第3章

# 轨迹分析

通过跟踪点或目标沿线(直线或曲线)的运动,可以获得点或目标的(动态)轨迹[杨2024]、[张2024]。在时空行为理解中,对动态轨迹的学习和分析[Morris 2008]可以刻画场景中各个运动目标的行为,并提供对场景变化和运动状态的把握。

路径和轨迹密切相关,但还是有所区别。路径主要考虑空间位置,是空间点的集合。轨迹通常还要考虑在各个空间位置的速度、加速度等,也可以称为活动路径,以强调运动的影响。一般情况下,路径主要考虑全局联系,而轨迹还关注局部特征(如果借助像素间联系的说法,可以认为路径基本对应通路,而轨迹还有连通的要求)。

轨迹分析工作包括对轨迹的检测、建模、学习、分类、识别等。

本章主要内容包括2节:

3.1节介绍对轨迹的检测(及跟踪)、建模、学习以及活动分析的基本原理和步骤;

3.2节介绍一种借助轨迹特征聚类树对不同运动的轨迹进行分类识别的方法。

## 3.1 轨迹学习和分析

图3.1.1为对视频进行动态轨迹学习和分析的流程框图。借助输入视频,先要对目标进行检测。这里既可以使用静止的摄像机对运动的目标进行检测和跟踪;也可以使用运动的摄像机对静止或运动的人进行检测和跟踪,如将摄像机安置在汽车上对路旁的行人进行检测,可以参见[贾2007]。为了获取不同目标的运动轨迹,可以采用不同的方法,如检测车辆轨迹的一个工作可以参见[赵2024],而检测行人轨迹的一个工作可以参见[王2024]。另外,还可以考虑关联轨迹片段、克服遮挡问题、构成完整轨迹,可以参见[孙

2024]。将连续检测和跟踪的结果结合起来，就可以得到目标的运动轨迹。利用这样的轨迹就可以自动构建场景模型。最后，使用这样的模型分析运动的情况，还可以提供对活动的标注。


图 3.1.1 对视频进行动态轨迹学习和分析的流程框图

在场景建模中，可以先将有事件发生的图像区域位置定义为**兴趣点**(POI)，在接下来的学习步骤中再定义**活动路径**(AP)。该路径刻画目标是如何在兴趣点之间运动/游历的。这样构建的模型称为 POI/AP 模型。

POI/AP 模型中学习的主要工作如下。

(1) **活动学习**：可以通过比较**轨迹**进行活动学习，不同轨迹的长度可能不同，关键是要保持对相似性的直观认识。

(2) **适应**：研究管理 POI/AP 模型的技术。要使技术能在线地适应如何增添新发生的活动、删除不再继续的活动，并验证模型。

(3) **特征选择**：确定对特定任务正确的动力学表达层次。例如，仅使用空间位置信息就可以确定汽车行驶的路线，若希望检测事故，则还需要速度信息。

### 3.1.1 场景建模

借助动态轨迹对场景的自动建模包括目标跟踪、兴趣点检测和活动路径学习 3 个要点[Makris 2005]。

#### 1. 目标跟踪

对目标的**跟踪**需要在每一帧中对可以观察到的各个目标进行身份维护。例如，在 $T$ 帧视频中被跟踪的目标会生成一系列可以推断的跟踪状态：

$$S_T = \{s_1, s_2, \cdots, s_T\} \tag{3.1.1}$$

式中：各个 $s_t$ 可以描述位置、速度、外观、形状等目标特性。以这些特性为基础的轨迹信息构成了进一步分析的基石。认真分析这些信息，即可以识别和理解活动。

#### 2. 兴趣点检测

场景建模的首要任务是找出图像中的感兴趣区域。在指示跟踪目标的地图中，这些区域对应图中的节点。通常考虑的两种节点为入/出区域和停止区域。以一位教师去教

室授课为例，前者对应教室门，后者对应讲台。

入/出区域是目标进入或离开**视场**(FOV)，或者被跟踪目标出现或消失的位置。这些区域经常借助 2-D 的**高斯混合模型**(GMM)建模，$Z \sim \sum_{i=1}^{W} w_i N(\mu_i, \sigma_i)$，其中包括 $W$ 个分量。这个模型可以使用期望最大值(EM)算法求解。进入的点数据包括第 1 个跟踪状态确定的位置，而离去的点数据包括最后 1 个跟踪状态确定的位置。考虑用密度准则进行区分，在状态 $i$ 的混合密度定义为

$$d_i = \frac{w_i}{\pi\sqrt{|\sigma_i|}} > T_{\mathrm{d}} \tag{3.1.2}$$

这被用于测量高斯混合的紧凑程度。其中，阈值

$$T_{\mathrm{d}} = \frac{w}{\pi\sqrt{|\boldsymbol{C}|}} \tag{3.1.3}$$

指示信号聚类的平均密度。$w$ 为用户定义的权重，$0<w<1$；$\boldsymbol{C}$ 为在区域数据集合中所有点的协方差矩阵。紧凑的混合指示正确的区域，而宽松的混合指示因跟踪中断而导致的跟踪噪声。

停止区域源于场景地标点，即目标在一段时期内趋于固定的位置。停止区域可以采用两种方法确定：一是在该区域被跟踪点的速度低于某个事先确定的较低的阈值；二是所有被跟踪点至少在某个时间段内保持在一个有限的距离环中。通过定义半径和时间常数，采用第一种方法仍然可能包含运动较慢的目标；而采用第二种方法可以保证目标保持在特定范围里。对活动进行分析时，除了确定位置，也要把握在每个停止区域花费的时间。

### 3. *活动路径学习*

为了理解行为，需要确定**活动路径**。可以使用兴趣点从训练集中滤除虚警或跟踪中断的噪声，只保留进入活动区域后开始并在活动终止区域内结束的轨迹。经过活动区域的跟踪轨迹分为进入活动区域和离开活动区域的两段，一个活动要定义在目标开始动作和结束动作两个感兴趣点之间。

