

基于语义轨迹的视频事件探测

王晓峰^{1),2)} 张大鹏^{1),2),3)} 王 缙⁴⁾ 史忠植¹⁾

¹⁾(中国科学院计算技术研究所智能信息处理重点实验室 北京 100190)

²⁾(中国科学院研究生院 北京 100049)

³⁾(燕山大学信息科学与工程学院 河北 秦皇岛 066004)

⁴⁾(中国科学技术大学计算机科学与技术学院 合肥 230026)

摘 要 视频事件探测是视频内容自动理解领域的一个重要研究问题. 在视频事件探测中, 感兴趣对象的运动轨迹常被作为视频中探测事件的一种重要依据. 目前基于轨迹的事件探测方法主要集中于根据轨迹几何特征进行视频事件探测, 而忽略了与轨迹相关的语义信息. 然而我们知道, 轨迹的产生往往受到一些与轨迹相关联的语义信息的影响, 如轨迹产生时的地理信息等. 将轨迹相关联的语义信息整合到轨迹中可以使我们了解更多关于轨迹的信息. 语义轨迹为我们提供了一个将语义信息与轨迹信息有效整合的方法. 该文将语义轨迹应用到视频事件探测领域, 提出了一个基于语义轨迹的视频事件探测方法. 该方法将视频中抽取的感兴趣对象的原始轨迹转化为语义轨迹, 并根据语义轨迹探测可能的视频事件. 同时该方法还提供了一个描述语义轨迹特征以及对语义轨迹与轨迹特征进行匹配的方法. 最后我们通过实验分析验证了基于语义轨迹的视频事件探测方法的有效性.

关键词 视频事件; 事件探测; 语义轨迹; 本体; 推理

中图法分类号 TP391 **DOI号**: 10.3724/SP.J.1016.2010.01845

Semantic Trajectory Based Video Event Detection

WANG Xiao-Feng^{1),2)} ZHANG Da-Peng^{1),2),3)} WANG Fei⁴⁾ SHI Zhong-Zhi¹⁾

¹⁾(The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

²⁾(Graduate University of Chinese Academy of Sciences, Beijing 100049)

³⁾(Institute of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei 066004)

⁴⁾(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026)

Abstract Video event detection is a key research problem for automatic video content understanding. Trajectories of object are the one important clue for video event detection. Currently, the most trajectory based event detection methods focus on the geometric characteristics of the trajectory and neglect the semantic information related to the trajectory. However, as we known the trajectory producing process is affected by the trajectory related semantic information, such as geographic information related to trajectory etc. Combine the semantic information related to the trajectory with raw trajectory information can make us obtain more knowledge of trajectory. Semantic trajectory provides a way to effectively combine trajectory information with semantic knowledge. In this paper, the semantic trajectory is applied into video event detection and proposed a semantic trajectory based method for video event detection. This method can transform the trajectory of the interested object in video to semantic trajectory and detect the video event based on the semantic trajectory. Moreover, this method provides an approach for depicting se-

收稿日期: 2010-08-22. 本课题得到国家自然科学基金(60775035, 61035003, 60933004, 60903141, 61072085)、国家“九七三”重点基础研究发展规划项目基金(2007CB311004)、国家科技支撑计划(2006BAC08B06)资助. 王晓峰, 男, 1978年生, 博士研究生, 主要研究方向为语义 Web、数据挖掘、智能规划理论与认知科学等. E-mail: wangxf@ics.ict.ac.cn. 张大鹏, 男, 1979年生, 博士研究生, 主要研究方向为人工智能理论. 王 缙, 男, 1987年生, 主要研究方向为图像处理、模式识别与数据挖掘. 史忠植, 男, 1941年生, 研究员, 博士生导师, 主要研究领域为人工智能、机器学习、多主体系统与认知科学.

semantic characters of semantic trajectory and matching the trajectory with trajectory characteristic description. Finally, the authors demonstrate the effectiveness of the proposed method through the empirical studies.

Keywords video event; event detection; semantic trajectory; ontology; reasoning

1 引 言

随着信息技术的迅猛发展,数字视频成为一种重要的信息资源.由于数字视频资源数量巨大且还在不断增加,因此对于数字视频的智能检索成为一个迫切需要解决的问题.所谓对于数字视频的智能检索是指从高层语义的角度对数字视频进行检索.而视频检索中的高层语义是指视频中出现的一些事件,也即人们看到视频后所联想到的一些概念,如飞机起飞、行人行走等.自动探测视频中的事件是实现智能视频检索的基础^[1-3].但由于视频是非结构化信息,在低层视觉特征上(颜色、纹理、形状)相近的视频从高层语义角度看可能相去甚远.因此视频中的事件探测是一个极富挑战性的研究课题.

一般地,根据视频信息的内容结构可将视频信息划分为两类:

(1) 场景相对固定的视频(constraint video),如监控视频、足球、篮球的体育比赛视频^[4-5];

(2) 场景频繁变化的视频(unconstraint video),如新闻视频、电影视频等^[1-3].

针对不同类型的视频需要采用不同的方法进行事件探测.对于场景相对固定的视频一般运用以对象为中心(object-centered)的方法进行视频事件探测.以对象为中心的方法将视频事件视为视频中对象的特定动作行为(如射门动作),或者某种时空关系变化(如进入商店购物).一般以对象为中心的事件探测方法通过对象识别、跟踪技术以及动作识别技术获取视频中对象的行为信息,继而根据对象的行为信息进行特定事件的探测、识别.相比较场景固定的视频,场景频繁变化的视频中包含了大量的物体、复杂的动作,不同角度、场景拍摄视频的频繁切换.因此场景频繁变化的视频中的事件探测较为复杂,不宜采用以对象为中心的方法进行事件探测.文献[1]提出了一个将视频内容投影到多维概念空间的方法来实现对于场景频繁变化视频的事件探测.文献[2]在此基础上提出一个基于多层次时序对齐核函数(multilevel temporal alignment kernel),并

基于该核函数实现场景频繁变化视频中的事件探测.本文研究的是基于轨迹的视频事件探测技术,因此我们将重点探讨在场景相对固定的视频中的事件探测.

视频中物体的运动轨迹一定程度反映了物体的行为目的.具有相同或相近行为目的的对象往往具有相似运动轨迹,例如公路上的汽车的运行轨迹通常是很相似的.此外通过视频跟踪技术可以较容易地获取视频中物体的运动轨迹信息.因此在对场景相对固定的视频进行事件探测时,物体运动轨迹常被作为探测事件的重要依据^[4-5].一般可将轨迹表示为如下形式的序列:

$$Trajectory(o) = \{(x_0, y_0, t_0), (x_1, y_1, t_1), \dots, (x_n, y_n, t_n)\}, t_0 < t_1 < \dots < t_n,$$

其中 o 表示产生轨迹 $Trajectory(o)$ 的对象,轨迹 $Trajectory(o)$ 中的任意元组 (x_i, y_i, t_i) 表示在时间点 t_i 对象 o 的空间位置关系为 (x_i, y_i) . x_i 为水平坐标, y_i 为垂直坐标值.为了和语义轨迹相区别,我们将 $Trajectory(o)$ 称为原始轨迹(raw trajectory).

目前主要的基于轨迹的事件探测方法是基于原始轨迹的视频事件探测方法,该方法可以被划分为两个主要步骤:对象的轨迹抽取和对象轨迹的学习.其中对象的轨迹抽取又包括了对象探测和对象跟踪.而对象轨迹学习包括了轨迹聚类、聚类建模以及事件识别.

轨迹是对象移动行为的记录.对于轨迹所记录的对象移动行为的解释往往需要参考轨迹产生时的语境信息(context information).例如,在公路上发现许多移动缓慢的轨迹,那么如果这些轨迹是汽车的轨迹,那么该地点可能有红绿灯,或者正在堵车.由于原始轨迹只具备随时间而变化的空间位置信息,所以一般基于原始轨迹的事件探测方法仅通过轨迹几何特征来探测事件.而忽略了轨迹本身所具有的语义特征信息.因此基于原始轨迹的探测方法不能很好地探测一些拥有在几何特征上差异较大而在语义特征上较为相似的轨迹的事件.

例 1. 图 1 是关于一个商店的监控视频.在该视频中分别包含了 3 个行人的运动轨迹 t_1, t_2, t_3 . 由

于轨迹 t_2, t_3 在几何特征上较为相似, 因此通过基于轨迹的几何特征的聚类方法 t_2, t_3 被划分为一类, 而把 t_1 单独作为一类. 然而从语义特征角度看, t_1, t_2 为进入商店的轨迹, 而 t_3 为经过商店的轨迹, 因此从语义特征相似性的角度看 t_1, t_2 应该划分为一类, 而 t_3 单独为一类.



