

## 视频特征

视频指的是通过电信号方式捕捉、记录、处理、存储、传输和重现一系列静态图像的技术。视频技术起初是为电视系统所开发的,但如今已经发展成各种格式,以便消费者记录和保存视频内容。网络技术的发展进一步推动了视频的存储和传输,使得视频片段能够以流媒体的形式存在于互联网上,供用户接收和播放。在互联网全面普及的现代社会,视频信号已经成为人类最重要的信息载体之一。

回溯视频的发展历程,从 1888 年世界第一部电影《朗德海花园场景》诞生至今,其整体发展只有一百多年。然而,随着 1080P、HDR 高清视频的出现以及现今的 8K 视频、网络直播、虚拟现实(VR)等新技术的发展,我们的生活已经被视频技术深入地渗透。这些新技术不仅为我们提供了更加沉浸式的体验,也对视频的语义重构和信息媒介运营产生了深远影响。

随着人工智能(AI)的介入,整个视频内容生态得到前所未有的变革。从内容的创作、生产、分发到播放、营销乃至变现,视频生产流程逐渐实现了自动化和智能化。针对诸如视频分类、字幕生成、视频问答等多种应用任务,获取精准的视频特征成为关键。目前,主要视频特征提取方法包括:手工特征提取、基于神经网络的特征提取和基于预训练模型的特征提取方法。值得注意的是,视频本身是一个多模态信息的结合体,不仅包含图像和声音,还有深度、动作等其他模态信息。本章将对这三种特征提取方式进行详细探讨,并简要介绍其他可以表征视频内行为的模态信息。

### 5.1 手工特征提取

手工特征提取主要通过预先定义的公式对视频进行编码,得到特征信息。以下是一些代表性的方法。

- 时空兴趣点:指的是在连续视频帧中时间和空间维度都发生显著变化的像素点,它们可以通过 Harris-3D 角点检测法进行检测。
- 改进的密集轨迹特征:它是在 Dense Trajectories(DT)算法的基础上,通过增加相机估计以及改进正则化和编码方式实现的。

- 三维尺度不变特征变换：它是将 SIFT 算法从二维空间扩展到三维空间，主要用于三维空间中兴趣点的检测和描述。
- 三维梯度方向直方图：它也是将 HOG 算法从二维空间扩展到三维空间，用于在三维空间中的兴趣点检测和描述。
- 立方体(Cuboid)：是一种基于 Gabor 滤波器的时空兴趣点检测方法，专用于描述视频片段的行为特征，进而用于视频中的行为识别。

本节将围绕这几种特征提取方式展开介绍，帮助读者深入理解手工特征提取在视频分析中的应用。

### 5.1.1 时空兴趣点

时空兴趣点(Space-time Interest Points, STIPs)是指视频中同时在时间维度和空间维度上出现显著变化的像素点。以行走的人为例：在连续的视频帧中，人的腿部因为步行而发生的变化是最显著的，这与其他身体部位的较小变化形成对比。通过提取这些显著变化的像素点，我们可以更好地分析并识别不同的活动和行为。

关于时空兴趣点的检测，存在多种方法，包括 Hessian 特征点检测算法、Harris-3D 角点检测算法以及 Dollar 兴趣点检测算法。尤其值得一提的是，Harris-3D 角点检测方法在不同的光照和尺度变化情况下均表现出稳定的性能。故接下来，我们将以 Harris-3D 角点检测方法为例，进一步探讨如何有效地检测 STIPs。为了更直观地展示，图 5.1 给出了使用 Harris-3D 方法在行走者的腿部运动上检测到的时空兴趣点。



图 5.1 行走者腿部运动时空兴趣点检测结果

Harris-3D 角点检测的核心思想是识别时空维度中的角点，即特征值在局部范围达到最大的像素点。Harris-3D 角点检测计算的主要步骤如下。

#### 1) 构建线性标度空间表示

(1) 映射函数。首先，我们需要一个映射函数  $f: \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ ，以将视频图像序列映射到一维空间。常见的映射方式为像素值提取函数，即  $f(x, y, t) = I(x, y, t)$ 。其中， $I(x, y, t)$  是在时刻  $t$  和空间位置  $(x, y)$  的像素值。

