圆圈。可以将此加权函数想象成一个钟形的曲面,中心权值高,边上权值低。其目的是减少某一位置微小的变化引起的整个描述子的突然改变。

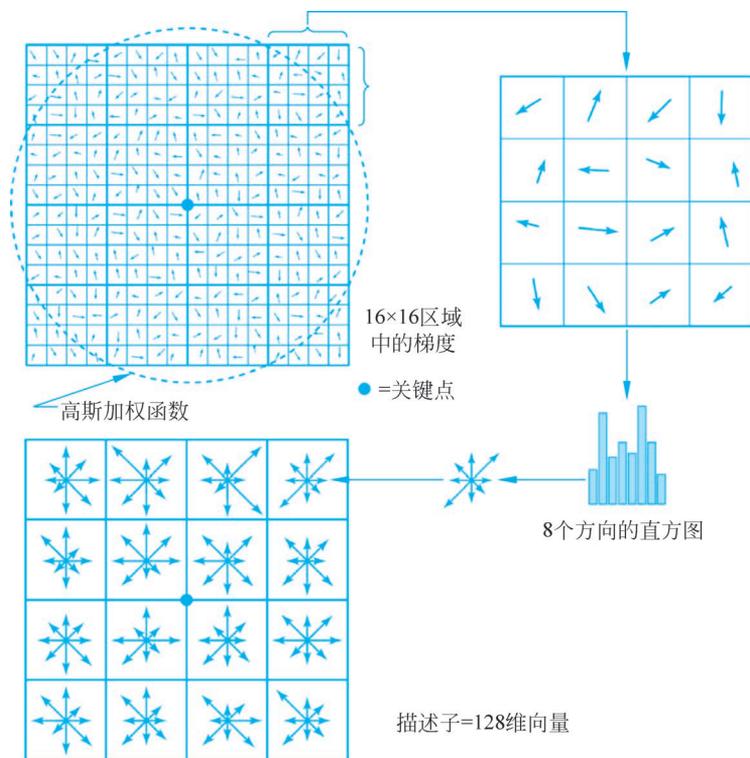


图 5.24 提取关键点描述子的方法

每个关键点会计算出周围  $16 \times 16$  个梯度方向及高斯函数加权后的幅值。将这些梯度划分成  $4 \times 4$  个子区域,每个区域包含 16 个梯度,图 5.24 右上角的图展示了其中一个子区域。并将梯度方向划分为 8 个方向,每个方向相差  $45^\circ$ ,按照  $4 \times 4$  个子区域和 8 个方向来统计直方图,如图 5.24 左下角的图所示。在构造  $4 \times 4 \times 8$  的直方图过程中,对于

$16 \times 16$  区域内的每个像素, SIFT 并没有将加权后的梯度幅值直接加到子区域和角度距离其最近的直方图上, 而是使用了三线性插值将其按比例分配到相邻的直方图上。图 5.25 进一步解释了三线性插值的原理。在这一步要构建  $4 \times 4 \times 8$  的直方图, 如图 5.25(a) 所示直方图的  $x$  轴和  $y$  轴表示了图像平面, 按照子区域划分出  $4 \times 4$  个直方图。如图 5.25(b) 所示  $z$  轴表示方向角, 按照  $45^\circ$  的间隔划分出 8 个直方图, 且该轴第 7 个直方图与第 0 个直方图相邻。图 5.25 所示的区域内一点  $P$ , 其所在子区域坐标为  $(1, 2)$ , 其梯度方向最靠近  $270^\circ$  (在角度轴上最靠近第 6 个直方图)。若不采用三线性插值, 则其加权梯度幅值  $M$  会直接加到离其最近的  $(1, 2, 6)$  位置的直方图上。SIFT 所用的三线性插值方法则将加权梯度幅值  $M$  按比例分配到在  $x$ 、 $y$ 、 $z$  三个轴上相邻的 8 个直方图中, 如图 5.25(c) 的立方体所示, 立方体 8 顶点表示在位置和角度上距离点  $P$  最近的 8 个直方图。具体分配方式: 加权幅值  $M$  会在三个轴上乘  $1-d$  后加到相应的直方图中, 其中  $d$  是像素位置或方向距离直方图中心的距离, 此距离是在直方图坐标下的, 并且只对相邻的直方图有贡献, 因此  $d \in [0, 1]$ 。以计算点  $P$  对  $(1, 1, 5)$  位置的直方图的贡献为例, 点  $P$  在  $x$  轴和  $y$  轴与子区域  $(1, 1)$  中心点的距离分别为  $d_{x0}$  和  $d_{y0}$ , 在角度上与第 5 个直方图 (代表  $225^\circ$ ) 的距离为  $d_{z0}$ 。 $d_{x0}$ 、 $d_{y0}$  和  $d_{z0}$  三个距离都应转化为  $0 \sim 1$  之间的直方图坐标。以  $d_{z0}$  为例, 设  $P$  的角度为  $250^\circ$ , 那么