图 1 行人轨迹示例

文献[6-7]提出了将轨迹与地理本体相结合构造语义轨迹的方法, 并将语义轨迹应用于对 GPS 跟踪数据的查询、分析过程. 与原始轨迹相比, 语义轨迹显示地描述了轨迹中所包含的语义地理信息. 例如原始轨迹 $T(o) = \{(x_0, y_0, t_0), (x_1, y_1, t_1), \dots\}$ 通过结合地理本体分析得到语义轨迹 $S_T(o) = \{Begin, BusStation, Museum, \dots, End\}$. 语义轨迹 $S_T(o)$ 表示了对象 o 从车站乘车到了博物馆. 语义轨迹为我们提供了一个轨迹的高层语义视图, 使我们可以从高层语义的角度处理轨迹信息.

语义轨迹非常适合于表示视频中对几何特征不敏感而对语义特征敏感的轨迹, 如逛商店的轨迹、行人停车场停车的轨迹等. 根据语义轨迹我们可以从高层语义层面构建关于视频事件的定义与探测机制. 然而原有的基于轨迹的事件探测方法仅适合于分析原始轨迹, 不适合于发现、表示语义轨迹以及定义和探测基于语义轨迹的视频事件. 所以需要构建一个基于语义轨迹的事件探测方法. 该探测方法能从视频中抽取感兴趣对象的语义轨迹, 并根据对象语义轨迹探测相关视频事件.

本文提出一个基于语义轨迹的视频事件表示探测方法. 该方法根据改进的 CamShift 方法从视频中抽取运动对象的原始轨迹, 而后结合由本体定义的兴趣区域生成原始轨迹对应的语义轨迹. 同时, 该方法提供了一个基于正则式的轨迹特征描述方法, 并基于该方法构建了基于语义轨迹的视频事件定义方法. 最后通过基于正则式的包含关系判定方法根

据语义轨迹探测视频中的事件. 通过实验分析, 我们验证了该方法能有效发现视频中的语义事件.

综上所述, 本文有如下几个重要贡献:

(1) 提出了一个基于语义轨迹的事件探测框架, 实现了基于轨迹的语义特征的事件探测;

(2) 提出了一种根据视频中原始轨迹生成语义轨迹的方法;

(3) 提出了基于正则式的轨迹特征描述方法以及基于正则式包含关系的视频事件匹配方法;

本文第 2 节介绍语义轨迹相关概念; 第 3 节为基于语义轨迹的事件探测框架; 第 4 节为实验分析; 第 5 节为相关工作; 第 6 节为全文总结.

2 语义轨迹

语义轨迹由 Move 和 Stop 组成, 它们的定义如下.

定义 1(Stop). 一个 Stop S 由元组 $S = (R_c(S), \delta_{\min}(S), \delta_{\max}(S))$ 构成. $R_c(S)$ 表示一个平面区域, $\delta_{\min}(S), \delta_{\max}(S)$ 表示时间阈值. $R_c(S)$ 称为 S 的空间区域, $\delta_{\min}(S)$ 称为 S 的最小时间阈值, $\delta_{\max}(S)$ 称为 S 的最大时间阈值.

Stop 将对象在特定区域的持续运动轨迹或者停留作为基本元素加入到语义轨迹中. 根据 Stop 的区域以及时间阈值可以从原始轨迹中发现 Stop 的实例. 假设存在一个轨迹 T , 轨迹 $T' = \{(x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1}), \dots, (x_{i+k}, y_{i+k}, t_{i+k})\}$ 为其子轨迹, 对于某个 Stop $S' = (R_c(S'), \delta_i(S'))$, 如果 T' 满足: ① $\forall (x, y, t) \in T', \text{有 } (x, y) \in R_c(S')$; ② $\delta_{\min}(S) \leq t_{i+k} - t_i \leq \delta_{\max}(S)$; ③ 不存在其它包含轨迹 T' 的 T 子轨迹满足条件①、②, 那么则称 T' 对应一个 Stop S' 的一个实例.

定义 2(Move). Move 是语义轨迹中连接 Stop 实例的元素的统称. 一个 Move 的实例可由 $M = (s_i, s_{i+1})$ 表示, 其中 s_i, s_{i+1} 表示 M 所联接的 Stop 实例.

Move 将语义轨迹中的 Stop 实例联接起来, 形成一个由 Move 和 Stop 的实例组成的有序序列. 本文中, 我们用本体来描述及表示语义轨迹. 其中不同类型的 Stop 和 Move 用本体里的类(Class)来表示. 组成实际语义轨迹的元素用 Stop 实例和 Move 关系来表示.

定义 3(语义轨迹). 一条语义轨迹是由一条由 Move 关系和 Stop 的实例组成的有序序列. 其中每个 Move 关系连接两个 Stop 实例. 语义轨迹中除

第一个以及最后一个 Stop 实例外, 每个 Stop 实例都由两个 Move 关系相连接; 序列中的第一个以及最后一个 Stop 实例仅由一个 Move 关系所连接.

给定一个原始轨迹 T 以及一个 Stop 类集合 $Set_{Stop} = Stops \cup \{Stop_{begin}, Stop_{end}\}$, 可以按照如下规则生成 T 对应的语义轨迹 $S_T^{[6-7]}$:

(1) 分别生成 $Stop_{begin}, Stop_{end}$ 的一个实例, 其中 $Stop_{begin}$ 的实例作为语义轨迹的第一个 Stop 实例, $Stop_{end}$ 的实例作为语义轨迹的最后一个 Stop 实例;

(2) 根据 $Stops$ 中的 Stop 定义以及轨迹 T 生成有序的 Stop 实例序列;

(3) 用 Move 关系连接序列中的每对相邻的 Stop 序列;

Stop 集合 $Set_{Stop} = Stops \cup \{Stop_{begin}, Stop_{end}\}$ 中的 $Stops$ 表示用户定义的 Stop 集合. $Stop_{begin}$ 和 $Stop_{end}$ 表示语义轨迹 S_T 的起始 Stop 和结束 Stop. $Stop_{begin}, Stop_{end}$ 分别记录了轨迹开始和结束的空间位置.

例 2. 给定用户定义的商店 $Stop_{shop} = (R_{shop}, \delta_{min_shop}, \delta_{max_shop})$, 那么 $Shopping_Trajectory = \{Instance(begin), Instance(shop), Instance(end)\}$ 表示一条逛商店的语义轨迹, 如图 2 所示. 在 $Shopping_Trajectory$ 中, $Instance(begin)$ 是 $Stop_{begin}$ 的实例, $Instance(shop)$ 是 $Stop_{shop}$ 的实例, 而 $Instance(end)$ 是 $Stop_{end}$ 的实例.

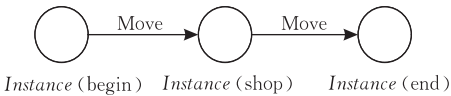


图 2 语义轨迹结构示例

3 基于语义轨迹的事件探测

如图 3 所示, 本文提出的基于语义轨迹的事件探测方法包括了底层轨迹抽取、语义轨迹生成以及

事件探测 3 个部分. 其中低层运动轨迹抽取部分采用背景差分和 CamShift 算法实现对视频中对象的检测、跟踪及轨迹抽取; 语义轨迹生成部分通过感兴趣区域 (Region of Interests, ROI)、本体知识将低层的运动轨迹信息转化为基于本体表示的语义轨迹; 事件探测部分通过基于本体的推理机制实现对于视频中特定事件的探测. 下面我们将对视频事件探测框架中的各个部分进行详细讨论.

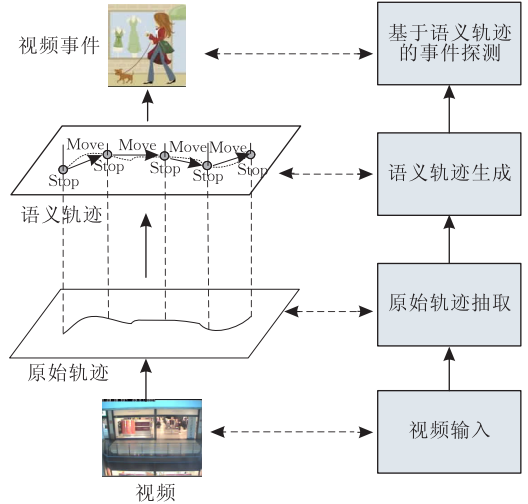


图 3 基于语义轨迹的视频事件探测框架

3.1 原始轨迹抽取

原始轨迹抽取部分负责发现视频中感兴趣的移动物体, 跟踪并记录下物体的运动轨迹. 原始轨迹抽取主要包括了: ① 对象检测过程, 以从视频中发现运动对象; ② 对象类型识别过程, 以判断对象是否为感兴趣对象 (如汽车、人); ③ 对象跟踪过程, 以跟踪对象运动轨迹.

本文中, 我们利用差分法分离视频中的前景和背景 (图 4(a) 为原视频, 图 4(b) 为差分后的结果), 并采用了向下采样、竖直方向膨胀、竖直方向腐蚀、向上采样 4 个基本的图像去噪技术, 实现去除噪声及对象强化, 将分离的对象各个部分连接起来, 形成整体, 最终效果如图 4(c) 所示.