(2) 时空高斯核函数。此函数根据空间方差  $\sigma_l^2$  和时间方差  $\tau_l^2$  计算像素值的权重，计算过程见式(5-1)。

$$g(x, y, t; \sigma_l^2, \tau_l^2) = \frac{\exp\left(-\frac{x^2 + y^2}{2\sigma_l^2} - \frac{t^2}{2\tau_l^2}\right)}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}} \quad (5-1)$$

(3) 线性标度空间表示。将视频图像序列与上述高斯核函数在时空域进行卷积,得到线性标度空间表示  $L(\cdot; \sigma_l^2, \tau_l^2)$ , 计算过程见式(5-2)。

$$L(\cdot; \sigma_l^2, \tau_l^2) = g(\cdot; \sigma_l^2, \tau_l^2) * f(\cdot) \quad (5-2)$$

为了更好地理解此过程,可以考虑一个具体的例子:设点  $p(x, y, t)$  在时空域内,我们首先使用映射函数  $f(\cdot)$  将其映射到一维空间;然后,使用高斯核函数  $g(\cdot)$  计算此点的权重;最后,通过权重值与像素值结合,得到其线性标度空间表示。

### 2) 构建线性标度空间表示计算时空二阶矩阵 $\mu$

利用线性标度空间  $L$  的偏导数来构造二阶海森(Hessian)矩阵,计算过程见式(5-3)。

$$\mu = g(\cdot; \sigma_l^2, \tau_l^2) * \begin{bmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{bmatrix} \quad (5-3)$$

其中,  $L_x, L_y, L_t$  分别表示  $L$  的  $x, y$  和  $t$  方向的偏导数。 $\sigma_l^2$  和  $\tau_l^2$  分别为在空间和时间域的高斯平滑核的平滑尺度。

### 3) 构建 Harris-3D 响应函数 $H$

响应函数用于确定时空域中哪些点可能是兴趣点,其定义见式(5-4)。

$$H = \det(\mu) - k \text{trace}^3(\mu) = \lambda_1 \lambda_2 \lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3 \quad (5-4)$$

其中,  $\lambda_1, \lambda_2, \lambda_3$  是矩阵  $\mu$  的三个特征值,  $k$  为经验系数。当一个点在所有三个时空域方向上都有较大的响应时,我们将其视为一个时空兴趣点。

## 5.1.2 改进的密集轨迹特征

在 5.1.1 节中,我们已经深入探讨了时空兴趣点如何在指定的空间和时间位置对视频信息进行编码。本节将重点放在另一种技术上:改进的密集轨迹特征(improved Dense Trajectories, iDT)。这项技术着重于跟踪图像随时间变化的特征。考虑到 iDT 的起点是密集轨迹特征(Dense Trajectories, DT),我们首先为大家概述一下 DT 算法的基本原理。

### 1. 基本原理

图 5.2 为我们提供了 DT 算法的简明概览,其核心步骤包括:在各个时空尺度上进行密集采样;运用光流法对特定空间尺度的特征点进行跟踪,从而得知特征点的运动路径;基于这些轨迹提取关键特征。具体实施步骤如下。

#### 1) 密集采样

鉴于不同尺度下的图像揭示了不同的图像特点,DT 算法将图像分为多个尺度。在每一尺度,采用固定的步长  $W$  来密集采样图像。在 DT 算法中,图像通常被划分为 8 个尺度,每个尺度的采样步长  $W$  设为 5 像素。但考虑到密集采样可能带来许多冗余数据,DT 采用了与文献“Good Features to Track”中类似的方法:计