$$d_{z0} = \frac{250 - 225}{45} = \frac{5}{9}$$

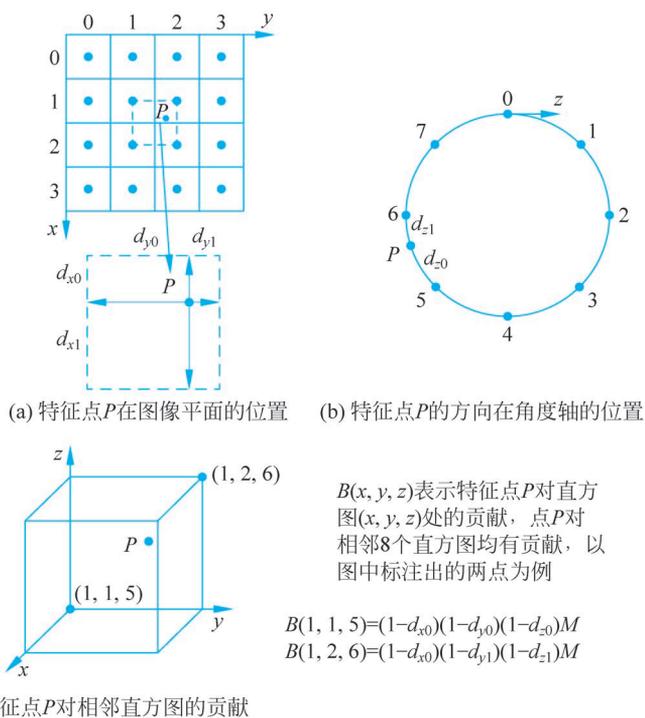


图 5.25 三线性插值分配直方图

转换成直方图坐标之后,在三个轴上乘以  $1-d$  进行分配,那么点  $P$  对  $(1,1,5)$  位置的直方图的贡献为

$$B(1,1,5) = (1 - d_{x_0})(1 - d_{y_0})(1 - d_{z_0})M$$

对区域内  $16 \times 16$  个像素都按照上述方法进行三线性插值分配,最终统计出  $4 \times 4 \times 8$  的直方图。

上面构造了  $4 \times 4 \times 8$  的直方图,并由此构造出 128 维向量作为描述子。为了减弱光照的影响,SIFT 还对描述子进行两步归一化:第一步是将此向量归一化到单位长度,这一步使描述子对图像对比度变化具有不变性。相机饱和度等原因仍然可能产生非线性的光照变化,这些影响可能会导致一些梯度相对幅值的巨大变化,但是不太会影响其方向。第二步设定了一个 0.2 的阈值,将归一化后的向量大于阈值的每一维都设为 0.2,之后再归一化到单位长度。

### 5.4.2 MOPS 描述子

MOPS 描述子是由 Brown 等在 2005 年提出的具有不变性的特征<sup>[8]</sup>,其相对于 SIFT 特征描述子实现更加简单。对于图像尺度变化不大的任务,如图像拼接,可以使用 MOPS 特征描述子方便计算。

MOPS 算法的第一步是提取关键点,关键点的提取采取的是多层 Harris 角点检测。与 5.3.3 节所述有所不同,MOPS 使用高斯金字塔提取多尺度角点而非使用尺度空间。首先对于输入图像  $I(x, y)$  构建了高斯图像金字塔  $P_l(x, y)$ ,采用的金字塔图像平滑的标准差  $\sigma_p = 1.0$  和下采样率  $s = 2$ ,在每一层金字塔都提取 Harris 角点,下式高度凝练了 5.3 节中讨论的 Harris 矩阵构造的过程:

$$\mathbf{H}_l(x, y) = g_{\sigma_i}(x, y) \nabla_{\sigma_d} P_l(x, y) \nabla_{\sigma_d} P_l(x, y)^T \quad (5.47)$$