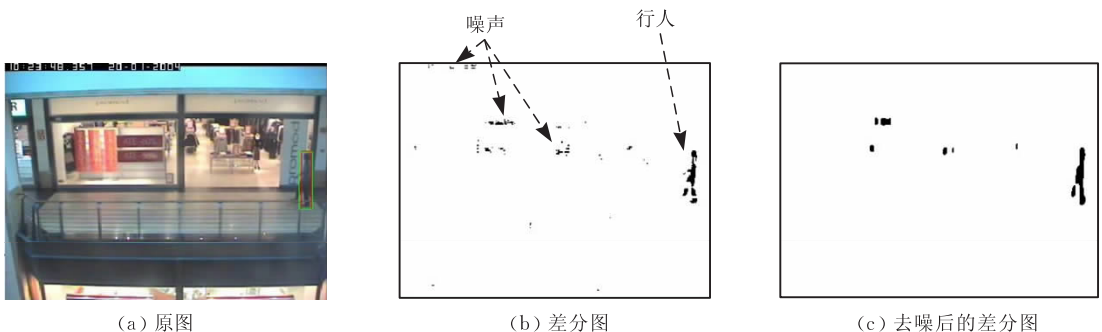


图 4 背景差分和去噪示例

我们在去噪的差分图像上构建前景对象矩形外围的轮廓. 而后对每个矩形轮廓内的对象用识别对象的二值分类器进行分类, 例如用判别人或者判别汽车的二值分类器对于矩形轮廓内的对象进行分类. 识别特定对象的二值分类器根据视觉特征判定某个区域是否为某个对象, 如果是则输出 1, 否则输出 0. 本文中, 我们采用 OpenCV 提供的库函数构建基于 Haar 特征的识别对象类型的二值分类器.

Bradski^[8]等人最早提出 CamShift 运动物体跟踪算法, 该算法建立在 MeanShift^[9]算法基础之上, 是一种基于运动物体颜色信息进行跟踪的算法. 我们采用了 Cam-shift 算法作为跟踪对象轨迹的方法. 图 5 为利用 CamShift 得到的一个行人轨迹.



图 5 由 CamShift 生成的物体运动轨迹

经过 CamShift 算法提取出的行人运动轨迹存在一定的抖动, 因此需要对轨迹做平滑处理. 设三维的点序列 $T = \{(x_0, y_0, t_0), (x_1, y_1, t_1), \dots, (x_n, y_n, t_n)\}$ 为一个行人的运动轨迹, 其中 (x_i, y_i) ($0 \leq i \leq n$) 为行人空间二维坐标, t_i 为一维时间坐标. 那么可以通过移动平均的方法来实现对轨迹 T 的平滑. 图 6 为轨迹经平滑后所得结果. 由该图可以看出经过平滑后的轨迹明显减少了波动.



图 6 平滑后的物体运动轨迹

3.2 语义轨迹生成

根据语义轨迹的定义可知, 语义轨迹由 Move

和 Stop 的实例组成. 由于语义轨迹中的 Move 实例只是用于表示 Stop 实例之间的时序关系, 因此可以将一条语义轨迹等价为一个由 Stop 实例组成的有序序列. 而根据原始轨迹生成语义轨迹的过程可以转化为根据原始轨迹生成 Stop 实例有序序列的过程.

3.2.1 定义 Stop

视频中的 Stop 概念是基于感兴趣区域定义的, 因此要定义 Stop 概念, 首先要确定视频中的感兴趣区域. 视频中的感兴趣区域是指视频中具有特定语义含义的空间区域, 比如提款机附近区域、停车泊位、地铁乘车口等. 一般运动对象在感兴趣区域的停留与某些视频事件存在一定的语义关联. 比如, 视频中对象在提款机附近的停留可能发生提款事件; 汽车停留在泊位可能发生汽车停车事件. 视频中的感兴趣区域可以通过分析轨迹的几何特征来自动发现, 也可以通过专家的先验知识来预先定义一些视频中的感兴趣区域. 由于先验知识定义的感兴趣区域较为准确, 且能对感兴趣区域中的 Stop 定义提供很好的语义解释, 因此本文中讨论了感兴趣区域仅限于通过先验知识预定义的感兴趣区域. 此外, 为了充分利用现有的语义 Web 的表示及推理方法, 我们针对监控视频构建领域本体, 并用领域本体来定义和描述感兴趣区域.

为了详细说明定义视频中的 Stop 概念以及根据 Stop 概念集生成语义轨迹的过程, 我们以商店的监控视频为例讨论如何划分感兴趣区域, 如何根据感兴趣区域定义 Stop 概念以及如何根据原始轨迹生成语义轨迹.

例 3. 图 7 为商场监控视频中的感兴趣区域划分. 这些感兴趣区域分别为商场内部(Shop)、商场玻璃柜台(ShowCase)、商场玻璃柜台外的走道(ShowCaseWalkWay)、商场大门外的走道(DoorWalkWay)、商场走廊入口(Entrance)、商场走廊出口(Exit). 我们根据商场监控视频构建了一个本体

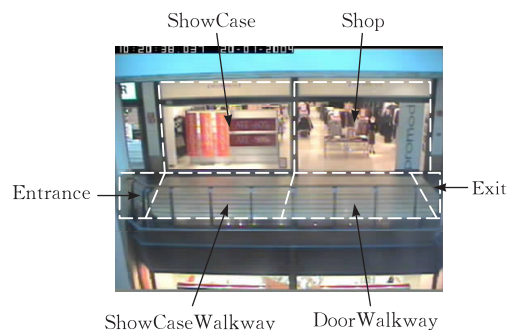


图 7 商场感兴趣区域

Ont4Shop,并用该本体表示商场监控视频中的感兴趣区域.图8为 Ont4Shop 中关于感兴趣区域部分的类关系图.在 Ont4Shop 中,商场监控视频中的每个感兴趣区域被表示为的一个本体类.

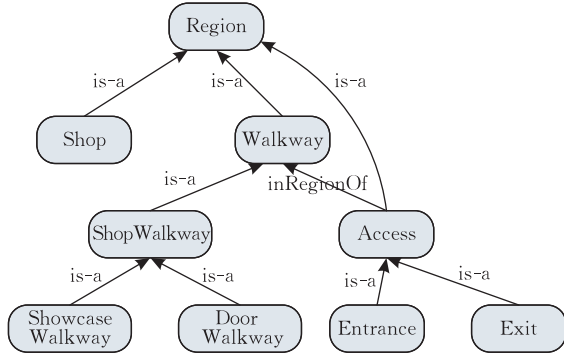


图8 Ont4Shop

在定义视频中感兴趣区域之后就可以基于感兴趣区域定义视频中的Stop概念集合.在 Ont4Shop

$PrimitiveStop \sqsubseteq Region \sqcap \exists HasTime.Thing$
 $ShopAroundStop \equiv PrimitiveStop \sqcap \exists InRegionOf.Shop \sqcap \exists HasTime.ShopAroundInterval$
 $OutOfShopShoppingStop \equiv PrimitiveStop \sqcap \exists InRegionOf.ShopWalkway$
 $InShopShoppingStop \equiv PrimitiveStop \sqcap \exists InRegionOf.Shop \sqcap \exists HasTime.PassShopInterval$
 $ShoppingStop \equiv ShoppingStopOutOfShop \sqcup ShoppingStopInShop$

图9 Stop 定义

根据本体中 Stop 的定义我们仅能知道 Stop 和感兴趣区域之间的关系,但不能确定 Stop 与原始轨迹的关系.由定义1可知,原始轨迹中满足在给定时间阈值,分布在某个给定空间区域的最长子轨迹就是 Stop 的一个实例.所以可以通过对 Stop 类定义其空间分布限定 R 以及时间阈值 δ 来将原始轨迹与 Stop 类关联起来.即通过 R, δ 发现原始轨迹中的 Stop 实例.为了简化讨论,我们假定所有 Stop 类中的空间区域限定是相同的,该区域限定在构建本体时给定.而时间阈值根据本体定义中的不同类型的 Interval 来表示.本体中不同类型的 Interval 的时间阈值则是在构建本体时确定.此外,为了区分同一个感兴趣区域的 Stop,我们限定相同感兴趣区域中的 Stop 在时间限定的交集为空.即任意两个在同一感兴趣区域里的 Stop $S = (R(S), \delta_{\min}(S), \delta_{\max}(S))$ 和 $S' = (R(S'), \delta_{\min}(S'), \delta_{\max}(S'))$, 必然有 $\delta_{\max}(S) < \delta_{\min}(S')$ 或者 $\delta_{\max}(S') < \delta_{\min}(S)$ 成立.

3.2.2 构建语义轨迹

根据定义 Stop 类集合就可以将一个原始轨迹转化为一个 Stop 实例组成的有序序列,进而生成原始轨迹对应的语义轨迹.需要指出的是,在视频跟踪

中我们定义了5种类型的 Stop: PrimitiveStop、ShopAroundStop、OutOfShopShoppingStop、InShopShoppingStop 以及 ShoppingStop.图9为5种类型的 Stop 所对应的描述逻辑定义.