算每个密集采样点的自相关矩阵(即 Hessian 矩阵)。如果该矩阵的特征值低于预定阈值,这意味着该采样点周围的图像结构不明显,因此这类点会被排除。

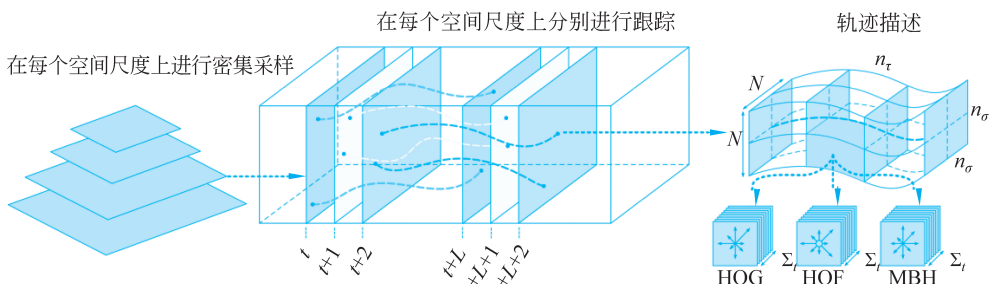


图 5.2 DT 算法提取动作特征流程图

## 2) 特征点跟踪

处理视频图像时,特征点跟踪在多个空间尺度上都非常重要。但对于那些背景较为单调的区域,我们可以跳过跟踪步骤。对于给定的一帧图像  $I_t$ ,首先选择跟踪的特征点,然后采用光流法跟踪特征点得到光流向量,将这些特征点得到的光流向量整合在一起,就可以得到光流场  $\omega_t$ ,光流场由水平分量  $u_t$  和垂直分量  $v_t$  组成。这里,光流实际上是描述画面中像素或特征点运动的向量场。设在时刻  $t$  有一个特征点  $\mathbf{P}_t$ ,其坐标为  $(x_t, y_t)$ 。那么,在  $t+1$  时刻,这个特征点的坐标可以通过式(5-5)进行更新。

$$\mathbf{P}_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega_t) \big|_{(x_t, y_t)} \quad (5-5)$$

其中,  $M$  代表中值滤波器,  $*$  代表矩阵的卷积操作。中值滤波器作用于光流场,有助于减少噪声和不连续性,使特征点的运动方向更加稳定。当我们在连续帧中跟踪特征点时,这些特征点的位置就可以连接起来,形成一个轨迹,例如  $(\mathbf{P}_t, \mathbf{P}_{t+1}, \mathbf{P}_{t+2}, \dots)$ 。

**注意:** 由于多种原因(例如,摄像机的微小移动或物体的快速移动),特征点的跟踪可能会逐渐偏离真实轨迹,我们称之为“轨迹漂移”。为了缓解这个问题,我们限制轨迹的长度为  $L$  帧。也就是说,当我们跟踪了  $L$  帧之后,我们将重新进行密集特征点采样,并重新启动跟踪过程。通常,  $L$  的值被设定为 15。

## 3) 基于轨迹的特征提取

当我们跟踪视频中的特征点时,得到的轨迹形状实际上蕴含了丰富的局部运动信息。这种局部运动可以为我们提供关于动作或行为的关键线索。对于一个长度为  $L$  的轨迹,我们可以通过其位移序列  $(\Delta \mathbf{P}_t, \Delta \mathbf{P}_{t+1}, \dots, \Delta \mathbf{P}_{t+L-1})$  描述它。其中,  $\Delta \mathbf{P}_t$  是连续帧之间的位移。为了使描述轨迹的方式不依赖其长度,我们将整个轨迹进行归一化,见式(5-6)。

$$T = \frac{(\Delta \mathbf{P}_t, \Delta \mathbf{P}_{t+1}, \dots, \Delta \mathbf{P}_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\Delta \mathbf{P}_j\|} \quad (5-6)$$

其中,  $\|\cdot\|$  表示 L2 范数。这个归一化过程会得到一个 30 维的轨迹描述符, 因为每个位移向量  $\Delta P_i$  有两个维度。