式中:  $\mathbf{H}_l(x, y)$  为金字塔  $l$  层中  $(x, y)$  位置对应的 Harris 矩阵;  $\nabla_{\sigma_d}$  为用  $\sigma_d = 1.0$  的高斯平滑后再提取梯度的算子;  $\nabla_{\sigma_d} P_l(x, y)$  为该算子在金字塔  $l$  层图像  $P_l(x, y)$  中提取的图像二维梯度向量;  $g_{\sigma_i}(x, y)$  为 Harris 角点检测中提到的窗口高斯加权函数,其中  $\sigma_i = 1.5$ 。

与 5.3 节中不同的是,MOPS 中计算的 Harris 角点的响应为

$$R_l(x, y) = \frac{\det(\mathbf{H}_l(x, y))}{\text{tr}(\mathbf{H}_l(x, y))} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \quad (5.48)$$

在  $3 \times 3$  邻域内寻找角点响应  $R_l$  的局部极大值并且高于阈值  $T$  的点作为角点。角点精度会被进一步细化到亚像素精度,即使用  $3 \times 3$  邻域的响应值拟合一个二维的二次函数并找到最大值点。

对于每一个关键点,要计算关键点的方向,使其对方向变化具有鲁棒性。关键点的方向定义为其梯度方向,但与式(5.47)不同的是,此梯度定义为  $\nabla_{\sigma_o} P_l(x, y)$ ,其中  $\sigma_o = 4.5$ ,也就是说提取梯度时高斯平滑采用了更大的标准差,这样对关键点方向的估计更加具有鲁棒性。

第二步是计算描述子,其需要实现在图像特征之间的可靠高效匹配。MOPS 在关键点分布在高斯图像金字塔不同的层次上,层次越高,该层图像采样率越低。MOPS 描述子在关键点周围采样图像,所用的采样率低于关键点所在金字塔层次的采样率,这样提取的特征对于关键点位置更不敏感。对于一个有方向的特征点 $(x, y, l, \theta)$ ,其中 $l$ 为其所在金字塔层次, $\theta$ 为关键点方向。在亚像素级的关键点位置周围采样 $8 \times 8$ 像素的图像块,采样点之间采取5像素的间隔,如图5.26所示。为了保证描述子具有方向不变性,采样时需要根据关键点的方向进行旋转。采样之后,此图像块需要归一化使得平均值为0、标准差为1,这使得特征对灰度值的仿射变换(亮度变化和对比度变化)具有不变性。最后,对 $8 \times 8$ 的图像块使用 Haar 小波变换,生成一个64维的包含小波系数的描述子向量。

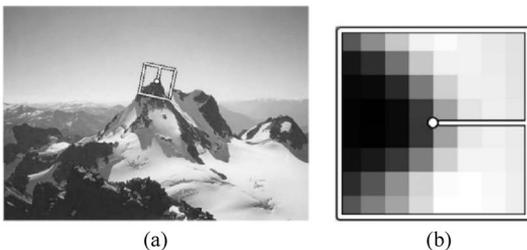


图 5.26 MOPS 采样的  $8 \times 8$  像素的图像块<sup>[8]</sup>

### 5.4.3 HOG 描述子

HOG 描述子用梯度方向统计直方图来表示图像的物体特征,并用于该类物体检测。梯度直方图的思想最早由 McConnell 在 1986 年提出,用于模式识别<sup>[9]</sup>,直至 2005 年 Navneet Dalal 和 Bill Triggs 通过 HOG 描述子与支持向量机相结合在静态图像中识别行人<sup>[10]</sup>,才开始得到广泛应用。下面先介绍 HOG 描述子是如何构造检测行人描述子的。

假设有一个行人的图像块,计算 HOG 描述子的第一步是直接计算图像梯度,其最常用的模板是一维中心对称的模板,如图 5.27 所示,可以将其理解为 5.1.2 节中介绍的 Prewitt 算子的一维情况。Dalal 和 Triggs 也测试了 Sobel 算子等更为复杂的模板来计算两个方向的梯度,但是这些模板在检测行人的任务中表现不佳。通常来讲,计算梯度前需要对图像进行平滑,但是他们通过实验发现省略求图像梯度前的高斯平滑步骤甚至会使得 HOG 描述子在行人检测中表现更好。



