

监控视频运动目标的频繁轨迹模式挖掘*

代科学 张 军 李国辉 来 旭

(国防科技大学 信息系统与管理学院 湖南 长沙 410073)

摘要 运用数据挖掘的方式分析运动轨迹数据在目前可以说是一种尝试。基于传统数据挖掘方法 Apriori 的思想,提出了一种监控视频运动目标的频繁轨迹模式挖掘算法。首先以特征点表示连续的运动目标轨迹,然后设计了一种轨迹相似度测量机制,最后基于相似度测量获得的轨迹段频率,将 Apriori 挖掘算法应用到 2D 运动轨迹数据,从中发现频繁轨迹隐含的一些感兴趣模式。实际的和人工生成的轨迹数据实验表明了算法的正确有效性。

关键词 视频挖掘 监控视频挖掘 轨迹数据挖掘

中图分类号 :TP391 文献标识码 :A

Mining Frequent Trajectory Patterns of Moving Objects
from Surveillance Video

DAI Ke-xue ZHANG Jun LI Guo-hui LAI Xu

(College of Information System and Management, National Univ. of Defense Technology, Changsha 410073, China)

Abstract :It is only a tentative attempt to analyze motion trajectory data in the data mining style. This paper proposes an algorithm based on Apriori, a conventional data mining method, for mining frequent trajectory patterns of moving objects from surveillance video. Firstly, feature points are extracted for presenting continuous trajectories. Then a trajectory similarity measure is designed to obtain the frequency of trajectories. Finally, based on the values, Apriori algorithm is applied to automatic finding of interesting patterns in 2D object trajectories. Experiments on real life and man-made trajectory data set show the validity of the algorithm.

Key words video mining ; surveillance video mining ; trajectory data mining

面对越来越容易获取的大量视频数据,人们也越来越需要一些技术自动从中发现规则、规律信息,实现对视频数据的自动理解。轨迹特征是视频运动目标的一个主要特征,很多视频存储和检索技术都依靠它^[1-2]。但是,用数据挖掘的方式来分析这类数据的研究还开展得不够^[3],已有的文献主要是以数据统计或神经网络的方法实现轨迹模式学习和目标行为预测。如 Stauffer 等通过建立目标位置和速度的统计模型学习出目标的行为模式^[4]。Sumpter 等以神经网络来建立目标运动轨迹时空模式的统计模型,实现对目标行为的预测^[5]。胡卫明等采用层次自组织神经网络方法对运动目标轨迹的分布模式进行学习,利用学习得到的轨迹分布模式检测可能的异常轨迹事件,以及对目标行为进行预测^[6]。马国兵等采用神经网络和混配遗传算法对运动目标轨迹进行预测,取得了非常高的精度^[7]。

本文提出了一个发现运动目标频繁轨迹模式的监控视频挖掘算法。挖掘的模式隐含在多个运动目标的轨迹段中,而这些轨迹段具有一定的空间或时间关联关系。根据频繁轨迹隐含的模式知识,可以实现异常轨迹行为的检测告警。该问题与文献^[8]都是通过计算曲线段的相似度从曲线数据中发现模式,但比之更复杂。因为时间序列曲线是一维的,而轨迹数据可能是二维或者三维的,还可以包含速度和加速度。

在传统的“购物篮”模式挖掘中^[9],一个事务是一个项集,挖掘出的模式包含很多项。本文提出的

* 收稿日期 :2006-06-30
基金项目 :国家自然科学基金资助项目(60273066)
作者简介 :代科学(1976—),男,博士生。

轨迹模式挖掘跟它类似,“购物篮”的项等价于一个单位长度的轨迹段,一个事务就是一条轨迹的一个分段,而模式等价于单位长度的轨迹段集。这里,轨迹段指的是运动目标整个轨迹的一部分。挖掘过程的第一步是对运动目标的轨迹数据进行预处理,提取出轨迹的特征点,并转换轨迹表示方式,然后计算轨迹段的相似度,最后利用 Apriori 思想^[10],设计一个模式长度增长算法,由此挖掘运动目标轨迹数据中的频繁轨迹模式。

1 轨迹数据预处理

本文采用文献 [11-12] 的运动目标检测跟踪算法提取轨迹数据。在获得轨迹曲线后,利用文献 [2] 的方法对轨迹坐标数据进行转换,抽取轨迹的特征点。这样可以减少原始轨迹位置的测量误差,有利于后续的相似度测量,并且减小数据集的大小,加快模式挖掘的速度。