虽然轨迹形状信息是有价值的, 但为了提供一个更加鲁棒的特征描述, 我们还需要利用其他的描述符。DT 采用了三种光流描述符: HOG, HOF 和 MBH。如图 5.2 右侧所示, 首先围绕特征点轨迹我们构造一个  $N \times N \times L$  的立方体, 其中,  $L$  是帧的长度,  $N$  是像素的大小, 通常取 32。此立方体进一步被细分成  $n_\sigma \times n_\sigma \times n_\tau$  的时空网格, 常见设置是  $n_\sigma = 2$  和  $n_\tau = 3$ 。每一个小立方体单元都会计算 HOG、HOF 和 MBH 描述符。描述符的维度  $D$  可以由以下方式得到:  $D = n_\sigma * n_\sigma * n_\tau * b$ 。其中,  $b$  表示量化方向的数量。在计算 HOG 和 MBH 描述符时,  $b$  为 8; 而在计算 HOF 描述符时, 考虑到了一个额外的零度方向, 所以  $b$  为 9。因此, 可以得到: 轨迹描述符为 30 维; HOG 为 96 维; HOF 为 108 维; MBH 为 96 维。最后, 这些描述符被串联起来。为了得到一个固定长度的特征表示, 我们通常会使用 Bag-of-Features(BoF)编码。

## 2. iDT 算法对 DT 算法的改进

DT 算法在基于光流技术跟踪特征点时, 并未完全考虑相机移动带来的影响。而这一影响在某些情况下可能对于动作识别是无益的。为了解决这一问题, iDT 算法对 DT 算法进行了若干重要的改进。以下是这些改进的详细说明。

### 1) 增加相机运动估计

由于相机的移动, 背景可能会带有大量的相机轨迹信息。参考图 5.3(d), 白色的轨迹就是因相机移动产生的, 而这些轨迹对于动作识别可能是无关的或误导的。假如我们可以估算出相机的运动, 那么就能够从整体轨迹中剔除由相机移动产生的部分, 仅保留对动作识别有帮助的轨迹。如图 5.3(b) 和图 5.3(c) 的第二、三行展示了在移除相机运动轨迹之前和之后的对比。一般来说, 去除这些运动轨迹后的图像更有利于动作识别。

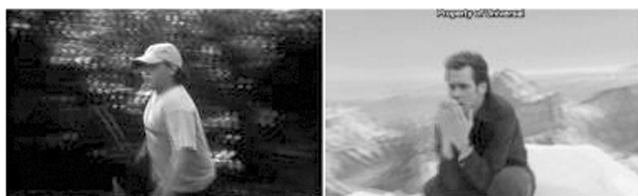
相邻两帧之间的变化往往是微小的, 这就意味着它们之间存在某种关联。iDT 算法假设这种关联是单应性的, 即一个帧上的点可以通过一个单应性矩阵映射到另一帧上的对应点, 并采用 RANSAC 方法估计这种单应性。为了选择充足且分布均匀的特征点进行匹配, iDT 在特征点的寻找上, 结合了 SURF 算法和 Shi-Tomasi 角点检测算法。

### 2) 特征正则化方式的改进

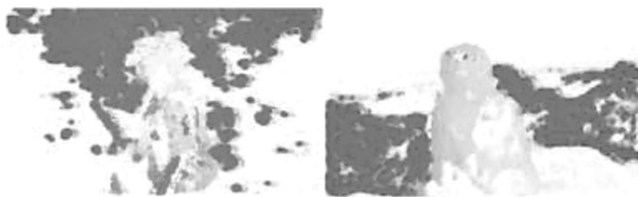
DT 算法采用的是 L2 归一化方式, 而 iDT 算法则优化为 L1 归一化方式, 使得特征更加稳定, 并对噪声更具有鲁棒性。

### 3) 特征编码方式的改进

DT 算法采用的是 BoF 编码方式, 而 iDT 算法则改进为使用 FV(Fisher Vector)编码。相较于 BoF 编码, FV 编码更加灵活, 其可以更好地抑制噪声, 具有更强的表达能力, 且不依赖于固定词袋大小。



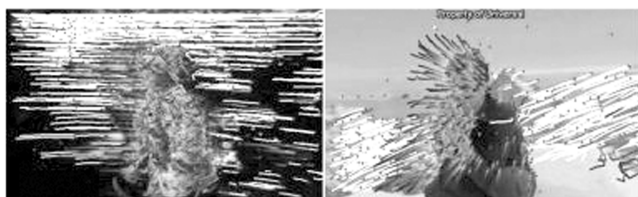
(a) 重叠的两个连续帧的图像



(b) 两个帧之间的光流



(c) 去除相机运动后的光流



(d) 白色轨迹为相机运动产生

图 5.3 相机运动对光流的影响

### 5.1.3 三维尺度不变特征变换

尺度不变特征变换 (Scale-Invariant Feature Transform, SIFT) 是一种在计算机视觉和图像处理中常用的特征描述和检测方法。SIFT 算法具有以下几个特点。