图 5.27 HOG 使用的梯度检测算子

第二步是通过梯度的幅值和方向统计直方图。在第一步中求出了 $x$ 方向和 $y$ 方向的梯度,可以根据式(5.2)和式(5.3)计算出梯度的幅值和方向。之后将图像分割成 $8 \times 8$ 的网格,如图 5.28 所示,计算每个网格中的梯度直方图。梯度的方向可以分为有符号梯度和无符号梯度两种:有符号梯度方向定义为 $0^\circ \sim 360^\circ$ ;无符号梯度方向定义为 $0^\circ \sim 180^\circ$ ,一个梯度的方向以及它旋转 $180^\circ$ 之后的方向被认为是一样的。Dalal 和 Triggs 发现,在行人检测任务中使用无符号梯度效果更佳,因此将 $0^\circ \sim 180^\circ$ 的范围划分为 9 份构造直方图,直方图中的每一个柱表示 $20^\circ$ 的范围。网格中的每个点都将其梯度的幅值加到其方向对应的直方图的柱中。

第二步是通过梯度的幅值和方向统计直方图。在第一步中求出了 $x$ 方向和 $y$ 方向的梯度,可以根据式(5.2)和式(5.3)计算出梯度的幅值和方向。之后将图像分割成 $8 \times 8$ 的网格,如图 5.28 所示,计算每个网格中的梯度直方图。梯度的方向

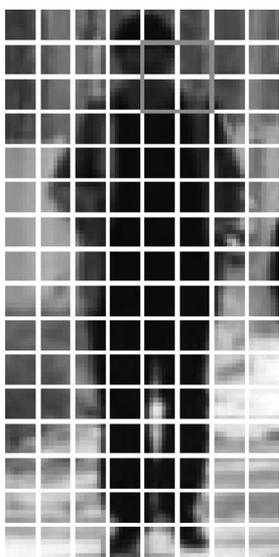


图 5.28 HOG 中  $8 \times 8$  网格的划分  
(深色框代表归一化使用的  $16 \times 16$  块)

第三步是归一化。HOG 将每 4 个网格组合成  $16 \times 16$  的块再进行归一化,如图 5.28 中的深色框所示。每个深色框范围中包含 4 个网格,也就是说有 4 个 9 柱的直方图,将它们组成 36 维向量。以图 5.28 为例,图像共可以提取出 105 (15 行  $\times$  7 列) 个  $16 \times 16$  的块,每个块都有一个归一化的 36 维向量,将这些向量拼接可以得到  $36 \times 105 = 3780$  维的向量作为描述子即可描述一个行人的图像块。

如果用 HOG 描述关键点附近的特征,可以采取类似的方法。假设在图像中已经检测出了 Harris 角点,之后在角点周围选取一个  $16 \times 16$  的块,将其按照上述方式划分成 4 个  $8 \times 8$  的网格。分别对每个网格计算直方图,并在  $16 \times 16$  的块中将 4 个直方图归一化生成 36 维向量作为描述此关键点的特征描述子。HOG 描述子计算简单,对于光照和平移变换具有不变性,但是无法应对物体的旋转。由于使用网格在空间上粗略采样并采用了归一化处理,HOG 能够忽略行人身体移动造成的影响,检测出直立状态下的行人。

## 5.5

### 特征匹配



前面几节介绍的 SIFT 算法、MOPS 算法等已经检测出了一系列特征点,并且对于每个特征点能够知道其在图像中的位置,以及描述特征点局部图像信息的描述子,这些描述子一般被表示成向量的形式。希望在图像之间找到特征点的匹配,使得匹配的特征点对的描述子尽可能相似。在 5.5.1 节中介绍了如何高效地求解特征点的匹配。在 5.5.2 节中介绍了通过两张图像中的若干对匹配点求解单应性矩阵来描述图像间像素的变换,从而实现图像对齐。