为了区分随时间变化的动作目标(如沿着人行道走或跑的行人)，需要在路径学习中加入时间动态信息。图 3.1.2 给出活动路径学习算法的 3 种基本结构，它们的主要区别包括输入种类、运动矢量、轨迹(或视频片段)，以及运动抽象的方式。在图 3.1.2(a)中，输入是时刻 $t$ 的单个轨迹，路径中的各个点隐含地进行了时间排序。在图 3.1.2(b)中，一个完整的轨迹被用作学习算法的输入以直接建立输出的路径。图 3.1.2(c)中将路径按照视频时序进行分解，**视频片段**(VC)被分解为一组动作单词以描述活动，或者说视频片段根据动作单词的出现而被赋予某种活动的标签。

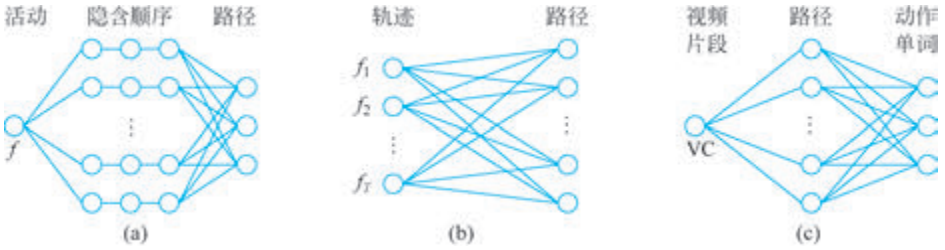


图 3.1.2 活动路径学习算法的 3 种基本结构

### 3.1.2 轨迹学习

为了刻画目标运动的情况，可以使用对目标运动情况进行动态测量的序列。例如，常用的轨迹表达就是一个运动序列：

$$G_T = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_T\} \tag{3.1.4}$$

其中，运动矢量：

$$\boldsymbol{g}_t = [x^t, y^t, v_x^t, v_y^t, a_x^t, a_y^t]^{\mathrm{T}} \tag{3.1.5}$$

表示从跟踪中获得的目标在时刻 $t$ 的动态参数，包括位置$[x, y]^{\mathrm{T}}$、速度$[v_x, v_y]^{\mathrm{T}}$ 和加速度$[a_x, a_y]^{\mathrm{T}}$。

仅使用轨迹就可能以无监督的方式学习 AP，其步骤如图 3.1.3 所示。预处理步骤要建立用于聚类的轨迹，主要工作包括归一化和降维；聚类步骤可以提供全局和紧凑的路径模型表达，主要工作是在聚类过程中测量距离或相似性；建模步骤要给出确定的路径，主要工作包括确定聚类中心以及分解子路径。尽管图 3.1.3 中有 3 个分离的顺序步骤，它们也常结合在一起。下面对 3 个步骤分别给予详细解释。


图 3.1.3 轨迹学习步骤

#### 1. 轨迹预处理

路径学习研究中的大部分工作需要获得适合聚类的轨迹。当进行跟踪时，主要困难源于时间变化的特性，这会导致轨迹长度不一致。此时需要采取步骤，保障不同尺寸的输入之间可以进行有意义的比较。另外，轨迹表达在聚类中应该能直观地保持原始轨迹的相似性。

**轨迹预处理**主要包括两方面内容。

(1) 归一化：目的是保证所有轨迹具有相同的长度 $L_t$。两种简单的技术是填零和扩展，填零是在较短的轨迹后面增加一些零项，扩展是将原始轨迹最后时刻的部分延伸扩展至需要的长度，两者都有可能将轨迹空间扩展得非常大。除了检查训练集确定轨迹的长度 $L_t$ 之外，也可以利用先验知识进行重采样和平滑。重采样结合插值能保证所有轨迹具有相同的长度 $L_t$。平滑可用于消除噪声，平滑后的轨迹也可以通过插值和采样以得到需要的长度。

(2) 降维：降维将轨迹映射到新的低维空间，从而可以使用更加鲁棒的聚类方法。这里可以通过假设一个轨迹模型并确定能最有效地描述该模型的参数来实现。常采用的技术有矢量量化、多项式拟合、多分辨率分解、隐马尔可夫模型(HMM)、子空间方法、频谱方法及核方法等。

通过限制唯一轨迹的数量来实现矢量量化。若忽略轨迹动力学并且仅仅基于空间坐标，则可以将轨迹看作简单的 2-D 曲线，并且可以用 $m$ 阶的最小均方多项式近似(各 $w$ 为权系数)：

$$x(t)=\sum_{k=0}^{m} w_k t^k \tag{3.1.6}$$

在频谱方法中可以对训练集构建相似矩阵 $\boldsymbol{S}$，其中元素 $s_{ij}$ 表示轨迹 $i$ 和轨迹 $j$ 之间的相似性。还可以构建拉普拉斯矩阵，即

$$\boldsymbol{L}=\boldsymbol{D}^{-1/2}\boldsymbol{S}\boldsymbol{D}^{-1/2} \tag{3.1.7}$$

式中：$\boldsymbol{D}$ 为对角矩阵，其第 $i$ 个对角元素为 $\boldsymbol{S}$ 中第 $i$ 行元素之和。

通过分解 $\boldsymbol{L}$ 可以确定其最大的 $K$ 个本征值。将对应的本征矢量放入一个新矩阵，其行对应在频谱空间变换后的轨迹，而频谱轨迹可以使用 $K$-均值方法获得。

多数研究者将轨迹归一化与降维结合来处理原始轨迹，保证其可以使用标准的聚类技术。

### 2. 轨迹聚类

聚类是在没有标记的数据中确定结构的常用机器学习技术。它在观察场景时收集运动轨迹并且将其归入类似的类别。为了产生有意义的聚类，**轨迹聚类**过程要考虑 3 个问题：①定义一个距离(对应相似性)测度；②确定聚类更新的策略；③对聚类进行验证。分成以下两部分讨论。