该描述逻辑定义的对应该语义为

PrimitiveStop. 一个原子概念,它为 Region 的一个子类,同时它还有 HasTime 关系,表明 Stop 需要有时间限定;

ShopAroundStop. 它表示行人表现出对商场中的商品的兴趣且在商场内逗留挑选商品.行人在商场内有较长时间的停留;

OutOfShopShoppingStop. 它表示行人在商场外做了一定时间的停留观察商场内部的物品;

InShopShoppingStop. 它表示行人进入商场,但很快就离开;

ShoppingStop. 它表示 ShoppingStopInShop 和 ShoppingStopOutOfShop 并集.

时形成的轨迹为连续的,因此我们需要将连续轨迹离散化,以便于生成语义轨迹.本文中,我们通过采用固定频率的采样对连续轨迹离散化.即每个固定时间间隔在连续轨迹上采用一个样本点.因此采样生成的离散原始轨迹的连线点之间的时间间隔是相同的.此外为了保证所有 Stop 的实例都是可发现的,我们限定所有 Stop 的最小时间阈值均大于对连续轨迹进行采样的时间间隔.

对于同一个感兴趣区域 I 中的 Stop $S_i = (R(S_i), \delta_{\min}(S_i), \delta_{\max}(S_i))$ 和 $S_j = (R(S_j), \delta_{\min}(S_j), \delta_{\max}(S_j))$, 如果满足 $R(S_i) \subseteq R(S_j)$ 且 $\delta_{\max}(S_i) < \delta_{\min}(S_j)$, 那么分布于 I 的原始轨迹片段中,可能会同时发现 Stop S_i 和 S_j 实例.例如,在商场监控视频中发现 ShopAroundStop 实例的轨迹片段中可能会发现 InShopShoppingStop 实例(如图10所示).显然该轨迹片段对应的是 ShopAroundStop 实例而非 InShopShoppingStop 实例.为了避免在一个轨迹片段中重叠识别出多个 Stop 实例,我们限定一个满足空间约束的轨迹片段 e 至多对应一个 Stop $S = (R(S), \delta_{\min}(S), \delta_{\max}(S))$ 的实例.其中 S 满足条件 $\delta_{\min}(S) \leq |t_{\begin{smallmatrix} begin \\ end \end{smallmatrix}} - t_{\end{smallmatrix}}| \leq \delta_{\max}(S)$. 这里, $t_{\begin{smallmatrix} begin \\ end \end{smallmatrix}}$ 分别

表示轨迹片段 e 的第一个轨迹点的时间, 和 e 的最后一个轨迹点的时间. 由于之前限定了同一感兴趣区域的 Stop 在时间约束上没有交集, 所以对于任意在某感兴趣区域的轨迹片段至多存在一个 Stop 满足条件 $\delta_{\min}(S) \leq |t_{\text{begin}} - t_{\text{end}}| \leq \delta_{\max}(S)$.

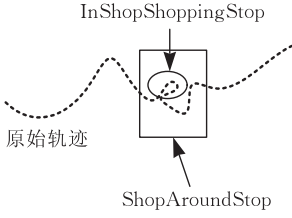


图 10 发现 Stop 实例示例图

为实现语义轨迹的构建, 本文提出一个从原始轨迹发现 Stop 实例序列贪心算法. 该算法的主要思想是, 首先根据原始轨迹在感兴趣区域中的分布情况将原始轨迹划分为若干轨迹子片段. 对于每个轨迹片段, 以该片段的第一个轨迹起始点, 按照轨迹点顺序逐步扩展, 直至扩展轨迹点分布超出 Stop 的空间区域限定. 结合感兴趣区域的 Stop 集合确定扩展轨迹点集合对应的 Stop 实例. 在分析扩展轨迹节点集合对应的 Stop 实例之后, 将扩展节点集合中的节点从轨迹片段中删除, 而后对轨迹片段中的剩余部分继续进行轨迹点扩展、扩展节点集合分析等步骤, 直至轨迹片段没有任何节点(所有节点都已被删除). 算法输出的结果为一个 Stop 实例序列.

算法 1 中的步 6 中的 $get_first(Queue_{ep})$ 函数取出队列 $Queue_{ep}$ 中的第一个元素并将其从 $Queue_{ep}$ 中删除. 步 9 中的 $first_node(e)$ 返回轨迹片段 e 的第一个轨迹节点(按时间顺序). 步 11 中的 $Region(nodeSet)$ 返回包括 $nodeSet$ 中所有节点的最小矩形区域. 步 11 用于判断包括 $nodeSet$ 中所有节点的最小矩形区域 $Region(nodeSet)$ 是否满足空间区域约束 R_{Δ} . 步 20 中的 $|t(n_i) - t(n_j)|$ 为轨迹点 n_j 与轨迹点 n_i 的时间差值. 假设原始轨迹的长度为 n (即原始轨迹包含 n 的轨迹点), 那么算法 1 的时间复杂度为 $O(n^2)$.

算法 1. 发现 Stop 实例序列算法.

Require: raw trajectory $T_{\text{raw}} = \{(x_0, y_0, t_0), (x_1, y_1, t_1), \dots, (x_n, y_n, t_n)\}$, Stop Region Restriction R_{Δ} ;

Ensure: return Stop Instance Sequence;

1. Segment the T_{raw} into episode sequence;
2. Put the episode sequence into a queue $Queue_{ep}$;
3. */* Initialize the Stop sequence Seq_{Stop} */*
4. $Seq_{\text{Stop}} \leftarrow \{\}$;
5. while $Queue_{ep}$ is not empty do

6. $e \leftarrow get_first(Queue_{ep})$;
7. while e is not empty do
8. $nodeSet \leftarrow \{\}$;
9. $n \leftarrow first_node(e)$;
10. Add n to $nodeSet$;
11. while $Region(nodeSet)$ satisfies R_{Δ} do
12. Remove n from e ;
13. $n \leftarrow first_node(e)$;
14. Add n to $nodeSet$;
15. end while
16. Remove n from $nodeSet$;
17. if $|nodeSet| > 1$ then
18. Let n_i be the first added node in $nodeSet$;
19. Let n_j be the last added node in $nodeSet$;
20. if $|t(n_i) - t(n_j)|$ satisfies the temporal restriction of Stop S then
21. Create an instance of S ;
22. Add the created instance to Seq_{Stop} ;
23. end if
24. end if
25. end while
26. end while
27. return Seq_{Stop} ;

通过算法 1 可以得到一个关于原始轨迹的 Stop 实例序列. 我们在 Stop 实例序列的第一个元素前插入一个特殊的 PrimitiveStop 实例 S_{begin} , 在 Stop 实例序列的最后一个元素后面插入特殊的 PrimitiveStop 实例 S_{end} , 而后对于任意两个相邻的 Stop 实例动用了一个 Move 关系连接, 就可以将 Stop 实例序列转化为语义轨迹.

例 4. 图 11 为根据算法 1 提取到的语义轨迹. 在原始轨迹上的点表示对应语义轨迹中的 Stop 实例节点.



图 11 Stop 实例节点

3.3 基于语义轨迹的事件探测

事件探测过程就是通过分析视频中对象的行为确定对象的行为是否是满足某个事件的定义. 事件探测包括了两个基本部分: ① 如何定义事件; ② 如何将观测到的对象行为与事件定义进行匹配. 如

图 12 所示,在基于语义轨迹的事件探测框架中,我们将语义事件定义为本体中的概念,同时根据视频中发现的语义轨迹生成包含语义轨迹的最小概念;然后通过推理引擎判断事件定义所对应的概念是否包含了语义轨迹所对应最小概念;如果包含,则认为视频中包含了相应的语义事件.下面我们将针对语义事件表示(定义)以及语义事件探测进行详细讨论.

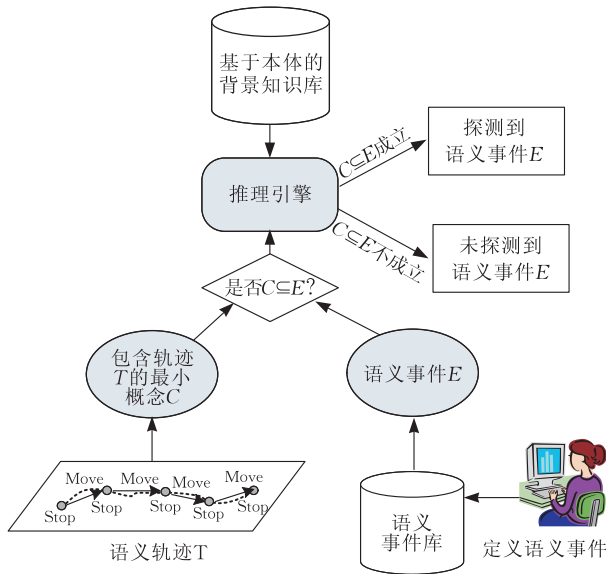


图 12 事件探测框架

3.3.1 基于语义轨迹的事件表示

如前面所讨论的,基于轨迹的事件探测是构建在同一事件中的对象运动轨迹都具有共同的轨迹特征这一假设基础之上的.基于原始轨迹的事件探测考虑的是轨迹的共同几何特征,而基于语义轨迹的事件探测则应该考虑的是轨迹的共同语义特征.基于上述考虑,我们将事件定义为本体中的概念(类),而将事件中的轨迹所共同具备的语义特征作为事件概念的约束.下面的描述逻辑定义式为基于语义轨迹的事件的一般形式:

$VideoEvent \equiv$

$Event \sqcap \exists hasTrajectory.someTypeTrajectory,$

其中 $Event$ 是所有事件类的父类. $\exists hasTrajectory.someTypeTrajectory$ 为对 $Event$ 的约束.该约束的对应语义是某种类型的事件必须具备属于 $someTypeTrajectory$ 类型的语义轨迹. $someTypeTrajectory$ 代表了某类具有共同语义特征语义轨迹集合,我们称之为语义轨迹类.可以通过如下的形式来定义语义轨迹类 $someTypeTrajectory$:

$someTypeTrajectory \equiv$

$Trajectory \sqcap trajectoryRestriction,$

其中 $Trajectory$ 为所有轨迹的父类,而 $trajectoryRestriction$ 为根据语义轨迹特征定义的约束. $someTypeTrajectory$ 定义式的语义含义为 $someTypeTrajectory$ 是满足 $trajectoryRestriction$ 约束的语义轨迹的集合.语义轨迹是序列型数据.然而目前的本体语言,如 OWL-DL,并不支持关于序列型数据集合的描述.因此,我们对本体语言 OWL-DL 进行了一定的扩展,使其可以支持对于序列型数据集合的描述、推理.

由于描述逻辑是本体的逻辑基础^[10],所以我们基于描述逻辑来讨论本体的描述能力以及如何扩展本体实现对于序列型数据集合的描述.描述逻辑(如 ALC)中包括了两种类型的集合描述:

(1) 论域中的个体集合,在描述逻辑中用概念描述;

(2) 论域中个体构成的二元组集合,在描述逻辑中用角色描述.

序列是论域中个体构成的多元组集合(二元组及更多元组),但目前的描述逻辑并不具备对于序列集合的描述能力,所以无法直接通过本体来描述语义轨迹集合.同时,我们注意到语义轨迹的 $Stop$ 实例序列是通过 $Move$ 角色(关系)确定的,即语义轨迹 $\{s_1, s_2, \dots, s_{n-1}, s_n\} \equiv \{Move(s_1, s_2), \dots, Move(s_{n-1}, s_n)\}$.因此我们在本体中引入基于角色的序列集合,以使本体支持由某种类型的角色确定的序列.下面是在 ALC 基础上定义的基于角色的序列集合的语法及其对应语义.

定义 4(基于角色的序列类). $Seq(R)$ 表示基于角色 R 的序列类, $Seq(R^n)$ 为基于角色 R 的 n 长度序列类,那么给定解释 $I = (\Delta^I, \cdot^I)$,非空集合 Δ^I 为论域, \cdot^I 为解释函数,且 \cdot^I 将个体概念映射为 Δ^I 的子集,将角色映射为 $\Delta^I \times \Delta^I$;将概念 $Seq(R^n)$ 映射为

$$Seq(R^n)^I = \{ \{s_1, s_2, \dots, s_{n-1}, s_n\} \in \Delta^I \times \Delta^I \cdots \Delta^I \times \Delta^I \mid (s_1, s_2) \in R^I \wedge \cdots \wedge (s_{n-1}, s_n) \in R^I \},$$

其中 n 为大于等于 2 的正整数;将概念 $Seq(R)$ 映射为

$$Seq(R)^I = Seq(R^2)^I \cup Seq(R^3)^I \cup \cdots.$$

定义 4. $Seq(R^n)^I$ 中的 $\Delta^I \times \Delta^I \cdots \Delta^I \times \Delta^I$ 包含 $|\{s_1, s_2, \dots, s_{n-1}, s_n\}|$ 个 Δ^I . $Seq(R^n)$ 表示由 R 角色确定的长度为 n 的序列集合. $Seq(R)^I$ 表示由 R 角色约定的长度大于等于 2 的序列集合.

除了具备序列型数据集合的描述能力,我们还

希望本体具备描述序列轨迹的特征的能力. 这样可以在语义轨迹类的定义中加入关于该类语义轨迹共用语义特征的约束. 众所周知对于序列的特征(模式)进行描述最为直接且有效的方式就是基于正则表达式形式的特征描述, 因此我们在本体中引入基于正则表达式描述的序列集合.

可通过如下的规则递归构建描述序列特征的正则表达式:

- ① $R_{RegEx} := C$;
- ② $R_{RegEx} := R \cdot R'$;
- ③ $R_{RegEx} := R | R'$;
- ④ $R_{RegEx} := R^*$;

其中 C 为本体中的任意非序列类(概念), R, R' 为描述序列集合的正则表达式. 描述序列特征的正则表达式的含义如下:

规则 $R_{RegEx} := C$ 的含义为: 满足 $R_{RegEx} := C$ 的序列中只有一个个体的, 该个体必须为类 C 的一个实例;

规则 $R_{RegEx} := R \cdot R'$ 的含义为: $Seq_S = \{s_0, s_1, \dots, s_n\}$ 为满足 $R_{RegEx} := R \cdot R'$ 的序列, 则必然存在 $s_i \in Seq_S$, 使得 Seq_S 的子序列 $\{s_0, \dots, s_i\}$ 满足约束 R , Seq_S 的子序列 $\{s_{i+1}, \dots, s_n\}$ 满足约束 R' ;

规则 $R_{RegEx} := R | R'$ 的含义为: 满足 $R_{RegEx} := R | R'$ 的序列必然满足约束 R 或者满足约束 R' ;

规则 $R_{RegEx} := R^*$ 的含义为: 满足 $R_{RegEx} := R^*$ 的序列, 必然满足 $\{R, R \cdot R, R \cdot R \cdot R, \dots\}$ 中的任意一个正则式.

根据描述序列特征的正则表达式, 可以定义满足正则表达式的序列类(集合)的语法及语义.

定义 5(基于正则表达式的序列类). $RegExClass(reg)$ 表示基于正则表达 reg 的序列类, 给定解释 $I = (\Delta^I, \cdot^I)$, 非空集合 Δ^I 为论域, \cdot^I 为解释函数, 那么 \cdot^I 将个体概念映射为 Δ^I 的子集, 将角色映射为 $\Delta^I \times \Delta^I$, 将 $RegEx(reg)$ 映射为

$$RegExClass(reg) = \{ \{s_1, s_2, \dots, s_{n-1}, s_n\} \in \Delta^I \times \Delta^I \cdots \Delta^I \times \Delta^I \mid \{s_1, s_2, \dots, s_{n-1}, s_n\} \text{ 满足正则表达式 } reg \}.$$

定义 5 中的 n 为任意大于 0 的正整数. 根据基于角色的序列类以及基于正则表达的序列类, 可以将一般形式的语义轨迹类定义重写如下:

$$someTypeTrajectory \equiv Trajectory \sqcap (Seq(Move) \sqcap RegExClass(reg)).$$

在语义轨迹类定义中 $Seq(Move)$ 表示轨迹类是由 $Move$ 关系确定的序列集合, $RegExClass(reg)$ 表示该轨迹类必须满足正则式 reg 所描述的序列特征. $(Seq(Move) \sqcap RegExClass(r))$ 对于与 $trajectoryRestriction$. 此外, 我们将 $Trajectory$ 定义为由 $Stop$ 实例组成的任意序列, 即 $Trajectory^I = \{ \{s_1, s_2, \dots, s_{n-1}, s_n\} \in \Delta^I \times \Delta^I \cdots \Delta^I \times \Delta^I \mid n = 1, 2, \dots \}$. 此外验证一个序列是否满足某个正则式, 可以通过将正则式转化为对应有限状态自动机, 再把序列作为输入串, 最后检测序列输入完后自动机是否达到一个接受状态.

例 5. 根据图 13 所示轨迹 T 可以构建一包含语义轨迹 T 语义轨迹类 $SemanticTrajectory(T) \equiv Trajectory \sqcap (Seq(Move) \sqcap RegExClass(r))$, 正则式 r 定义为 $Class(S_{begin}) \cdot Class(S_1) \cdot (PrimitiveStop)^* \cdot Class(S_2) \cdot Class(S_{begin})$, 其中 $Class(S_i)$ 表示 $Stop$ 实例 S_i 对应的本体类. r 的语义含义是轨迹第 1 个实例必须是 $Class(S_{begin})$ 的实例, 紧跟着的一个 $Stop$ 实例必须为 $Class(S_1)$ 的实例, 轨迹最后一个 $Stop$ 实例必须为 $Class(S_{begin})$ 的实例. 倒数第 2 个 $Stop$ 实例为 $Class(S_2)$ 的实例.