#### 5.5.1 匹配计算

假设有一个待匹配的特征点,需要在其他图像的特征点集合中找到描述子最相近的特征点作为匹配,其本质是一个最近邻搜索问题。

$d$  维空间  $R^d$  中的一个向量称为样本点,由有限样本点组成的集合为样本集。例如,一个 SIFT 描述子可以定义为 128 维空间中的一个样本,而从一个图像中提取的全部特征点的描述子可以构成一个样本集。最近邻搜索问题定义为给定一个样本集  $E$  和一个样本集之外的样本点  $q$ ,寻找  $q$  在  $E$  中的最邻近  $p$ ,使得任意样本集中的  $p'$ ,都满足  $p$  与  $q$  的距离不大于  $p'$  与  $q$  的距离。此距离其实是特征点描述子的匹配度,一般用欧几里得距离来衡量。下面介绍几种常用的解决最邻近问题的方法。

首先是最简单直接的穷举法。穷举所有样本集  $E$  中的样本  $p'$ ,并计算其与  $q$  的距

离,找到距离最小的样本  $p$  即为最邻近。穷举法的优点是算法简单,并且能够确定地寻找到最邻近。假设样本集  $E$  中一共有  $n$  个样本,穷举法查询一次的时间复杂度为  $O(nd)$ ,即样本数量与维度的乘积,当样本集中样本数量过多时,会导致穷举法效率下降。

下面介绍使用局部敏感哈希(也称为位置敏感哈希,LSH)的方法。LSH的基本思想:先对所有样本进行位置敏感哈希,这种哈希在两个样本距离越相近时,它们之间发生哈希冲突的概率越大。在预处理阶段,计算样本集  $E$  中每个样本的哈希值  $\text{key}(p)$ ,将哈希值相同的样本放到一起称为桶,桶可以根据哈希值来索引。在查询时先对待查询的样本  $q$  做同样的哈希处理,并根据得到的哈希值索引对应的桶。之后计算样本  $q$  与这个桶中所有样本的距离,并找出桶内与样本  $q$  的最邻近样本。用桶内的计算代替全局的计算,这样能够在保证成功概率的同时提高速度。相比于穷举法,使用 LSH 的最邻近算法效率更高,但是有概率无法找到最邻近的样本。

由上面的讨论可以看出,使用 LSH 查询最邻近样本的效率和质量的好坏主要取决于哈希函数。下面介绍一种基于向量夹角余弦的 LSH,用两个向量的夹角来衡量两个向量是否相似,向量夹角越小,越相似。两个向量夹角的余弦定义为

$$\cos(x_1, x_2) = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}\| \|\mathbf{y}\|} \quad (5.49)$$

任意两个向量夹角为  $0^\circ \sim 180^\circ$ ,因此  $\cos(x_1, x_2) \in [-1, 1]$ 。首先在  $d$  维空间  $R^d$  中随机选取  $m$  个单位向量  $\mathbf{w}_1 \cdots \mathbf{w}_m$ 。对于样本集  $E$  中每个样本  $x$ ,计算其与上述  $m$  个单位向量的夹角余弦。根据余弦值的符号来构造哈希函数的哈希值(或称为键):

$$\text{key}(x) = (\text{key}_1(x), \cdots, \text{key}_m(x))^T \quad (5.50)$$

可以认为是一个  $m$  维向量,且每一维只有  $+1$  和  $-1$  两个取值,其中

$$\text{key}_i(x) = \begin{cases} +1, & \cos(x, w_i) \geq 0 \\ -1, & \cos(x, w_i) < 0 \end{cases} \quad (5.51)$$

这样最多有  $2^m$  种键,每种键对应着一个桶。将样本集中的样本根据键放入对应的桶中,同一个桶中样本点的键相同。如果想查询样本  $q$  的最邻近,就要先计算出  $\text{key}(q)$ ,再与  $\text{key}(q)$  所对应的桶中的所有样本计算距离选取最邻近。这种算法预处理的复杂度为  $O(nmd)$ ,查询一次的复杂度为  $O(md + n_i d)$ ,其中  $n$  是样本集  $E$  的大小, $i$  是查询样本  $q$  的键, $n_i$  是键为  $i$  的桶内的样本数量。当查询大量样本时,利用局部敏感哈希能够大大提高效率。