- 旋转不变性。通过计算关键点周围的特征梯度直方图,并为每个关键点分配主方向,SIFT 可以实现对图像旋转的不变性。也就是说,即使图像旋转了一定角度,SIFT 仍然可以检测到相同的特征。
- 尺度不变性。通过使用高斯金字塔构建图像的尺度空间,SIFT 能在不同尺度下找到关键点,从而实现对图像尺度的不变性。这意味着,无论图像被缩小或放大到何种程度,SIFT 都能准确地识别出相同的特征。

SIFT 是一种高度鲁棒的特征描述符,但主要针对 2D 图像设计。处理视频数

据时,传统的 2D-SIFT 可能会失效。为了解决这一问题,Scovanner 等提出一种 3D 版本的 SIFT 描述符,称为 3D-SIFT。如图 5.4 所示,从左至右分别展示了 2D-SIFT 描述符、多个时间点的 2D-SIFT 描述符以及 3D-SIFT 描述符。可以看到,3D-SIFT 描述符相当于在 2D-SIFT 描述符的基础上,增加了时间维度,使其能描述三维空间内的特征。

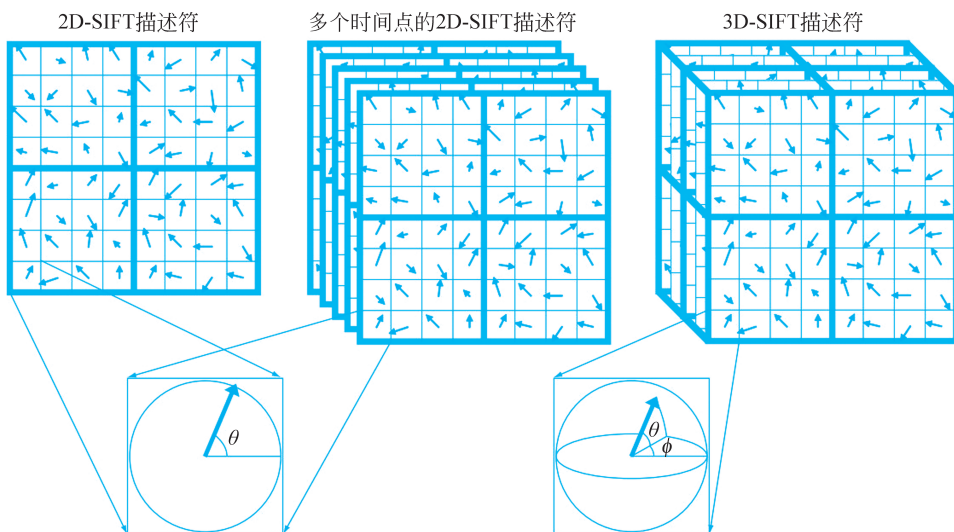


图 5.4 2D-SIFT 描述符与 3D-SIFT 描述符

3D-SIFT 描述符的计算步骤可以概括为以下三部分。

#### 1) 时空梯度方向计算

在兴趣点的 3D 邻域中计算每个像素点的时空梯度大小和方向。这包括三个维度:两个空间维度  $x$  和  $y$ ,以及一个时间维度  $t$ 。通过这些信息,可以得到每个像素点在三维空间中时空梯度的大小  $m_{3D}$ ,以及两个方向值  $\theta$  和  $\phi$ ,具体见式(5-7)。

$$\begin{cases} m_{3D}(x, y, t) = \sqrt{L_x^2 + L_y^2 + L_t^2} \\ \theta(x, y, t) = \arctan\left(\frac{L_y}{L_x}\right) \\ \phi(x, y, t) = \arctan\left(\frac{L_t}{\sqrt{L_x^2 + L_y^2}}\right) \end{cases} \quad (5-7)$$