目标的轨迹特征由高曲率的轨迹点表示,即通过计算三个相邻的高曲率点得到轨迹特征点,然后将轨迹特征点坐标 (x, y) 转换为以 (λ, θ) 表示。 λ 表示相邻特征点的距离, θ 表示相邻两条线的夹角,轨迹右转为正方向,左转为负方向。这里,轨迹曲率和方向的判断依据是运动轨迹在图像坐标下的由左至右的位置数据。如图 1 所示。具体算法请见文献 [2]。

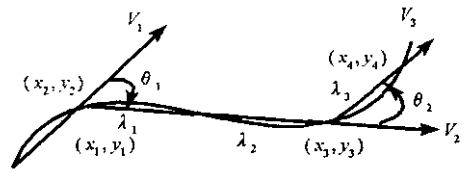


图 1 轨迹特征点抽取及其表示转换
Fig. 1 Feature points of trajectory and its converted presentation

这种表示方法对旋转和平移变换不敏感,而对伸缩敏感。它能表示任意长度轨迹段的局部信息,因为每个 (λ, θ) 只依据三个邻接点。当然,这种表示方法对轨迹段的相似度量不尽一致, $k+1$ 个 (x, y) 特征点只有 $k-1$ 个 (λ, θ) , 实际上有 k 个 λ 值,但只有 $k-1$ 个 θ 值。因为 λ 值对应轨迹段的一条线,而 θ 值对应两条线的夹角, k 条线对应 $k-1$ 个夹角。因而两轨迹段的相似度量不考虑最后一条线的长度。

2 轨迹相似度计算

在传统事务数据库中,模式频率通常被定义为包含组成模式的项集的事务个数。但是对于运动轨迹数据,表示事务的轨迹段数据并不完全相等或一致,事务中项的支持度和整个模式的支持度应基于一定的轨迹段相似度测量方法。即将模式的频率定义为模式与事务之间的相似度。因此,正确的轨迹相似度测量方法至关重要,它应满足下列要求:

- 相似度测量结果必须与人的判断一致。
- 对旋转和平移变换不敏感,而对伸缩敏感,以体现轨迹形状上的相似。
- 效率高,有助于模式挖掘算法。

假设 s_1 和 s_2 是长度为 n 的两条轨迹段, $s_1 = ((\lambda_1^1, \theta_1^1), \dots, (\lambda_n^1, \theta_n^1)), s_2 = ((\lambda_1^2, \theta_1^2), \dots, (\lambda_n^2, \theta_n^2))$ 根据长度的比值越接近 1 越相似,夹角的差值越接近 0 越相似的原则,本文定义的相似度测量方法如下:

$$sim(s_1, s_2) = \begin{cases} D - dist(s_1, s_2), & D - dist(s_1, s_2) > T_{sim} \\ 0, & otherwise \end{cases} \quad (1)$$

其中

$$dist(s_1, s_2) = \frac{1}{n} \left[\alpha \cdot (n - \sum_{i=1}^n r_i) + \sum_{i=1}^n |\theta_i^1 - \theta_i^2| \right] \quad (2)$$

$$r_i = \begin{cases} \frac{\lambda_i^1}{\lambda_i^2}, & \lambda_i^1 \leq \lambda_i^2 \\ \frac{\lambda_i^2}{\lambda_i^1}, & \lambda_i^2 < \lambda_i^1 \end{cases} \quad (3)$$

D 是一个给定的最大相似度, T_{sim} 是相似度阈值, α 是一个调整距离长度和夹角的权重。

3 频繁轨迹模式挖掘

3.1 定义频繁轨迹段的支持度

模式挖掘算法的目的是发现经常出现的连续轨迹段。在 Apriori 算法框架中, 模式存在于模式包含的项集中。一个项就是一个 λ 长度的轨迹段, 即 (λ, θ) 。项集就是所有轨迹的所有 (λ, θ) , 也就是最长的频繁而连续的轨迹段集。一个候选模式就是一段频繁而连续的轨迹段, 即 (λ, θ) 列表。

一个轨迹段的出现频率根据它与其他轨迹在形状上的相似度计算得到。候选模式 p 的支持度定义为轨迹段 p 与与其等长的其他所有轨迹段的相似度的和。如果候选模式 p 与与其等长的轨迹段 q 的相似度都大于相似度阈值, 则轨迹段 q 是模式 p 的一个实例, 该轨迹段的第一个特征点的起始时间就是该实例的开始时间, 该实例的相似度值被加入该候选模式的支持度中。