下面介绍一种基于 KD 树的最邻近搜索算法。KD 树是 Jon Louis Bentley 提出的空间划分的方法<sup>[11]</sup>,主要用于多维空间的数据搜索,最邻近搜索正是其应用之一。

首先介绍 KD 树的构建过程。对于  $K$  维空间的样本集,构造一个二叉树的结构,二叉树上的每个非叶子节点都代表了对  $K$  维空间的一个超平面划分。构建 KD 树的流程图如图 5.29 所示,递归地展开 KD 树,即划分样本点。对于每一次展开,首先选择样本点方差最大的维度  $k_i \in \{1, 2, \cdots, K\}$  进行超平面的划分,对维度  $k_i$  计算所有样本点在此维

度下的中位数  $k_v$  并用其分割样本点,第  $k_i$  维小于  $k_v$  的样本点将其划分到左子树,否则将其划分到右子树。之后对于左右子树分别递归地向下划分,最后构造出 KD 树。构造 KD 树时,除了可以根据样本点在此维度下的方差的大小顺序来选择,也可以直接按照第 1 维、第 2 维……第  $K$  维的顺序来选择。

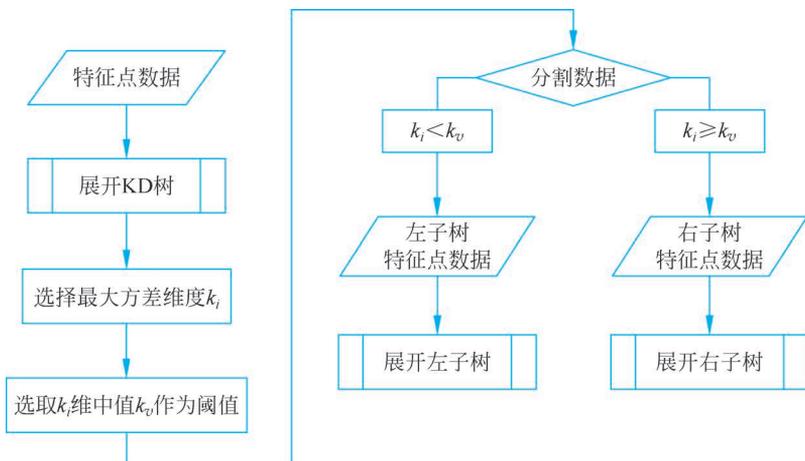


图 5.29 KD 树构建流程图

KD 树构造完成后,接下来需要用查询算法来找到距离样本最近的点,所需要的数据结构为一个记录样本点的栈。在构建好的 KD 树上查询样本点  $q$  的最邻近的算法如下。

(1) 从 KD 树的根节点开始,根据划分深度优先搜索到最底层的划分点  $p_0$  ( $p_0$  是 KD 树中的叶子节点),并将搜索路径上的点放到栈中。

(2) 初始设置最底层的划分点  $p_0$  作为最邻近点,并初始化最短距离为  $p_0$  与  $q$  的距离。

(3) 通过栈回溯,对比当前栈顶点  $p_{top}$  和  $q$  的距离与当前的最短距离,若当前栈顶点距离  $q$  更近,则更新当前栈顶点为最邻近点,最短距离更新为  $p_{top}$  和  $q$  的距离。

(4) 比较样本点到栈顶划分的超平面的距离和当前最短距离,如果当前最短距离更大,那么说明样本点以最短距离画出的超球面与栈顶划分的超平面相交。这种情况下需要进入划分另一边的子树进行搜索,按照步骤(1)中的深度优先搜索方法进入最底层并回溯。否则,不进入另一边的子树搜索,栈顶出栈,回溯至上一层。

(5) 如果回溯到根节点,即根节点出栈之后,那么结束搜索得到最邻近点及其距离。

### 5.5.2 图像对齐

本节介绍如何使用基于图像特征点的方法来实现图像对齐。前面已经介绍了如何在图像中寻找特征点,并且能够通过特征点的描述子寻找最邻近的特征点。有了这些知识,假设给定两个图像,能够在这两个图像中分别提取特征点,并找到一组特征点的匹配。通过对齐这些匹配的特征,可以计算一种映射使得一个图像能够变化为另一个,即图像的对齐。

图像对齐算法的核心是计算一个简单的  $3 \times 3$  矩阵,称为单应性矩阵。单应性矩阵描述了两个平面之间的映射关系,如果两个图像中的特征点都落在同一个平面上,那么这两张图之间的关系可以用单应性矩阵描述。