其中,  $L_x$ 、 $L_y$  表示像素点在空间方向上的梯度,  $L_t$  表示像素点在时间方向上的梯度。

#### 2) 构建加权直方图

将  $\theta$  和  $\phi$  划分为同等大小的区间(称为 bin),并使用经线和纬线法构建加权直方图。这个过程需要按照立体角  $\omega$ (立体角:平面角在三维空间中的扩展,用于刻画物体的三维空间角度)进行归一化,以得到一个均匀的网格。然后,在每个 bin

内统计相应角度范围内的梯度值,即  $m_{3D}$ ,并根据高斯加权项赋予不同权重,见式(5-8)。

$$\text{hist}(i_\theta, i_\phi) = \frac{1}{\omega} m_{3D}(x', y', t') e^{\frac{-((x-x')^2 + (y-y')^2 + (t-t')^2)}{2\sigma^2}} \quad (5-8)$$

其中,  $e^{\frac{-((x-x')^2 + (y-y')^2 + (t-t')^2)}{2\sigma^2}}$  是一个高斯加权项,用于给邻域中不同位置的像素点赋予不同的权重,  $\omega$  表示每个直方图 bin 的立体角,  $i_\theta$  表示角度方向的索引,  $i_\phi$  表示位置方向的索引。

在最后构建的加权直方图中,我们寻找其峰值,代表了兴趣点的显著梯度方向。这个主要方向被保存下来,并用于对关键点附近的区域进行旋转,确保 3D-SIFT 在时空中具有旋转稳定性。

**注意:** 直方图是对数据进行统计的一种方法,并且将统计值组织到一系列事先定义好的 bin 中。其中,bin 为直方图中经常用到的一个概念,可以译为“直条”或“组距”,其数值是从数据中计算出的特征统计量,这些数据可以是梯度、方向、色彩或任何其他特征。

### 3) 构建 3D-SIFT 描述符

为了构建 3D-SIFT 描述符,首先要调整兴趣点的 3D 邻域,以使主方向对齐至  $\theta = \phi = 0$ 。为达到这一目的,需要将兴趣点周边的每个像素位置与矩阵公式(5-9)相乘:

$$\begin{bmatrix} \cos\theta \cos\phi & -\sin\theta & -\cos\theta \sin\phi \\ \sin\theta \cos\phi & \cos\theta & -\sin\theta \sin\phi \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \quad (5-9)$$

为了精确描述兴趣点附近的特征,我们采取了密集采样策略。如图 5.4 最右侧展示,兴趣点附近的区域被划分为 8 个子块。在每个子块中,我们使用五线性插值方法来描述特征,其中五个维度分别是 3D 空间内的三个坐标和两个梯度方向  $(\theta, \phi)$ 。在 3D-SIFT 策略中,我们采用了  $2 \times 2 \times 2$  和  $4 \times 4 \times 4$  两种配置的子直方图,以及  $8 \times 4$  的直方图来表示  $\theta$  和  $\phi$ 。具体来说,  $2 \times 2 \times 2$  子直方图配置意味着在 3D 空间中,我们将空间分为  $2 \times 2 \times 2$  个子块;而  $8 \times 4$  的直方图表示梯度方向被分为 8 个方向,每个方向对应一个子直方图,并在空间上取样 4 个点。通过串联所有直方图,我们可以得到 256 维和 2048 维的 3D-SIFT 描述符,这些描述符详细地捕获了兴趣点周边的特征。描述符的维度是基于子直方图数量来定义的,维度  $D$  由公式  $D = \text{直方图子区域的个数} \times \text{子直方图中区域的个数}$  给出。例如,对于  $2 \times 2 \times 2$  的子直方图配置,描述符维度是  $8 \times 2 \times 2 \times 2 \times 4 = 256$ ;对于  $4 \times 4 \times 4$  的子直方图,描述符维度是  $8 \times 4 \times 4 \times 4 \times 4 = 2048$ 。