(1) 距离/相似测量：聚类技术依赖距离(相似)测度的定义。前面讨论预处理时，轨迹聚类的一个主要问题是相同活动产生的轨迹长度可能不同。解决这个问题既可以采用预处理方法，也可以定义一个与尺寸独立的距离测度(如果两个轨迹 $G_i$ 和 $G_j$ 长度相同)，即

$$d_{\mathrm{E}}(G_i,G_j)=\sqrt{(G_i-G_j)^{\mathrm{T}}(G_i-G_j)} \tag{3.1.8}$$

若两个轨迹 $G_i$ 和 $G_j$ 长度不同，则对欧几里得距离不随尺寸变化的改进是比较两个长度分别为 $m$ 和 $n(m>n)$ 的轨迹矢量，并使用最后点 $\boldsymbol{g}_{j,n}$ 的累积失真：

$$d_{ij}^{(\mathrm{c})}=\frac{1}{m}\left\{\sum_{k=1}^{n}d_{\mathrm{E}}(\boldsymbol{g}_{i,k},\boldsymbol{g}_{j,k})+\sum_{k=1}^{m-n}d_{\mathrm{E}}(\boldsymbol{g}_{i,n+k},\boldsymbol{g}_{j,n})\right\} \tag{3.1.9}$$

欧几里得距离比较简单，但是存在时间偏移的情况下效果不佳，因为仅对准的序列可能匹配。为此，可以考虑使用豪斯道夫距离[章 2024]。另外，还有一种距离测度，其不依赖完整的轨迹(不考虑野点)。假设轨迹 $G_i=\{\boldsymbol{g}_{i,k}\}$ 和 $G_j=\{\boldsymbol{g}_{j,l}\}$ 的长度分别为 $T_i$ 和 $T_j$，则

$$D_{\mathrm{o}}(G_i,G_j)=\frac{1}{T_i}\sum_{k=1}^{T_i}d_{\mathrm{o}}(\boldsymbol{g}_{i,k},G_j) \tag{3.1.10}$$

式中：

$$d_{\mathrm{o}}(\boldsymbol{g}_{i,k},G_j)=\min_{l}\left[\frac{d_{\mathrm{E}}(\boldsymbol{g}_{i,k},\boldsymbol{g}_{j,l})}{Z_l}\right],\quad l\in\{\lfloor(1-\delta)k\rfloor,\cdots,\lceil(1+\delta)k\rceil\} \tag{3.1.11}$$

其中：$Z_l$ 为归一化常数，也是点 $l$ 处的方差；$\delta$ 为小于 1 的正数。

$D_{\mathrm{o}}(G_i,G_j)$ 用于比较轨迹与存在的聚类。比较两个轨迹，可以使用 $Z_l=1$。这样定义的距离测度是从任意点到其最佳匹配之间的平均归一化距离，此时最佳匹配处于中心位于点 $l$、宽度为 $2\delta$ 的滑动时间窗口中。

(2) 聚类过程和验证：预处理后的轨迹可以使用非监督的学习技术进行组合，并且能将轨迹空间分解为感知上相似的聚类(如道路)。对聚类的学习方法有迭代优化、在线自适应、分层方法、神经网络和共生分解。

借助聚类算法学习的路径需要进一步验证，因为真实的类别数未知。多数聚类算法需要给定所期望的类别数 $K$ 一个初始值，但这个值通常不正确。为此可以对不同的 $K$ 分别进行聚类，再取最好结果对应的 $K$ 作为真正的聚类数。这里判断准则可以使用**紧密和分离准则**(TSC)，以比较不同聚类中相应轨迹之间的距离。若给定训练集 $D_T=\{G_1,G_2,\cdots,G_M\}$，则有

$$\mathrm{TSC}(K)=\frac{1}{M}\frac{\sum_{j=1}^{K}\sum_{i=1}^{M}f_{ij}^2d_{\mathrm{E}}^2(G_i,c_j)}{\min_{ij}d_{\mathrm{E}}^2(c_i,c_j)} \tag{3.1.12}$$

其中：$f_{ij}$ 是轨迹 $G_i$ 对聚类 $C_j$(其中的样本用 $c_j$ 表示)的模糊隶属度。

### 3. 轨迹建模

轨迹聚类后，可以根据得到的路径建立**图**模型，以进行有效的推理。路径模型是对

聚类的紧凑表达。能使用两种方式对**轨迹建模**：一是考虑完整的路径，端点到端点的路径上不仅有平均的重心/中心线，两边还有包络指示路径范围，沿着路径还可能有一些中间状态给出测量顺序，如图 3.1.4(a)所示；二是将路径分解为若干子路径，或者将路径表示为包含子路径的树，预测路径的概率从当前节点依次分别指向叶节点，如图 3.1.4(b)所示。

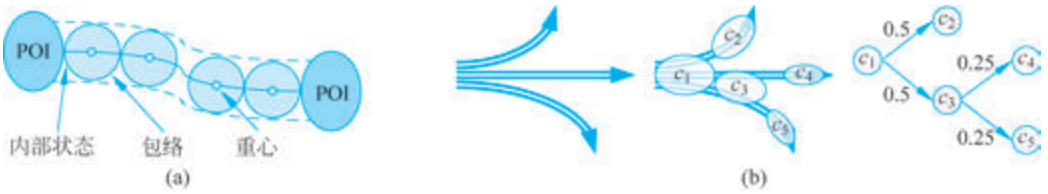


图 3.1.4　两种轨迹建模方式

### 3.1.3　活动分析

一旦建立了场景模型，就可以对目标的活动和行为进行分析。下面借助监控视频分析来解释，其一个基本功能就是对感兴趣的事件进行验证。一般来说，只有在特定环境下才容易针对性地定义是否感兴趣。例如，停车管理系统关注是否还有空的车位，而智能会议室系统关心的是人员之间的交流。除了识别特定的行为外，还需要检查所有非典型的事件。通过对场景进行长时间观察，系统可以进行一系列活动分析，从而学习到哪些是感兴趣的事件。