单应性矩阵是如下形式的  $3 \times 3$  矩阵:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \quad (5.52)$$

令  $(u_1, v_1)$  和  $(u_2, v_2)$  表示一对在第一个和第二个图像中匹配的两个特征点的像素位置,用齐次坐标表示它们通过单应性矩阵的变换关系:

$$s \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \quad (5.53)$$

单应性矩阵的求解在 3.2.3 节有详细的讲解,最少可以通过 4 对匹配求解单应性矩阵。

得到单应性矩阵  $\mathbf{H}$  之后,对第一个图像中的任意一个像素可以通过式(5.53)求解在第二个图像中的对应像素。实际中可以在图像中找到很多对匹配的特征点,因此得出的方程是一个超定方程组,一种选择是求解最小二乘解来求解单应性矩阵。但是,考虑到会出现误匹配的情况,可以使用 5.2.3 节中介绍的 RANSAC 算法来剔除误匹配的外点,得到更为精准的单应性矩阵。

## 本章小结

本章主要介绍了几种图像的特征,包括边缘、线和角点,并介绍了它们的检测方法。5.1 节主要介绍了图像边缘的概念,对边缘建模并发现梯度计算在检测边缘的重要作用。此外,该节介绍了几种计算梯度的边缘检测算子以及 Canny 边缘检测算法。5.2 节介绍了图像中直线的这一重要特征,首先介绍了相对简单的模板卷积方法检测图像中的特定直线,为检测更多种类直线该节介绍了参数化的方法,包括用 Hough 变换来实现直线检测和 RANSAC 算法实现直线检测。RANSAC 算法能够有效地应对噪声的影响,但是只适用于一条直线的检测;Hough 直线检测则可以检测多条直线。5.3 节中寻找的是图像中的角点特征,介绍了 Harris 角点检测算法,通过使用数学公式表达探测窗口在移动时的灰度值变化,计算出 Harris 角点响应大的点为角点。使用多尺度 Harris 角点检测算法,增加了 Harris 角点的尺度不变性。5.4 节介绍了三种特征描述子,分别为 SIFT 描述子、MOPS 描述子和 HOG 描述子。这些描述子可以不同程度地应对尺度、旋转和光照的变化。SIFT 描述子对尺度、旋转和光照具有很强的不变性,但是计算复杂度较高。MOPS 描述子能够较为简便地计算,适用于尺度变化不大的图像匹配问题。HOG 描述子的计算最为简单,但是无法适应物体的旋转。5.5 节介绍了如何使用这些有描述子的特征点在图像之间进行高效率的匹配。最后介绍了根据匹配的特征点通过估计单应性矩阵来对齐图像。

## 习题

1. 参考图 5.2 画出屋顶边缘模型的灰度剖面及其一阶导数和二阶导数的函数图像。
2. 使用 OpenCV 中提供的 Canny 边缘检测函数来编程实现边缘检测,并理解其中参数的含义。
3. 参考图 5.13 构造出检测宽度为 2 像素的水平直线的模板和宽度为 2 像素的垂直直线的模板。
4. 设计在边缘图像中检测圆的 RANSAC 算法,并写出伪代码。
5. 使用 OpenCV 中提供的 Harris 角点检测函数来编程实现角点检测,并理解其中参数的含义。
6. 在 SIFT 描述子的计算中,参考图 5.24 中左上角  $16 \times 16$  的图像块。在图像块坐标下位置为 (4,4) 的像素(左上角  $4 \times 4$  区域中的右下角像素),假设其梯度的方向为  $0^\circ$ ,幅值为 128。试问此像素会对直方图中的正方体分别贡献多少权值?例如,此像素会对第一行第一列角度为  $0^\circ \sim 45^\circ$  的正方体贡献大小为 25 的权值。
7. 考虑二维空间中的一个样本集  $E = \{(2,3), (5,4), (9,6), (4,7), (8,1), (7,2)\}$ ,按照  $x$  轴和  $y$  轴循环划分的顺序构建 KD 树,并在 KD 树上分别搜索样本 (2.1,3.1) 和样本 (2,4.5) 在  $E$  中的最邻近样本点。

## 参考文献