轨迹段 p 的支持度定义为

$$supp(p) = \sum_{q \in TB} sim(p, q) \tag{4}$$

$TB = \cup T$, T 是给定轨迹集中的任意一个运动目标的任意轨迹。

如果 $supp(p) > T_{supp}$, 则候选模式 p 是频繁的。 T_{supp} 是一个给定的支持度阈值。

3.2 前后缀连接产生候选模式

Apriori 算法用于发现最大长度模式, 即不被其他模式包含的模式。对 k 长度的模式进行前后缀连接是产生 $k + 1$ 长度候选模式的有效方法。但是计算量非常大, 因为在计算 $k + 1$ 长度的候选模式时, 需要遍历长度为 k 的所有候选模式。因此, 候选模式的表示和索引非常重要。本文中的候选模式用一个数字数组表示, 并且按级组织, 即级 k 就是所有长度为 k 的模式。然后根据模式的前缀建立索引, 索引表的每个记录存储具有相同前缀的元素。表 1 是索引表的一部分, 其中 $\pi[i]$ 是索引项, i 是索引号。该索引表能够索引的模式长度为 4, 索引长度为 2, 项数为 6, 即有 6 个频繁而连续的轨迹段, 分别映射为数字 0 到 5。模式 $p = \langle s_1, s_2, \dots, s_k \rangle$ 的索引号定义为 $s_1 * 6 + s_2$ 。 s_1, s_2, \dots, s_k 是数字, 代表模式项, 即一个单位轨迹段, 根据索引号的定义规则 $s_1 * 6 + s_2$ 映射而来。例如 $\langle 0, 1, 5, 2 \rangle$ 表示一个长度为 4 的模式, 该模式的第一项映射为“0”, 第二项映射为“1”, 而“1”索引号能够搜索到所有的 6 个 $\langle 0, 1, *, * \rangle$ 模式。 $\pi[0]$ 的记录为空, 表示没有可与其进行后缀连接的模式。

表 1 模式索引表示例
Tab. 1 Sample index table

索引表项	候选模式
$\pi[0]$	
$\pi[1]$	$P0\{0, 1, 1, 5\}$ $P1\{0, 1, 2, 5\}$ $P2\{0, 1, 3, 0\}$ $P3\{0, 1, 3, 5\}$ $P4\{0, 1, 5, 1\}$ $P5\{0, 1, 5, 2\}$ $P6\{0, 1, 5, 5\}$
$\pi[2]$	$P0\{0, 2, 1, 5\}$ $P1\{0, 2, 5, 1\}$ $P2\{0, 2, 5, 5\}$
$\pi[3]$	$P0\{0, 3, 1, 5\}$
...	...
$\pi[15]$	$P0\{2, 3, 1, 5\}$ $P1\{2, 3, 5, 1\}$ $P2\{2, 3, 5, 5\}$
...	...

前后缀连接算法依据索引表, 对索引表的元素迭代计算实现前后缀连接。在对索引表的元素进行迭代时, 也对当前级的所有模式进行迭代。表的索引长度记为 j , 级 k 的元素记为 e , 从已有的两个模式中产生新的候选模式记为 $e \oplus q$ 。算法要点是:

- (1) 开始对整个索引表进行遍历。
- (2) 找出下一个元素 e , 假定 e^{k-1} 为 e 的 $k - 1$ 长度的后缀, e' 为 e^{k-1} 的 j 长度的前缀, 在索引表中可与 e 连接的元素的索引为 e' 。

(3) 遍历表的记录,找到索引为 e' 的记录。

(4) 同步骤(3)的方法找到下一个元素 q 。

(5) 如果 e 的 $k-1$ 长度的后缀等于 q 的 $k-1$ 长度的前缀,就将 e 和 q 连接,生成新的候选模式 $e \oplus q$ 。

(6) 转步骤(4),直到遍历完毕整个表的记录。

(7) 转步骤(2),直到整个索引项遍历完毕。

例如,假定 $e = \langle 1\ 2\ 3\ 1 \rangle$, $e' = \langle 2\ 3 \rangle$ 。那么索引为 $2 * 6 + 3 = 15$,通过遍历,找出以 $\langle 2\ 3 \rangle$ 为前缀的模式,如 $\{\langle 2\ 3\ 1\ 5 \rangle, \langle 2\ 3\ 5\ 1 \rangle, \dots\}$,则新的候选模式是 $\{\langle 1\ 2\ 3\ 1\ 5 \rangle\}$ 。