典型的活动分析如下。

(1) **虚拟篱笆**：任何监控系统都有一定的监控范围，在该范围的边界上设立哨兵，即可以对进入范围内发生的事件进行预警。这相当于在监控范围的边界建立了虚拟篱笆，一旦有入侵就触发分析。例如，控制高分辨率的**云台摄像机**(PTZ)获取入侵处的细节，开始对入侵数量进行统计等。

(2) **速度分析**：虚拟篱笆仅利用了位置信息，借助目标跟踪技术还可以获得动态信息，实现基于速度的预警，如车辆超速或路面堵塞。

(3) **轨迹分类**：速度分析仅利用了当前跟踪的数据，实际应用中还可以利用历史运动模式获得的活动路径。新出现目标的行为可以借助**最大后验**(MAP)路径描述：

$$L^{*}=\arg\max_{k} p(l_k \mid G)=\arg\max_{k} p(G,l_k)p(l_k) \tag{3.1.13}$$

这有助于确定哪个活动路径能够最充分地解释新的数据。因为先验路径分布 $p(l_k)$可以用训练集来估计，所以问题就简化为采用隐马尔可夫模型进行最大似然估计。

(4) **异常检测**：异常事件的检测常常是监控系统的重要任务。活动路径能够指示典

型的活动，如果新的轨迹与已有轨迹不相符，就能发现异常。异常模式可以借助智能阈值化进行检测：

$$p(l^* \mid G) < L_l \tag{3.1.14}$$

式中：与新轨迹 $G$ 最相像的活动路径 $l^*$ 的值仍小于阈值 $L_l$。

(5) **在线活动分析**：能够在线分析、识别、评价活动比使用整个轨迹描述运动更重要。一个实时系统应该能够根据尚不完整的数据对正在发生的行为进行快速推理(经常基于图模型)。这里包括以下两种情况：

① 路径预测：利用已有的跟踪数据预测未来的行为，并在收集到更多数据时细化预测。可以利用非完整的轨迹对活动进行预测，这表示为

$$\hat{L} = \underset{j}{\operatorname{argmax}} p(l_j \mid W_t G_{t+k}) \tag{3.1.15}$$

式中：$W_t$ 为窗函数；$G_{t+k}$ 为直到当前时间 $t$ 的轨迹及 $k$ 个预测的未来跟踪状态。

② 跟踪异常：除了将整个轨迹划归异常外，还需要在非正常事件发生时就能检测到，为此，可以用 $W_t G_{t+k}$ 代替式(3.1.14)中的 $G$ 实现。窗函数 $W_t$ 并不必须与预测中相同，且阈值可能需要根据数据量进行调整。

(6) **目标交互刻画**：更高层次的分析期望进一步描述目标之间的交互。与异常事件类似，严格地定义目标交互也比较困难。在不同的环境下，不同目标之间存在着不同类型的交互。以汽车碰撞为例，每辆汽车有其空间尺寸，可以将其看作个人空间。汽车行驶时，个人空间需要在汽车周围增加一个最小安全距离(最小安全区)，这个个人空间会随着运动而发生改变，速度越快，最小安全距离增加越多(尤其是在行驶方向上)。利用路径进行汽车碰撞评估示意图如图 3.1.5 所示，其中个人空间用圆表示，而安全区域随速度(包括大小和方向)改变而改变。如果两辆车的安全区域有交会，那么有可能发生碰撞，因此也可以借此进行分析以帮助规划行车路线。

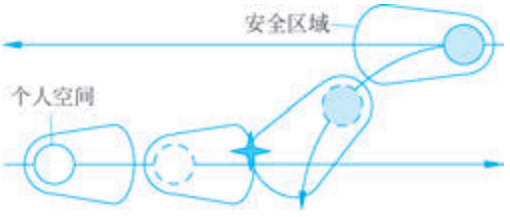


图 3.1.5 利用路径进行汽车碰撞评估示意图

最后需要指出：对于简单的活动，仅仅依靠目标位置和速度就能够开展分析；对于复杂的活动，还需要更多的测量，如加入剖面的弯曲度以判别古怪的运动轨迹。为了提供对活动和行为更全面的覆盖，常常需要使用多摄像机网络。另外，活动轨迹还可能源于互相连接的部件所构成的目标(如人体)，此时活动需要相对于一组轨迹进行定义。

## 3.2 轨迹特征聚类树

基于运动轨迹可以采用不同方法对各种运动进行分类识别。下面介绍一种借助轨迹特征聚类树的方法[Chen 2016]。该方法的工作流程图如图 3.2.1 所示，主要分为训练和测试两个阶段(分别参见图中上、下两个部分)。在训练阶段，首先从训练视频中提取轨迹特征；然后根据轨迹特征的时空位置和形状信息构建轨迹聚类树结构，同时利用轨迹特征训练**高斯混合模型**；接着，利用高斯混合模型对轨迹聚类树中每个节点所包含的轨迹聚类进行**费歇尔矢量**(FV)编码，得到相应的 FV 编码树结构；最后，利用树结构矢量训练 SVM 分类器进行分类。在测试阶段，同样地，首先从测试视频中提取轨迹特征，并根据轨迹特征的时空位置和形状信息构建轨迹聚类树结构；接着对树中每个节点所包含的轨迹特征聚类进行 FV 编码，得到相应的 FV 编码树结构；最后利用训练好的 SVM 分类器借助由测试视频获得的 FV 编码树进行视频分类。


图 3.2.1 轨迹特征聚类树方法的流程图

### 3.2.1 轨迹特征提取

轨迹特征提取包括提取轨迹特征和描述特征两个主要步骤。

1. 提取轨迹特征

从视频中提取轨迹特征的思路是利用光流场对稠密采样的特征点集合进行短期运动跟踪，并在特征点轨迹的时空邻域内提取多种描述符[Wang 2013]。具体是利用 Farnebäck 算法[Farnebäck 2003]计算视频帧 $I_t$ 的稠密光流场 $\Omega_t$，以 5 个像素为步长在视频帧上进行均匀稠密采样(即 5×5 网格采样)，得到特征点集合 $\{p_t\}$。