图 13 轨迹 T

3.3.2 基于语义轨迹的事件探测

基于语义轨迹的事件探测的主要思想是通过判断一条语义轨迹是否满足事件对应的语义轨迹的语义特征来确定视频事件是否发生. 对于任意语义轨迹 $S_{trajectory} = \{s_{begin}, s_1, s_2, \dots, s_{end}\}$, 可以用形如 $R = RegExClass(Class(s_{begin}) \cdot Class(s_1) \cdot Class(s_2) \cdots Class(s_{end}))$ 的正则式序列类描述该轨迹的语义特征, 其中 $Class(s)$ 表示包含实例 s 的最小 $Stop$ 类. 显然具有正则式序列类 R 约束的语义轨迹类 $Class(S_{trajectory}) \equiv Trajectory \sqcap (Seq(Move) \sqcap R)$ 为包含 $S_{trajectory}$ 的最小语义轨迹类. 因此我们可以通过判断语义轨迹类 $Class(S_{trajectory})$ 是否为某个语义事件所对应的语义轨迹 $someTypeTrajectory$ 的子类来确定一条语义轨迹是否具备某个事件所对应的语义特征.

要判定一个语义轨迹类是否包含另外一个语义轨迹类需要相应的推理技术支持, 然而目前的本体推理算法只能判定描述逻辑中的概念的包含关系推理, 不能被直接应用于语义轨迹类的包含关系推理.

因此我们提出了一个判定语义轨迹类包含关系的算法. 由语义轨迹的一般定义式可以看出, 判定两个语义轨迹类 $C_1 \equiv Trajectory \cap (Seq(Move) \cap R_1)$ 和 $C_2 \equiv Trajectory \cap (Seq(Move) \cap R_2)$ 是否满足包含关系 $C_1 \subseteq C_2$ 的问题与判定 $(Seq(Move) \cap R_1) \subseteq (Seq(Move) \cap R_2)$ 是否成立的问题是等价的. 因此我们仅需要判断 $R_1 \subseteq R_2$ 是否成立就能确定 $C_1 \subseteq C_2$ 是否成立. 由于 R_1, R_2 是基于正则式定义的序列类, 所以可以基于正则式包含关系的判定算法^[11] 构建判断 $R_1 \subseteq R_2$ 是否成立的算法.

在介绍语义轨迹类的包含关系推理方法前, 首先需要讨论一下根据正则式序列类 R 构建对应的自动机 $\mathcal{FA}(R)$.

定义 6. 正则式序列类 R 对应的确定有限状态自动机 $\mathcal{FA}(R)$ 为一个五元组 $(S, \Sigma, \delta, s_0, F)$, 其中:

- ① S 为有限状态集;
- ② Σ 为自动机的输入字符集, 这里的字符为 Stop 实例;
- ③ 状态转移函数 δ 为一个从 $S \times \Sigma$ 到 S 的单值映射;
- ④ $s_0 \in S$ 为自动机的初态;
- ⑤ $F \subseteq S$ 为接受状态集.

需要进一步说明的是 $\mathcal{FA}(R)$ 中的 δ 为形如 $\delta(s_i, Class_{Stop}) = s_j$ 的映射, 其中 $s_i \in S, Class_{Stop}$ 为 Stop 类 $\delta(s_i, Class_{Stop}) = s_j$ 的含义是: 如果当前状态为 s_i , 输入实例为 $Class_{Stop}$ 的实例时, 自动机状态转移到 s_j . 虽然输入字符集 Σ 是无限, 但 Stop 类是有限的, 而状态转移函数 δ 仅依赖于 Stop 实例所对应的 Stop 类, 所以状态转移函数的映射也是有限的.

引理 1. 对于任意正则式序列类 R 必然存在与其等价的确定有限状态自动机 $\mathcal{FA}(R)$.

证明. 证明过程与证明正则式必然存在等价有限状态机过程相似, 参看文献^[11].

引理 2. $\mathcal{FA}(R)$ 为与正则式序列类 R 等价的确定有限状态自动机, 则对于 $\mathcal{FA}(R)$ 的任意状态 s 以及状态转移函数 δ , 如果存在映射 $\delta(s, Stop_i) = s_j, \delta(s, Stop_j) = s_k, Stop_i \neq Stop_j \Rightarrow Stop_i \not\subseteq Stop_j \wedge Stop_j \not\subseteq Stop_i$.

证明. 设存在 $Stop_j \subset Stop_i$, 且 $\delta(s, Stop_i) = s_j, \delta(s, Stop_j) = s_k$, 那么当状态为 s 的时候, 如果当前输入为 $Stop_j$ 的实例时候, 那么自动机会产生两个状态转移 s_j, s_k 这与确定有限状态自动机对于一个输入只有唯一的状态转移这一性质相矛盾, 因此

引理 2 成立.

证毕.

如下方法为判断任意两个正则式序列类 R_1, R_2 是否满足 $R_2 \subseteq R_1$ 的基本算法. 该算法的基本思想是通过判定等价的命题 $\neg R_1 \cap R_2 = \emptyset$ 是否成立来判定 $R_2 \subseteq R_1$ 是否成立, 其中 $\neg R_1 = \{x \mid \neg x \in R_1\}$.

(1) 构建 R_1, R_2 对应自动机 $\mathcal{FA}(R_1), \mathcal{FA}(R_2)$;

(2) 构建 $\neg R_1$ 所对应的自动机 $\mathcal{FA}(\neg R_1)$;

(3) 构建 $\neg R_1$ 与 R_2 交集(交集为同时满足 $\neg R_1$ 与 R_2 的语义轨迹类, 记为 $\neg R_1 \cap R_2$) 所对应的自动机 $\mathcal{FA}(\neg R_1 \cap R_2)$. 如果 $\mathcal{FA}(\neg R_1 \cap R_2)$ 不存在从初始状态可达的可接受状态, 那么 $R_2 \subseteq R_1$ 成立. 否则 $R_2 \subseteq R_1$ 不成立.

第(2)中构建 $\mathcal{FA}(\neg R_1)$ 的方法是 $\mathcal{FA}(R_1)$ 中的所有接受状态变为非接受状态, 非接受状态变为接受状态, 其正确性可参考文献^[11]. 在(3)中 $\neg R_1$ 与 R_2 交集所对应的自动机 $\mathcal{FA}(\neg R_1 \cap R_2)$ 为如下形式:

$$\mathcal{FA}(\neg R_1 \cap R_2) = (S(\neg R_1) \times S(R_2), \Sigma,$$

$$\delta, (s_0(\neg R_1), s_0(R_2)), F(\neg R_1) \times F(R_2)),$$

其中, $S(\neg R_1), S(R_2)$ 分别为 $\neg R_1$ 与 R_2 所对应自动机的状态集集合; $S(\neg R_1) \times S(R_2)$ 为 $\mathcal{FA}(\neg R_1 \cap R_2)$ 的状态集, 它是两个状态集合的笛卡尔集; $s_0(\neg R_1), s_0(R_2)$ 分别为 $\neg R_1$ 与 R_2 所对应自动机的初始状态, $(s_0(\neg R_1), s_0(R_2))$ 为 $\mathcal{FA}(\neg R_1 \cap R_2)$ 的初始状态; $F(\neg R_1), F(R_2)$ 分别为 $\neg R_1$ 与 R_2 所对应自动机的接受状态集合, $F(\neg R_1) \times F(R_2)$ 为 $\mathcal{FA}(\neg R_1 \cap R_2)$ 的接受状态集合;

定理 1. 对于语义轨迹类包含关系的判定方法是可靠、完备的.

证明. 引理 1 保证了任意语义轨迹类的都可以转化为等价的自动机, 并通过类似正则表达式的包含关系判别方法来判定语义轨迹类的包含关系. 语义轨迹类包含关系的判定方法与判定正则式的包含关系过程相同, 而正则式包含关系的判定过程是有效、完备. 因此, 对于语义轨迹类包含关系的判定方法也是有效、完备.

算法 2 为基于语义轨迹约束的包含关系判定方法的事件探测算法.

算法 2. 事件探测算法.

Require: Semantic Trajectory $ST = \{s_{begin}, s_1, s_2, \dots, s_{end}\}$, Event Classes

$Events = \{E_1, E_2, \dots, E_n\}$;

Ensure: Detected Event Set Set_{Event} ;

1. Restriction $R \leftarrow \text{null}$;

2. for $i \leftarrow 1$ to $|ST|$ do

```

3.  $C \leftarrow \text{getClass}(ST[i]);$ 
4. if  $R = \text{null}$  then
5.    $R \leftarrow C;$ 
6. else
7.    $R \leftarrow R \cdot C;$ 
8. end if
9. end for
10. for  $e \in \text{Events}$  do
11.   $R_{\text{event}} \leftarrow \text{getRestriction}(e);$ 
12.  if  $R \sqsubseteq R_{\text{event}}$  then
13.    Create an instance of  $e;$ 
14.    Add the instance to  $\text{Set}_{\text{Event}};$ 
15.  end if
16. end for
17. Return  $\text{Set}_{\text{Event}};$ 