3.3 发现频繁而连续的轨迹段

为了自动产生所有的 k 长度候选模式,需将所有的项进行组合,有可能产生 N^k 个候选模式(N 是项集的个数)因此要利用 Apriori 的单调性,在需要计算支持度之前剪除候选模式,从而减少计算量。并且在产生 $k+1$ 级候选模式时只将 k 级候选模式保存在内存中。频繁轨迹模式挖掘算法的要点如下。

首先定义如下符号:

C_k : 候选模式集 k , 包含长度为 k 的所有候选模式。

L_k : 级 k , 包含长度为 k 的所有模式。

Result: 最大长度模式列表。

p_k : 一个长度为 k 的模式。

$\text{suffix}^k(p)$: 模式 p 的 k 长度后缀。

$\text{prefix}^k(p)$: 模式 p 的 k 长度前缀。

算法步骤(输入参数为运动目标轨迹集、项集(λ, θ)集合) T_{sim}, T_{supp}):

(1) 将所有长度为 1 的项加入 C_1 。

(2) 计算 C_1 中候选模式的支持度,将频繁的模式加入 L_1 。

(3) $k = k + 1$ 。

(4) 产生 C_k : 连接 L_{k-1} 中 $\text{suffix}^{k-2}(p_1) = \text{prefix}^{k-2}(p_2)$ 的模式 p_1 和 p_2 , 形成一个新的长度为 k 的候选模式 p_k , 将其加入 C_k 。

(5) 计算 C_k 中每个候选模式的支持度,如果 p_k 是频繁的,则将 p_k 加入 L_k 。

(6) 对于 L_{k-1} 中所有不是 L_k 中任何模式的前缀或后缀的模式 p_{k-1} , 将其加入 Result 列表。

(7) 转第(3)步重复执行,直到 L_k 为空。

对于 Apriori 模式挖掘算法,模式支持度必须满足的一个重要属性就是“单调性”。意思即,如果模式的长度延长了,但模式的支持度不增加。可以显然看出本文定义的候选模式及其相似度测量能够保持前后缀的单调性。因为对于任意的轨迹段 p, q, p', q', a, b , 有下式成立(证明略):

$$p' = pa, q' = qb \Rightarrow \text{sim}(p, q) \geq \text{sim}(p', q') \quad (5)$$

4 实验结果与分析

有关运动目标及其轨迹的数据通过视频处理技术提取并存储在一个文件中,然后在模式挖掘算法中定义一些数据结构进行调用。比如定义数组记录视频段及其中的运动目标信息,定义轨迹点 (x, y) 和 (λ, θ) 表示的转换方法等。模式挖掘结果是一个所有最大长度模式的列表,也记录在一个文件中,用户可以选择其中一些通过图形界面画出来。

4.1 挖掘实验结果及其解释

表 2 是 10 条运动目标轨迹的 4 单位长度 (λ, θ) 的数据,即根据每条轨迹的 4 段轨迹段数据发现频繁模式。图 2 是根据特征数据生成的对应轨迹段,其中 $S_i (1 \leq i \leq 10)$ 是轨迹段的编号,生成时未考虑第一轨迹小段的方向,均定为水平方向,仅作参考。

实验中 $\alpha = 1, D = 10$ (表示最大相似度为 10)。挖掘结果表明,表 2 中相似度大于 8.5 的有 8 对轨迹,最频繁的是轨迹 S1,有 3 条与之相似,分别是 S3, S6, S7。其中, S1 和 S6 的特征点数据 (λ, θ) 值完全一致(见表 2),因此它们形状非常相似(见图 2),相似度为最大值 10(见图 3)。

表 2 部分轨迹测试数据

Tab.2 Partial trajectory feature data for test

TID	ITEM0	ITEM1	ITEM2	ITEM3
1	10, 20.5	3, 20	6, 10.3	8, 2.2
2	3, 20.5	5, 20	8, -15	20, 15
3	10, 20.5	3, 20	6, 10	8, 2
4	3, 20.5	5, 6.2	7, 14.1	25, 4
5	3, 20.5	5, 6	7, 14.2	25, 5
6	10, 20.5	3, 20	6, 10.3	8, 2.2
7	10, 20.5	3, 20.5	6, 10	8, 2
8	8, 20.5	7, 20	6, 18	21, 2
9	11, 20.5	3.4, 20	5.8, 10.3	12, 33
10	3.2, 20	5, 20.2	8, -10	21, 15