这里关注的是运动区域的特征点轨迹。为了获得稳定可靠的特征点轨迹，需要从纹理和运动两方面考虑以滤除不符合要求的特征点。

(1) 纹理特征点。

在没有纹理的图像平滑区域，无法可靠地对特征点进行跟踪。为此，需要去除这些平滑区域的特征点[Shi 1994]。先计算特征点 $p_t$ 的自相关矩阵：

$$\boldsymbol{D}=\begin{bmatrix} d_x d_x & d_x d_y \\ d_y d_x & d_y d_y \end{bmatrix} \tag{3.2.1}$$

式中：$d_x$、$d_y$ 分别为特征点 $p_t$ 在 $x$ 和 $y$ 方向的梯度。

假设($\lambda_1$；$\lambda_2$)是矩阵 $\boldsymbol{D}$ 的特征值，则当 $\min(\lambda_1;\lambda_2)$小于设定的经验阈值时，该特征点就会被从特征点集合中去除掉。

(2) 运动特征点。

对于视频帧 $I_t$ 中的每个采样特征点 $p_t=(x_t,y_t)$，可以利用中值滤波后的光流信息来计算其在下一个视频帧 $I_{t+1}$ 的位置 $p_{t+1}$：

$$p_{t+1}=(x_{t+1},y_{t+1})=(x_t,y_t)+(M\otimes\Omega_t)\Big|_{(\bar{x}_t,\bar{y}_t)} \tag{3.2.2}$$

式中：$M$ 为中值滤波函数；模板大小为 3×3；$(\bar{x}_t,\bar{y}_t)$为特征点$(x_t,y_t)$的取整位置。

通过不断跟踪特征点在下一个视频帧的位置，可以得到一条特征点轨迹 $\boldsymbol{P}$ 矢量：

$$\boldsymbol{P}=[p_t,p_{t+1},p_{t+2},\cdots]^{\mathrm{T}} \tag{3.2.3}$$

利用光流场对特征点进行跟踪，光流计算的误差会使得特征点在跟踪的过程中偏离实际位置。为了降低这种跟踪漂移的风险，可以限定特征点跟踪的长度，如 $L=15$ 帧。在跟踪过程中，有的特征点会丢失或者跟踪提前结束，使得有些 5×5 区域中没有特征点。此时，需要在该区域补充一个采样特征点并进行下一步跟踪，以保证稠密地采样。另外，在特征点跟踪的过程中难免出现一些错误的特征点轨迹(如在静态的含有纹理的背景中或者噪声区域里)，因此需要根据轨迹的运动信息进行判断。如果轨迹的位置变化很小，或者变化超过指定的阈值，就可以认为这些轨迹是错误的，并将对应的特征点从特征点轨迹集合中去除掉。

图 3.2.2 为对举重活动视频进行特征点轨迹跟踪的示意图[Chen 2016]。图中红色圆圈表示特征点在当前时刻的位置，绿色线条表示特征点从上个时刻到当前时刻的运动轨迹。

### 2. 描述轨迹特征

在获得特征点轨迹之后，需要从轨迹的时空邻域内提取刻画表观和运动信息的描述符。先考虑轨迹的形状特征矢量：

$$\boldsymbol{T}=[\Delta p_t,\Delta p_{t+1},\cdots,\Delta p_{t+L-1}]^{\mathrm{T}} \tag{3.2.4}$$

式中：$\Delta p_t$ 为 $t$ 时刻特征点的位移，$\Delta p_t=p_{t+1}-p_t=(x_{t+1}-x_t,y_{t+1}-y_t)$。

对特征矢量 $\boldsymbol{T}$ 进行 $L_1$ 范数归一化就可以作为轨迹特征的形状描述符。将轨迹的时空邻域尺寸设为 $N\times N\times L$，其中 $N=32$ 像素，$L=15$ 帧。进一步，可以将时空邻域分成 2×2×3 的子块。对每个子块计算其特征描述符，如**梯度方向直方图**(HOG)[Dalal 2005]、**光流直方图**(HOF)[Laptev 2008]、**运动边界直方图**(MBH)[Dalal 2006]。其中，


彩图

图 3.2.2 特征点轨迹跟踪示意图

HOG 描述空域的表观信息，HOF 描述时域的运动信息，MBH 描述特征点轨迹相对运动的信息。最后，将各个子块对应的 HOG、HOF 和 MBH 描述符串接起来作为形成该轨迹最终的 HOG、HOF 和 MBH 描述符。

### 3.2.2 特征聚类树

在轨迹特征提取的基础上可以构建特征聚类树。这里通过对无监督的分层二分划分方式[Gaidon 2014]进行改进来得到数据驱动的树状分解结构，以较好地反映轨迹特征的时空分布。首先，计算轨迹特征之间的相似度矩阵 $\boldsymbol{W}\in\mathbf{R}^{N\times N}$，其中 $N$ 表示轨迹特征的数目。为了更好地反映轨迹的时空分布，采用轨迹的时空位置和形状信息来计算轨迹之间的相似度。这些特征包括：时空位置 $\boldsymbol{x}=(x_1,x_2,\cdots,x_L)$，$\boldsymbol{y}=(y_1,y_2,\cdots,y_L)$，$\boldsymbol{t}=(t,t+1,\cdots,t+L-1)$，以及空间变化信息 $\boldsymbol{v}_x=(\Delta x_1,\Delta x_2,\cdots,\Delta x_L)$，$\boldsymbol{v}_y=(\Delta y_1,\Delta y_2,\cdots,\Delta y_L)$。对于每种特征采用**径向基函数**(RBF)核，即高斯核函数 $k(f,f')=\exp[-\gamma d(f,f')^2]$计算其相似度，其中 $d(f,f')$计算两个特征之间的欧几里得距离，$\gamma$ 为归一化参数，$\gamma=1/(2\bar{d})$($\bar{d}$ 为特征之间欧几里得距离的均值)。最后将五种特征计算得到的相似度矩阵相乘作为轨迹特征的相似度矩阵 $\boldsymbol{W}$。