```

算法 2 中的步 2~9 为根据语义轨迹实例构建包含该轨迹的最小语义轨迹类的约束。其中,步 8 的 $\text{getClass}()$ 函数返回包含 Stop 实例的最小 Stop 类。步 10~16 为遍历视频事件类集合,通过对事件类的语义轨迹约束与轨迹的最小语义轨迹类的约束的包含关系判定,确定语义轨迹是否符合某事件类对应的语义轨迹特征。步 11 $\text{getRestriction}()$ 返回事件所对应的正则式序列类。如果语义轨迹具备某事件类所对应语义轨迹的语义特征,那么生成一个事件类的实例,并将该实例加入到 $\text{Set}_{\text{Event}}$ 。算法最终返回包含检测到事件类的对应实例集合 $\text{Set}_{\text{Event}}$ 。算法 2 的算法时间复杂度主要取决于步 12 中语义轨迹类包含关系判定过程的时间复杂度,而该步复杂度为 PSPACE-complete。所以算法时间复杂度为指数级。

4 实验分析

4.1 实验环境

本文实验数据集是关于商场的监控视频。该数据集来自 CAVIAR (<http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1>) 公共监控视频数据库第 2 个数据集中的正面拍摄图像。视频的分辨率为 384×288 。所有的这些视频都由同一个固定摄像头拍摄。实验在 CPU 为 AMD 3.0GHz,内存为 2GB,Windows XP 操作系统下进行。我们基于 OpenCV 实现了原始轨迹抽取算法,并通过 Java+Jena 实现了算法 1。同时我们用 Java+Pellete 实现了语义轨迹类包含关系判定的方法,并在此基础上实现了算法 2。

4.2 视频事件定义

我们根据商场视频的内容定义了 3 个事件 Shop-

AroundEvent、ShoppingEvent 以及 WalkEvent。它们的含义如下：

ShopAroundEvent. 行人有耐心地在商场购物。行人不仅进入了商场而且在商场做了一定时间的停留,挑选商品；

ShoppingEvent. 行人逛街。行人出行的目的是逛街购物但对视频中的商场并没有特别突出的兴趣,有可能行人只是在商场外看了一眼便离开或者经商场短暂停留后便离开去别的商场；

WalkEvent. 行人路过商场。行人对该商场并没有任何兴趣,直接从商场外走过。

这 3 个事件分别对应 3 个语义轨迹类,它们分别是 ShopAroundTrajectory、ShoppingTrajectory 和 WalkTrajectory。根据这 3 个语义轨迹类,可以将 3 个事件定义如下：

ShopAroundEvent \equiv

Event \sqsupseteq hasTrajectory.ShopAroundTrajectory,

ShoppingEvent \equiv

Event \sqsupseteq hasTrajectory.ShoppingTrajectory,

WalkEvent \equiv

Event \sqsupseteq hasTrajectory.WalkTrajectory.

对于 ShopAroundEvent,我们定义行人起始位置和结束位置都位于商场外也即走道内。在 ShopAroundEvent 中行人表现出了对商场商品的兴趣并在商场内逗留挑选商品,因此行人在商场内应该有较长时间的停留,我们将这个停留用本体类 ShopAroundStop 表示。

对于 ShoppingEvent,同样定义行人起始和结束位置都位于走道内。行人在商场外可能做了一定时间的停留观察,这一停留用本体类 OutOfShopShoppingStop 表示。行人也有可能进入商场,但很快就离开,这一停留用本体类 InShopShoppingStop 表示。

对于 WalkEvent,定义行人起始位置在走道入口,结束位置在走道出口或者起始位置在走道出口而结束位置在走道入口,表示行人从走道走过商场。

根据 Stop 类集合,我们将 3 个事件所对应的语义轨迹类的语义特征分别定义如下：

ShopAroundTrajectory \equiv Trajectory \sqcap (Seq(Move) \sqcap (RegExClass(r_1)));

$r_1 = (\text{PrimitiveStop} \sqsupseteq \text{IsLocatedIn.ShopWalkWay}) \cdot$

(PrimitiveStop) * \cdot ShopAroundStop \cdot

(PrimitiveStop) * \cdot (PrimitiveStop \sqcap

$\exists \text{IsLocatedIn.ShopWalkWay}$);

$$\text{ShoppingTrajectory} \equiv \text{Trajectory} \cap (\text{Seq}(\text{Move}) \cap (\text{RegExClass}(r_2)));$$

$$r_2 = (\text{PrimitiveStop} \cap \exists \text{IsLocatedIn. ShopWalkWay}) \cdot (\text{OutOfShopShoppingStop} | \text{InShopShoppingStop}) \cdot (\text{PrimitiveStop} \cap \exists \text{IsLocatedIn. ShopWalkWay});$$

$$\text{WalkTrajectory} \equiv \text{Trajectory} \cap (\text{Seq}(\text{Move}) \cap (\text{RegExClass}(r_3)));$$

$$r_3 = (\text{PrimitiveStop} \cap \exists \text{IsLocatedIn. Access}) \cdot (\text{PrimitiveStop} \cap \exists \text{IsLocatedIn. Access}).$$

4.3 实验分析

我们首先将视频分为单人和多人出现的视频, 然后比较在单人或多人情况下视频事件的探测的效率. 表 2 为实验中的一些相关参数值设置, 该参数为分析部分视频中行人行为之后确定.

表 2 实验参数

| Stop 概念 | 时间阈值 | 空间区域 R_{Δ} |
|-----------------------|--|------------------------------|
| ShopAroundStop | $(\delta_{\min} = 10\text{s}, \delta_{\max} = \infty)$ | $50 \times 70(\text{pixel})$ |
| OutOfShopShoppingStop | $(\delta_{\min} = 5\text{s}, \delta_{\max} = \infty)$ | $60 \times 20(\text{pixel})$ |
| InShopShoppingStop | $(\delta_{\min} = 3\text{s}, \delta_{\max} = 8\text{s})$ | $50 \times 70(\text{pixel})$ |

我们通过事件探测的准确率和召回率来验证我们提出的事件探测方法的性能, 其中事件探测的准确率定义为

$$\text{preciosn}(e) = \frac{|\text{正确探测到包含事件 } e \text{ 的视频}|}{|\text{探测到包含事件 } e \text{ 的视频}|}.$$

事件探测中的召回率定义为

$$\text{recall}(e) = \frac{|\text{正确探测到包含事件 } e \text{ 的视频}|}{|\text{探测到包含事件 } e \text{ 的视频}|}.$$

对比单人情况下的实验结果和多人情况下的实验结果, 可以发现单人情况下的事件探测的准确率和召回率都较高, 而在多人情况下事件探测的准确率和召回率都明显减低, 这主要因为在多人情况下 CamShift 算法对于行人的跟踪受到的干扰严重, 不能准确获取到行人的轨迹信息. 除此之外, 由于视频背景较为暗淡, 当行人身着深色衣服时也易导致跟踪失败, 从而影响最后的事件探测. 由此也可以看出, 轨迹信息的准确性影响事件探测性能的关键因素之一. 在单人情况下, 事件探测的准确率和召回率都比较高, 说明基于语义轨迹的事件探测方法具有较好的事件探测性能.

为了分析事件探测性能对于 Stop 中定义的空间阈值的敏感性, 我们针对 ShopAroundEvent 分析当 ShopAroundStop 为不同空间阈值下的事件探测性能. 我们将 ShopAroundStop 的空间阈值分别设为 20×20 、 50×70 、 70×100 . 在不同 Stop 空间阈值

情况下的事件探测性能如图 16、图 17 所示.

由图 16、图 17 可以看出, 随着空间阈值的增加, 事件探测的召回率逐渐提高, 而事件探测的准确率逐渐降低. 这说明 Stop 的阈值设置对于事件探测的性能存在一定的影响. Stop 空间阈值越小, 由原始轨迹产生的 Stop 实例越准确. 因此事件探测的准确率越高. 但空间阈值越小, 许多 Stop 实例可能被忽略, 因此事件探测的召回率减低.

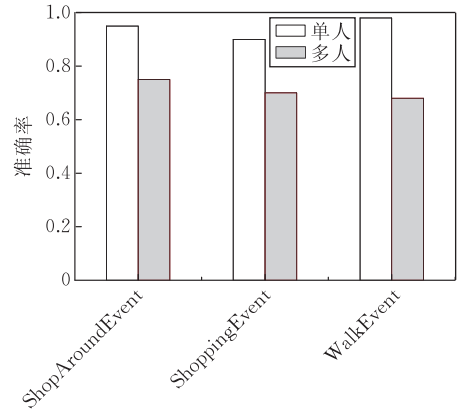


图 14 事件探测准确率

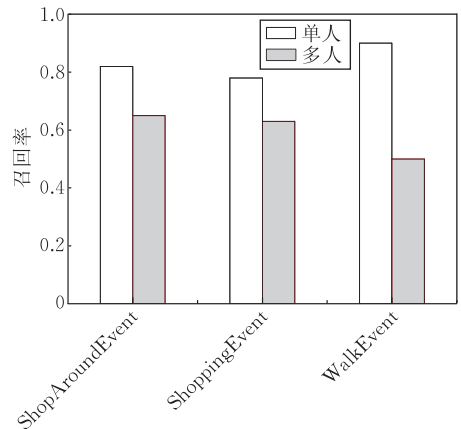


图 15 事件探测召回率

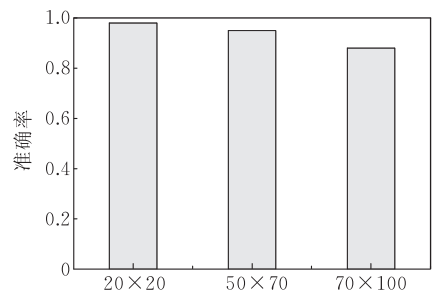


图 16 ShopAroundEvent 准确率

5 相关工作

目前视频事件探测可划分为基于模式识别的视

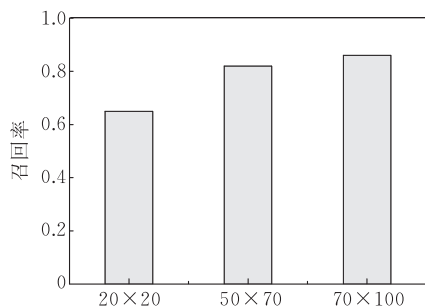


图 17 ShopAroundEvent 召回率

频事件探测方法和基于语义模型的视频事件探测方法. 基于模式识别的视频事件探测主要依赖于训练集数据所蕴含的知识来构建探测系统^[12-13,15]. 例如基于支持向量机模型的事件探测系统^[12]; 基于贝叶斯网, 条件随机场的事件探测系统^[13,16]. 构建基于模式识别的事件探测系统时往往需要大量的训练数据且这类系统不具备良好的知识扩展性.