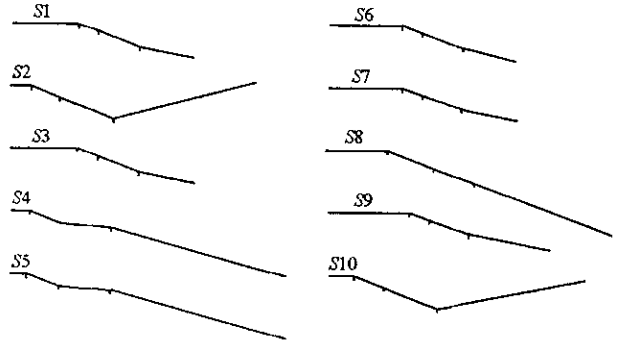


图 2 根据特征点数据生成的轨迹段
Fig.2 Trajectories illustrated with feature data

轨迹	相似度	距离
S1S2	0.0000	10.0125
S1S3	9.8750	0.1250
S1S4	0.0000	5.3307
S1S5	0.0000	5.6557
S1S6	10.0000	0.0000
S1S7	9.7500	0.2500
S1S8	0.0000	2.3228
S1S9	0.0000	7.8436
S1S10	0.0000	8.9373
S2S3	0.0000	9.9875
S2S4	0.0000	13.5583
S2S5	0.0000	13.3813
S2S6	0.0000	10.0125
S2S7	0.0000	10.1125

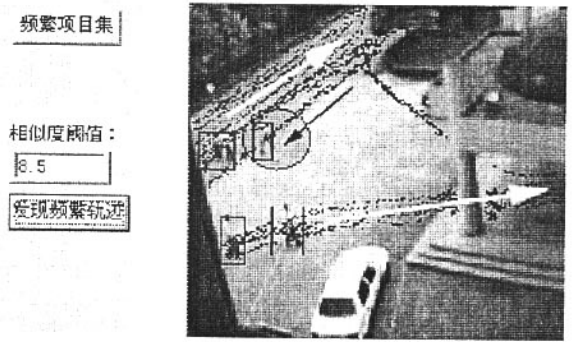


图 3 根据轨迹相似度发现频繁轨迹

Fig.3 Frequent trajectory patterns mining with similarity measure

图 4 实际模式解释

Fig.4 Practical explaining of patterns

根据频繁轨迹段的发生时间和多条频繁轨迹之间的时空关系,结合具体的监控场景,就可以解释频繁轨迹模式体现的语义信息。比如,对于图 4 所示的办公楼前的上班监控画面,通过挖掘可以发现单向主干道的和进入办公大楼的两种频繁轨迹,并具有如图中白色箭头所指的运动方向。当有目标的运动轨迹与“频繁轨迹”相反,如图中黑色箭头所指,监控系统就可以给出告警事件,说明该运动轨迹不是频繁的,可能存在异常;同时,主干道上的“频繁轨迹”和进入办公大楼的“频繁轨迹”存在时间关系,这种时间关系模式也可以体现一定的语义信息,不同时间段它们同时存续的时间长短不同,甚至不同时存在,这都可以结合具体的监控内容进行解释。

4.2 性能分析

为了测试算法对大规模数据的性能,实验时以半随机方式产生一些测试数据,先记录真实轨迹的所有 λ 和 θ 值,然后计算它们的累计直方图。利用这些直方图,设计一个随机轨迹数据产生器,使新产生的轨迹的 λ 和 θ 值符合真实数据的概率,并且轨迹的长度相同。性能测试的参数有:支持度阈值,相似度阈值,数据集大小(轨迹数和轨迹长度)以及项数,测试时只改变其中一个参数值(如表 3)。

由于模式支持度是相似度的和,候选模式是否频繁跟支持度阈值 T_{supp} 有关。为了很好地确定 T_{supp} ,实验时定义了一个支持率进行衡量,如 0.3,意思即模式的实例必须至少占有所有轨迹的所有可能实例总数的 30%,才认为该模式是频繁的。 $T_{supp} = \text{支持率} \times \text{轨迹数} \times \text{候选模式在一条轨迹中的所有可能实例数} \times T_{sim}$ 。其中,候选模式在一条轨迹中的所有可能实例数为:轨迹长度 - $k + 1$ 。实验结果表明:

(1)模式挖掘过程消耗的时间主要在计算候选模式的支持度,大约占 99%。这是正常的,因为大多

数关联规则挖掘都这样,这也说明本文的候选模式挖掘算法的设计和实现是正确和有效的。

(2)相似度和支持度阈值对算法效率的影响比较大,当增加相似度阈值或者降低支持度阈值,模式挖掘算法的性能都会降低。

(3)相似度阈值比支持度阈值对模式挖掘算法性能的影响更大。较低的相似度阈值会将较多的轨迹段认为是频繁的,模式项的增加会引起效率成指数级下降,并增加模式产生过程的时间(图5),因为可能的候选模式随项数成指数级增加。

(4)当固定其他参数,只改变数据集的大小也就是轨迹数多少的时候,算法的性能与其成反比关系,因为计算每个候选模式的支持度所需的时间线性增加了。

表3 测试参数值

参数名称	参数值
支持率	3%
相似度阈值	8.55
目标轨迹数	1500
轨迹长度	8
模式的项数	50

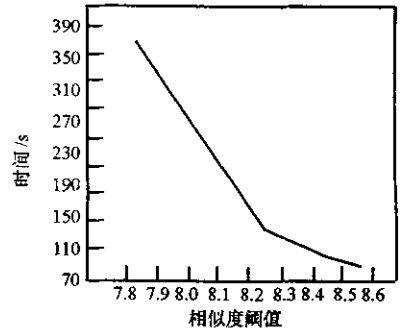


图5 模式挖掘的效率与相似度阈值的关系
Fig.5 Efficiency of candidate pattern finding vs. similarity threshold

5 结束语

实验表明相似度测量方法是正确的,模式挖掘算法能够挖掘出频繁的轨迹段。但是,目前挖掘的频繁轨迹模式不考虑具体的运动目标,而且模式轨迹段较短,这些模式的意义与价值也还需要大量的真实数据来检验。由于模式的长度与轨迹特征点的个数相等,轨迹段的形状对模式支持度的影响比较大,为了使算法能在大规模真实轨迹数据中有效,进一步的工作是设计较好的轨迹平滑和特征点提取技术,用以较少的特征点表示整个监控画面中所有运动目标轨迹的大致形状,从而得出更长的频繁模式轨迹。

参考文献:

- [1] Chang S F, Chen W, Meng H, et al. A Fully Automated Content-based Video Search Engine Supporting Spatiotemporal Queries[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1998, 8(5): 602-615.
- [2] Little J J, Gu Z. Video Retrieval by Spatial and Temporal Structure of Trajectories[A]. In Proceedings of SPIE Storage and Retrieval for Media Databases[C], San Jose, CA, 2001: 545-552.
- [3] Dimitrijevic M. Mining for Co-occurring Motion Trajectories: Sport Analysis[D]. MS Thesis of Univ. of British Columbia, 2002.
- [4] Stauffer C, Eric W, Grimson L. Learning Patterns of Activity Using Real-time Tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 747-757.
- [5] Sumpster N, Bulpitt A. Learning Spatio-temporal Patterns for Predicting Object Behavior[J]. Image and Vision Computing, 2000, 18(9): 697-704.
- [6] 胡卫明, 谢丹, 谭铁牛, 等. 轨迹分布模式学习的层次自组织神经网络方法[J]. 计算机学报, 2003, 26(4): 417-426.
- [7] 马国兵, 薛安克. 数据挖掘技术在运动目标轨迹预测中的应用[J]. 计算机工程与应用, 2004, 40(11): 210-211.
- [8] Das G, Lin K, Mannila H, et al. Rule Discovery from Time Series[A]. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining[C], New York, 1998: 16-22.
- [9] Hipp J, Guntzer U, Nakaeizadeh G. Algorithms for Association Rule Mining-a General Survey and Comparison[A]. In Proceedings of ACM International Conference on Knowledge Discovery and Data Mining[C], Boston, MA, USA, 2000, 2(1): 58-64.
- [10] Agrawal R, Lin K, Sawhney H S, et al. Fast Similarity Search in the Presence of Noise, Scaling and Translation in Time-series Databases[A]. In Proceedings of the 21st International Conference on Very Large Databases[C], Zurich, Switzerland, 1995: 490-501.
- [11] 林洪文, 姚作樑, 涂丹, 等. 基于减背景技术的运动目标检测方法研究[J]. 国防科技大学学报, 2003, 25(3): 66-69.
- [12] 田志刚, 涂丹, 李国辉. 一种基于概率匹配的视频运动目标跟踪算法[A]. 国防科技大学第三届研究生学术活动节论文集[C], 2003.