在实现二分划分的过程中，原来方法显现出两个问题需要解决：一个是孤立轨迹特征点带来的；另一个是不同视频样本时长差距带来的。

#### 1. 孤立轨迹特征点

为了进行二分划分，需要在轨迹特征的时空中建立一个图连接结构。图中的节点表

示轨迹特征，若两个轨迹特征属于 $K$-近邻，则对应的节点之间就有连接边。此时可以使用 $K$-D 树方法快速建立轨迹的 $K$-近邻图结构。近邻数 $K$ 设置为 10 就可以使视频中的轨迹特征形成全连接的图结构。在对一个节点进行划分时，可以根据轨迹特征的连通分量数目进行判断，优先选取划分后连通分量数目少的划分方式。但是，在不断划分的过程中，节点里面还包含很多个孤立轨迹特征点，导致连通分量数目增多，影响实际的聚类效果。为了解决这个问题，可以考虑仅保留每个节点内的最大连通分量来进行下一层划分，以避免孤立轨迹特征点对聚类的影响。

2. 视频样本时长差距

实际中的视频样本各异，即便是同一类型活动或行为的视频所对应的时长也可能差别很大，这会导致从中提取出的轨迹特征数量也有很大差别。为了解决这个问题，可以使用自适应的节点划分阈值。这里动态阈值可以采用多个百分位数（如 10%，20%，…，90%），依次对特征矢量进行阈值判断，确定节点所含最少和最多轨迹特征数目，避免划分过程中产生两个节点轨迹特征数目相差很大的情况。这样还可加快聚类的进程。

### 3.2.3 FV 编码树

在获得特征聚类树之后，需要对聚类树中每个节点包含的轨迹特征进行矢量表示，这里采用了**费歇尔矢量**编码方法[Sánchez 2013]。该编码方法使用了特征与聚类中心的一阶和二阶统计信息，比 $K$-均值最近邻编码方法具有更强的鉴别性，能够获得更好的分类性能。具体是先从训练样本中随机选取 256000 个轨迹特征来训练 GMM，GMM 的数目 $K$ 设置为 256。FV 编码对应的编码矢量长度为 $2DK$，其中 $D$ 是特征的维度。最后对 FV 编码矢量进行归一化。将 HOG、HOF、MBH 等特征归一化后的 FV 编码矢量连接起来作为节点所含轨迹特征最终的 FV 编码矢量。

在获得视频的 FV 编码树后，需要定义一个度量不同树状结构相似度的核函数。GraphHopper 核函数通过比较节点对之间的最短路径，对节点的相似度进行加权平均[Feragen 2013]：

$$K(G_1,G_2)=\sum_{n_1\in V_1}\sum_{n_2\in V_2}w(n_1,n_2)k(n_1,n_2) \tag{3.2.5}$$

式中：$w(n_1,n_2)$为统计节点 $n_1$ 和 $n_2$ 在最短路径中相同位置出现的次数。

实际中，因为在对轨迹特征进行二分划分的聚类过程中两个子节点之间是无序的，会导致采用无监督方式聚类得到的树结构不够稳定，所以直接利用树结构来度量相似度的效果很难保证。因此，可以考虑直接计算所有节点对之间的相似度作为核函数：

$$K_a(T_1,T_2)=w_r k(r_1,r_2)+\frac{1-w_r}{|V_1||V_2|}\sum_{n_1\in V_1}\sum_{n_2\in V_2}k(n_1,n_2) \tag{3.2.6}$$

式中：$r_1$、$r_2$ 分别表示树 $T_1$、$T_2$ 的根节点；$n_1$、$n_2$ 分别表示非根节点集合 $V_1$、$V_2$ 中的节点；$w_r$ 表示加权系数，当 $w_r=1$ 时，表示直接利用词袋模型进行行为识别。

式(3.2.6)利用了树中所有节点对的相似度进行计算，可以称为所有树节点对核函数。

上述核函数仅仅考虑了节点相似度，进一步还可以考虑边相似度，以完全利用聚类树的结构信息。可以将子节点和父节点的编码矢量连接在一起作为边的矢量表示。这样得到所有树边对的核函数(e 指示边)：

$$K_{\mathrm{e}}(T_1,T_2)=w_r k(r_1,r_2)+\frac{1-w_r}{|U_1||U_2|}\sum_{e_1\in U_1}\sum_{e_2\in U_2}k(e_1,e_2) \tag{3.2.7}$$

式中：$e_1$、$e_2$ 分别代表树 $T_1$、$T_2$ 中非根节点与其父节点连接形成的边；$U_1$、$U_2$ 分别代表两棵聚类树中边的集合。

另外，具有不同相似度的节点的聚类效果并不相同。如果对所有节点平等看待，可能会影响最后核函数的度量效果。为此，在计算核函数时仅考虑具有最大相似度的节点对，这样，所有树节点对核函数成为(上标(m)指示最大)

$$K_{\mathrm{a}}^{(\mathrm{m})}(T_1,T_2)=w_r k(r_1,r_2)+(1-w_r)\left[\frac{\sum_{n_1\in V_1}\max_{n_2\in V_2}\{k(n_1,n_2)\}}{|V_1|}+\frac{\sum_{n_2\in V_2}\max_{n_1\in V_1}\{k(n_1,n_2)\}}{|V_2|}\right] \tag{3.2.8}$$

相应地，所有树边对核函数成为(上标(m)指最大)

$$K_{\mathrm{e}}^{(\mathrm{m})}(T_1,T_2)=w_r k(r_1,r_2)+(1-w_r)\left[\frac{\sum_{e_1\in U_1}\max_{e_2\in U_2}\{k(e_1,e_2)\}}{|U_1|}+\frac{\sum_{e_2\in U_2}\max_{e_1\in U_1}\{k(e_1,e_2)\}}{|U_2|}\right] \tag{3.2.9}$$