相对于基于模式识别的事件探测, 基于语义模型^[17]的视频事件探测方法主要依赖于知识库中的知识, 具备更好的知识可扩展性. 基于逻辑的视频事件探测^[14,18]属于语义模型事件探测方法的一种, 其运用逻辑的方法对视频内容进行描述、推理进而判断事件的发生与否. 逻辑能对知识进行形式化表示, 并能通过严格的推理过程从已有的知识中获取新的知识, 是一种类似于人发现视频事件的过程. 由于大部分描述逻辑系统是可判定的, 相对于基于一阶逻辑的事件探测方法, 基于描述逻辑的事件探测方法具有更好的实用性.

轨迹是视频事件探测的一种重要视觉信息. Zhu^[9]等人提出用轨迹分析探测足球视频中的战术信息, Picicarelli^[5]等通过对轨迹进行分块聚类来识别视频中的异常事件. Patino^[19]等通过 agglomerative clustering 以及统计方法发现轨迹中的重要模式, 进而实现视频事件的探测. 与其它方法不同, 本文提出的基于语义轨迹的视频事件探测主要通过轨迹所蕴含的语义信息以及基于本体、正则表达式的推理技术来实现视频中的事件探测. 该方法能较好地将由本体表示的背景知识和视频中的对象运动轨迹信息进行事件探测, 对高层语义事件如停车、逛商店等, 具备较好的识别能力.

6 总结以及下一步工作

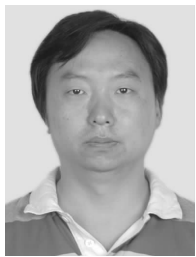
本文提出了一种基于语义轨迹的视频事件探测方法. 该方法能结合本体知识将原始轨迹转换为语义轨迹, 并通过基于正则式包含关系的推理方法, 实现基于语义轨迹的事件探测. 通过实验, 我们验证了

该方法的有效性. 然而, 我们也发现原始轨迹的准确性直接影响到事件探测的性能, 所以我们下一步的工作重点将是进一步研究对象跟踪算法, 提高准确抽取原始轨迹的能力, 尤其是在有多人的场景下的原始轨迹抽取. 本文中的事件及其对应的语义轨迹特征都是由用户事先定义的. 为了减轻用户负担, 我们将考虑用基于规则的学习算法来自动学习事件中的语义轨迹特征. 此外, 在未来的工作中, 我们将通过实验来比较基于语义轨迹的探测方法与基于一般轨迹的事件探测方法, 以进一步验证语义基于轨迹的事件探测方法的有效性.

参 考 文 献

- [1] Ebadollahi Shahram, Xie Lexing, Chang Shih-Fu, Smith John R. Visual event detection using multi-dimensional concept dynamics//Proceedings of the International Conference on Multimedia & Expo. Toronto, Ontario, Canada, 2006: 881-884
- [2] Dong Xu, Change Shih-Fu. Video event recognition using kernel methods with multilevel temporal alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(11): 1985-1997
- [3] Bertini Marco, Bimbo Alberto Del, Serra Giuseppe. Learning ontology rules for semantic video annotation//Proceedings of the 2nd ACM workshop on Multimedia semantics. Vancouver, British Columbia, Canada, 2008: 1-8
- [4] Morris Brendan Tran, Tivedi Mohan Manubhai. Survey of vision-based trajectory learning and analysis for surveillance. IEEE Transactions on Circuits and Systems for Video Technology, 2008, 18(8): 1114-1127
- [5] Picicarelli C, Foresti G L. On-line trajectory clustering for anomalous events detection. Pattern Recognition Letters, 2006, 27(15): 1835-1842
- [6] Spaccapietra Stefano, Parent Christine, Dmiani Maria Luisa, Macedo Jose Antonio de, Porto Fabio, Vangenot Christelle. A conceptual view on trajectories. Data and Knowledge Engineering, 2008, 65(1): 126-146
- [7] Yan Zhixian, Macedo Jose, Parent Christine, Spaccapietra Stefano. Trajectory ontologies and queries. Transactions in GIS, 2008, 12: 75-91
- [8] Bradski Gary R. Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, 1998, 2(2): 12-21
- [9] Fukunaga Keinosuke. Introduction to Statistical Pattern Recognition (2nd edition). San Diego, USA: Academic Press Professional Inc, 1990
- [10] Baader Franz et al. The Description Logic Handbook: Theory, Implementation and Applications (2nd edition). Cambridge, UK: Cambridge University Press, 2002

- [11] Hopcroft John, Motwani Rajeev, Ullman Jeffery D. Introduction to Automata Theory, Languages, and Computation (2nd edition). Boston, USA; Addison-Wesley Longman Publishing, 2000
- [12] Pittore Massimiliano, Basso Cruzio, Verri Alessandro. Representing and recognizing visual dynamic events with support vector machines//Proceedings of the International Conference on Image Analysis and Processing. Venice, Italy, 1999; 18-23
- [13] Hongeng Somboon, Nevatia Ram, Bremond Francois. Video-based event recognition: Activity representation and probabilistic recognition methods. Computer Vision and Image Understanding, 2004, 96(2): 129-162
- [14] Shet Vinary D, Harwood David, Davis Larry S. Vidmap: Video monitoring of activity with prolog//Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance. Como, Italy, 2005; 224-229
- [15] Piciarelli C, Foresti G L, Snidaro L. Trajectory clustering and its applications for video surveillance//Proceedings of the IEEE International Conference on Advanced Video and Signal based Surveillance. Como, Italy, 2005; 40-45
- [16] Zhang Jianguo, Gong Shaogang. Action categorization with modified hidden conditional random field. Pattern Recognition Letters, 2010, 43(1): 197-203
- [17] Ghanem Nagia, DeMenthon Daniel, Doermann David, Davis Larry. Representation and recognition of events in surveillance video using Petri nets//Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop. Washington, D. C. USA, 2004; 112-120
- [18] Fusier Florent et al. Video understanding for complex activity recognition. Machine Vision and Applications, 2007, 18(3): 167-188
- [19] Patino L, Behhadda H et al. Extraction of activity patterns of large video recordings. IET Computer Vision, 2008, 2(2): 108-128



WANG Xiao-Feng, born in 1978, Ph. D. candidate. His current research interests include logic based reasoning, data mining, machine learning, semantic Web, multi-agent systems, action planning, cognitive science.

Zhang Da-Peng, born in 1979, Ph. D. candidate. His current research interests include artificial intelligence theories.

WANG Fei, born in 1987. His research interests include image processing, pattern recognition, data mining.

SHI Zhong-Zhi, born in 1941, professor, Ph. D. supervisor. His research interesting include artificial intelligence, data mining, multi-agent systems, cognitive science.

Background

Video event detection is a key research problem for automatic video content understanding. Trajectories of moving object in video are the one of the important clue for video event detection. Currently, the most trajectory based event detection methods focus on the geometric characteristics of the trajectory and neglect the semantic information related to the trajectories. However, as we known the trajectory producing process is affected by the trajectory related semantic information, such as geographic information related to trajectory etc. Combine the semantic information related to the trajectory with raw trajectory information can make us obtain more knowledge of trajectories. Semantic trajectory provides a way to effectively combine trajectory information with semantic knowledge. In this paper, we apply the semantic traj-

ectory into video event detection and propose a semantic trajectory based method for video event detection. Based on the empirical studies, we demonstrated that the proposed method can effectively detect the video events.

The authors of this paper focused on developing methods and techniques for intelligent information processing for many years. They proposed many creative works on intelligent multimedia processing, such as automatic image labeling, object categorization for image understanding, logic based video content understanding etc. The proposed method in this paper is a part of research works of National Basic Research Priorities Program (No. 2007CB311004) which is aimed to represent and understand the content of unstructured information.