## 5.1.4 三维方向梯度直方图

方向梯度直方图(Histogram of Oriented Gradient, HOG)是一个强大的图像

特征描述符,对图像中的几何和光学变形具有高度稳定性。当将其从二维图像扩展到三维空间时,得到的特征描述符被称为 HOG-3D(Histogram of Oriented Gradient 3D)。这种三维描述符为计算机视觉任务,尤其是为动作识别和视频分析提供了时间域的扩展。

如图 5.5 所示,HOG-3D 的计算步骤如下。

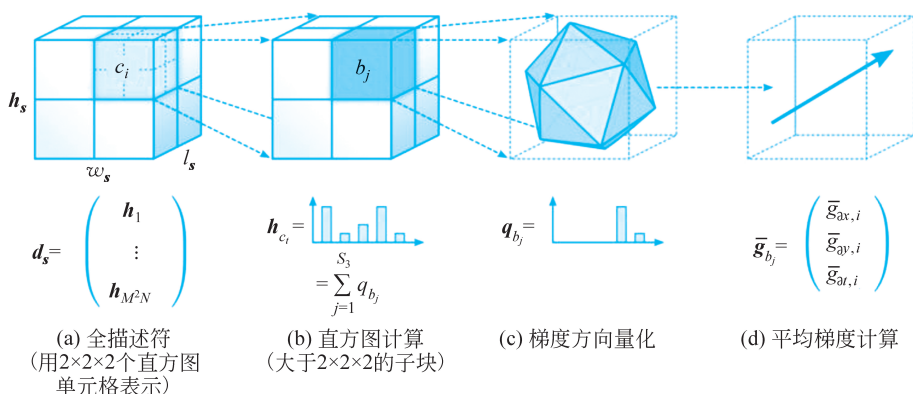


图 5.5 彩图

图 5.5 HOG-3D 描述符计算

### 1) 确定局部区域

以视频序列中的采样点  $s = (x_s, y_s, t_s, \sigma_s, \tau_s)^T$  为中心,确定一个宽度为  $w_s$ 、高度为  $h_s$ 、长度为  $l_s$  的立方体区域  $Q$ ,如图 5.5 左侧立方体所示。这里,  $w_s = h_s = \sigma_0 \sigma_s$ ,  $l_s = \tau_0 \tau_s$ , 其中  $\sigma_s$  和  $\tau_s$  分别表示采样点在空间尺度和时间尺度的大小;  $\sigma_0$  和  $\tau_0$  分别为调节空间尺度和时间尺度的缩放因子。将区域  $Q$  分割为  $M \times M \times N$  个小单元  $C$ (图 5.5 左侧深蓝色立方体),一般选取  $M=4, N=4$ 。对于每个小单元  $c_i (i=1, 2, \dots, M^2 N)$ ,它进一步被划分为  $S^3$  个子块(通常  $S=2$ )。每个子块被标记为  $b_j (j=1, 2, 3, \dots, S^3)$ ,如图 5.5(b)中深蓝色立方体所示。

### 2) 平均梯度向量计算

对于每个子块  $b_j$ ,计算其对应的平均梯度向量  $\bar{\mathbf{g}}_{b_j} = (\bar{g}_{b_j \partial x}, \bar{g}_{b_j \partial y}, \bar{g}_{b_j \partial t})^T$ ,具体的计算方法见式(5-10)。

$$\begin{aligned} \bar{g}_{b_j \partial x} = & [iv_{\partial x}(x + w_s, y + h_s, t + l_s) - iv_{\partial x}(x, y + h_s, t + l_s) - \\ & iv_{\partial x}(x + w_s, y, t + l_s) + iv_{\partial x}(x, y, t + l_s)] - \\ & [iv_{\partial x}(x + w_s, y + h_s, t) - iv_{\partial x}(x, y + h_s, t) - \\ & iv_{\partial x}(x + w_s, y, t) + iv_{\partial x}(x, y, t)] \end{aligned} \quad (5-10)$$

其中,  $v(x, y, t)$  是视频序列,  $w_s, h_s, l_s$  分别为立方体区域  $Q$  的宽度、高度、长度,而  $v_{\partial x}, v_{\partial y}$  和  $v_{\partial z}$  分别是视频序列在  $x, y$  和  $t$  方向上的偏导数。