最后，在计算节点相似度时还可以根据节点间的时空距离进行加权，得到加权的节点核函数($\gamma$ 是一个加权系数)：

$$K_{\mathrm{w}}(n_1,n_2)=\exp[-\gamma d(n_1,n_2)^2]\langle \mathrm{FV}(n_1),\mathrm{FV}(n_2)\rangle \tag{3.2.10}$$

前面介绍的 5 个核函数(式(3.2.5)～式(3.2.9))都是节点核函数 $K_{\mathrm{w}}(n_1,n_2)$ 的加权和。只要节点核函数满足正定条件，这 5 个核函数就满足正定条件，可以直接与 SVM 分类器进行结合。

### 3.2.4 实验结果和分析

该算法的性能借助数据库进行了实验验证和比较[Chen 2016]。

#### 1. 实验算法和数据库

在对算法的验证中，一方面考虑了原始的二分划分聚类方法[Gaidon 2014]，称为原

始算法；另一方面，分别考虑了对节点内属于最大连通分量的特征进行划分的方式(称为算法 1)和采用动态阈值作为判断划分条件的方式(称为算法 2)。上面介绍的借助轨迹特征聚类树的算法称为算法(1+2)。

实验中采用了 4 个通用视频数据库，分别为 Hollywood2 数据库[Marszalek 2009]、Olympic Sports 数据库[Niebles 2010]、HMDB51 数据库[Kuehne 2011]和 UCF50 数据库[Reddy 2013]，它们的一些概括情况如表 3.2.1 所示。

**表 3.2.1 4 个数据库的概况**

| 统 计 项 目 | Hollywood2 | Olympic Sports | HMDB51 | UCF50 |
|---|---|---|---|---|
| 行为类别总数 | 12 | 16 | 51 | 50 |
| 视频样本总数 | 1707 | 783 | 6766 | 6618 |
| 测试集样本数 | 823 | 134 | 1530 | 250 |

### 2. 算法效果比较

前述 4 种算法在视频时长相差较大的 Olympic Sports 数据库中进行实验得到的结果如表 3.2.2 所示，评价指标为**平均精确度均值**(mAP)。

**表 3.2.2 4 种算法在 Olympic Sports 数据库的效果**

| 聚 类 算 法 | 平均节点数 | mAP/% |
|---|---|---|
| 原始算法 | 519 | 91.5 |
| 算法 1 | 202 | 91.6 |
| 算法 2 | 34 | 91.9 |
| 算法(1+2) | 29 | 92.8 |

由表 3.2.2 可见，算法 1 和算法 2 都减少了平均节点数，也就是降低了孤立轨迹特征点的影响，并且还可以加速聚类树的计算；将它们结合的算法(1+2)效果更好。

### 3. 核函数比较

在式(3.2.10)中引入了时空距离加权的节点核函数。对于式(3.2.5)～式(3.2.9)所示的 5 个核函数，借助式(3.2.10)，还可以得到 5 个利用了时空距离加权的核函数。对这 10 个核函数的实验比较结果如表 3.2.3 所示(粗体表示最优)。

**表 3.2.3 不同核函数和节点核函数的比较** 单位：%

| 核 函 数 | 节点核函数 | Hollywood2 | Olympic Sports | HMDB51 | UCF50 |
|---|---|---|---|---|---|
| $K(G_1,G_2)$ | $K$ | 62.3 | 90.5 | 55.5 | 88.5 |
| | $K_w$ | 62.6 | 90.7 | 55.7 | 88.6 |

续表

| 核 函 数 | 节点核函数 | Hollywood2 | Olympic Sports | HMDB51 | UCF50 |
|---|---|---|---|---|---|
| $K_e(T_1,T_2)$ | $K$ | 64.1 | 92.4 | 59.1 | 91.4 |
| | $K_w$ | 64.3 | 92.5 | 59.2 | 91.4 |
| $K_e^{(m)}(T_1,T_2)$ | $K$ | 64.3 | 92.5 | 59.2 | 91.4 |
| | $K_w$ | 64.5 | 92.6 | 59.3 | 91.5 |
| $K_a(T_1,T_2)$ | $K$ | 64.0 | 91.9 | 58.9 | 91.3 |
| | $K_w$ | 64.1 | 92.1 | 59.0 | 91.4 |
| $K_a^{(m)}(T_1,T_2)$ | $K$ | 64.4 | 92.8 | 59.2 | 91.5 |
| | $K_w$ | **64.6** | **92.9** | **59.3** | **91.7** |

从表 3.2.3 中可以看出：

(1) 考虑了聚类树结构信息的核函数 $K(G_1,G_2)$ 的效果不如 $K_e(T_1,T_2)$ 和 $K_a(T_1,T_2)$，这说明聚类树的结构不够稳定，还不足以用来度量不同聚类树的差异。

(2) 仅考虑具有最大相似度节点对的核函数 $K_e^{(m)}(T_1,T_2)$ 比 $K_e(T_1,T_2)$ 效果好；同样，仅考虑具有最大相似度节点对的核函数 $K_a^{(m)}(T_1,T_2)$ 比 $K_a(T_1,T_2)$ 效果好。这说明不应当将所有节点同等对待。

(3) 核函数 $K_e^{(m)}(T_1,T_2)$ 与核函数 $K_n^{(m)}(T_1,T_2)$ 的效果比较接近，这说明对聚类树的边进行相似度量也是有效的。

(4) 借助对节点核函数进行时空距离加权所取得的效果有改善，但不太大。一个可能的因素是这里采用了节点内所有特征时空位置的平均值代表该节点的时空位置。由于一个节点可能包含大量的特征，使用平均值未能很好地反映节点内所有特征的分布情况。这还需要进一步探讨。

### 4. 时空划分比较

可以借助**时空金字塔**(STP)对时空进行划分。图 3.2.3 显示了 4 种将时空金字塔进行划分的方式，依次是未划分的时空方式(记为 O)、将时间划分为两部分的方式(记为 T2)、将空间水平划分为三部分的方式(记为 H3)、同时将时间划分为两部分并将空间水平划分为三部分的方式(记为 T2＋H3)。

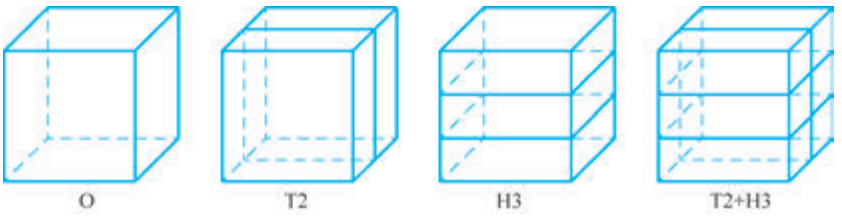


图 3.2.3 时空金字塔划分方式

对上述 4 种时空划分方式都进行了实验，得到的结果如表 3.2.4 所示。使用未划分的时空方式(O)在四个数据库上得到的结果都与原始算法[Gaidon 2014]基本一致。使用其他三种划分的时空方式(T2、H3、T2＋H3)，性能都有不同程度的下降，尤其是在由电影视频剪辑整理得到的 Hollywood2 数据库上性能下降最为严重。这里的一个解释是：原始算法使用的轨迹特征为**稠密轨迹特征**(DTF)，而算法 1、算法 2 和算法(1＋2)使用的轨迹特征均为**改进的轨迹特征**(ITF)。虽然在有摄像机运动时 ITF 特征比 DTF 特征更加鲁棒，但是在将 ITF 特征与时空划分方式相结合时每个划分区域中包含的特征数目相差很大，得到的 FV 编码矢量不能很好地描述该区域的表观和运动信息。这说明简单的时空划分方式并非对所有特征都有效。

**表 3.2.4 时空金字塔的影响** 单位：%

| 时空金字塔 | Hollywood2 | Olympic Sports | HMDB51 | UCF50 |
|---|---|---|---|---|
| O | 63.7 | 91.0 | 57.9 | 91.2 |
| T2 | 38.9 | 59.8 | 56.8 | 83.0 |
| H3 | 30.8 | 89.9 | 45.8 | 74.6 |
| T2＋H3 | 34.1 | 69.6 | 44.3 | 69.8 |

## 参考文献


[Chen 2016] Chen Q Q, Zhang Y-J. Cluster trees of improved trajectories for action recognition. Neurocomputing, 173: 364-372.

[Dalal 2005] Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proc. CVPR, V1: 886-893.

[Dalal 2006] Dalal N, Triggs B, Schmid C. Human detection using oriented histograms of flow and appearance. Proc. ECCV. 428-441.

[Farnebäck 2003] Farnebäck G. Two-frame motion estimation based on polynomial expansion. Proc. Scandinavian Conference on Image Analysis, 363-370.

[Feragen 2013] Feragen A, Kasenburg N, Petersen J, et al. Scalable kernels for graphs with continuous attributes. Proc. Advances in Neural Information Processing Systems, 216-224.

[Gaidon 2014] Gaidon A, Harchaoui Z, Schmid C. Activity representation with motion hierarchies. International Journal of Computer Vision, 107(3): 219-238.

[Kuehne 2011] Kuehne H, Jhuang H, Garrote E, et al. HMDB: A large video database for human motion recognition. Proc. ICCV, 2556-2563.

[Laptev 2008] Laptev I, Marszalek M, Schmid C, et al. Learning realistic human actions from movies. Proc. CVPR, 1-8.

[Marszalek 2009] Marszalek M, Laptev I, Schmid C. Actions in context. Proc. CVPR, 2929-2936.

[Morris 2008] Morris B T, Trivedi M M. A survey of vision-based trajectory learning and analysis for surveillance. IEEE-CSVT, 18(8): 1114-1127.

[Niebles 2010] Niebles J C, Chen C W, Fei-Fei L. Modeling temporal structure of decomposable motion segments for activity classification. Proc. ECCV, 392-405.

[Reddy 2013] Reddy K K, Shah M. Recognizing 50 human action categories of web videos. Machine Vision and Applications, 24(5): 971-981.

[Sánchez 2013] Sánchez J, Perronnin F, Mensink T, et al. Image classification with the Fisher vector: Theory and practice. International Journal of Computer Vision, 105(3): 222-245.

[Shi 1994] Shi J, Tomasi C. Good features to track. Proc. CVPR, 593-600.

[Wang 2013] Wang H, Schmid C. Action recognition with improved trajectories. Proc. ICCV, 3551-3558.

[贾 2007] 贾慧星, 章毓晋. 车辆辅助驾驶系统中基于计算机视觉的行人检测研究综述. 自动化学报, 33(1): 84-90.

[孙 2024] 孙瑾, 杜官明. 多目标跟踪中基于次模优化的轨迹片段生成方法. 电子与信息学报, 46(3): 995-1004.

[王 2024] 王汝言, 周玉蝶, 吴大鹏, 等. 群组感知的行人轨迹预测方法研究. 通信学报, 45(12): 44-56.

[杨 2024] 杨超群, 徐梦蝶, 梁潇洧, 等. 基于随机有限集滤波器的可分辨群目标跟踪技术研究综述. 信号处理, 40(10): 1763-1772.

[张 2024] 张鹏, 雷为民, 赵新蕾, 等. 跨摄像头多目标跟踪方法综述. 计算机学报, 47(2): 287-309.

[章 2024] 章毓晋. 图像工程(下册): 图像理解(第 5 版). 北京: 清华大学出版社.

[赵 2024] 赵文红, 王巍, 万子璐. 基于时空 Transformer 特征融合的车辆轨迹预测. 通信学报, 45(11): 267-276.