### 3) 平均梯向量量化

每个子块  $b_j$  的平均梯度向量量化后得到的是量化的梯度方向  $\mathbf{q}_{b_j}$ 。该量化过程涉及使用正多面体(通常  $n=12$  或  $20$ )。具体量化过程如下。

(1) 根据子块  $b_j$  的中心点的位置以及其长度、宽度、高度,计算出正  $n$  面体的每个面的中心坐标  $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n)^T$ , 其中  $\mathbf{p}_i = (x_i, y_i, t_i)^T$  表示第  $i$  个面的中心坐标。

(2) 将平均梯度向量  $\bar{\mathbf{g}}_{b_j}$  投影到通过原点(正  $n$  面体的重心)和所有面中心位置的坐标轴上,再对投影结果进行 L2 范数归一化,并将得到的投影记作  $\hat{\mathbf{q}}_{b_j}$ ,  $\hat{\mathbf{q}}_{b_j} =$

$$(\hat{q}_{b_1}, \hat{q}_{b_2}, \dots, \hat{q}_{b_n})^T = \frac{\mathbf{P} \cdot \bar{\mathbf{g}}_{b_j}}{\|\bar{\mathbf{g}}_{b_j}\|_2}。$$

(3) 为了使表示更紧凑,我们取投影值的绝对值,实现将方向相反的投影合并到同一个 bin 中,进而得到一个方向数量减半的直方图。

(4) 为了确保平均梯度向量  $\bar{\mathbf{g}}_{b_j}$  与通过原点和面中心的坐标轴完全对齐时只投影给直方图中的一个 bin,对投影向量  $\hat{\mathbf{q}}_{b_j}$  进行阈值处理:若投影向量  $\hat{\mathbf{q}}_{b_j}$  减去阈值  $r$  小于 0,即  $\hat{\mathbf{q}}_{b_j} - r < 0$ ,那么所有的负值元素被设为 0;否则保留原值。其中,阈值  $r = \mathbf{m}_i^T \cdot \mathbf{m}_j$ ,而  $\mathbf{m}_i$  和  $\mathbf{m}_j$  为相邻的两个坐标轴在相同维度空间的方向向量。

(5) 最后,将梯度大小信息与经过处理的方向信息结合起来,得到一个综合的

梯度向量,记作  $\mathbf{q}_{b_j}, \mathbf{q}_{b_j} = \frac{\|\hat{\mathbf{g}}_{b_j}\|_2 \cdot \hat{\mathbf{q}}'_{b_j}}{\|\hat{\mathbf{q}}'_{b_j}\|_2}$ ,其中  $\hat{\mathbf{q}}'_{b_j}$  表示经过阈值处理后的投影向量。

#### 4) 梯度投影求和

经过前面的平均梯度量化操作,每个子块都能得到一个  $n$  维的描述符。为了方便描述,将同一个单元中子块得到的描述符进行相加,得到该单元的描述符,其特征维度仍为  $n$ 。

#### 5) 构建 HOG-3D 描述符

将  $Q$  中的  $M^2 N$  个单元的描述符连接起来,得到 HOG-3D 描述符的最终特征向量。

假设将区域  $Q$  划分成  $M \times M \times N$  个单元,  $M=4, N=4$ ,使用正  $n$  面体进行量化  $n=12$ ,那么最终得到的特征维度为  $M \times M \times N \times n = 4 \times 4 \times 4 \times 12 = 768$ 。

### 5.1.5 立方体

为了解决 Harris-3D 角点检测的某些局限性,如对噪声敏感、计算复杂度高, Dollar 等提出了一种基于 Gabor 滤波器的时空兴趣点检测方法,被称为立方体。此方法在空间域中采用高斯滤波器,时间域则采用一维的 Gabor 滤波器。

Cuboid 特征提取分为五个步骤:兴趣点检测、生成 Cuboid、构建局部直方图、生成 Cuboid 原型、生成描述子。

#### 1) 兴趣点检测

首先,找到图像中的“兴趣点”。在空间中,我们对图像的每像素计算一个响应函数。响应函数的局部最大值对应的位置即兴趣点。这个想法可以被扩展到时间