

加速的 TLD 算法及其在多目标跟踪中的应用^①

金 哲, 刘传才

(南京理工大学 计算机科学与工程学院, 南京 210094)

摘 要: TLD (Tracking-Learning-Detection)算法是近期受到广泛关注的一种长时间视觉跟踪算法. 为提高该算法的运行速度, 一种 ATLD (Accelerated TLD)算法被提出, 对原始 TLD 算法做了两方面改进: 在检测模块引入基于灰色预测模型的目标位置估计和检测区域设置; 运用基于 NCC (Normalized Cross Correlation)距离的图像检索方法管理正负样本集. 并在此基础上实现了多目标跟踪. 通过实验比较了 ATLD 算法、原始 TLD 算法及两种近期改进的 TLD 算法. 实验结果表明: ATLD 算法在确保精度的前提下运行速度更快.

关键词: 目标跟踪; TLD; 目标位置估计; 图像检索

Accelerated TLD Algorithm and its Application in Multiple Target Tracking

JIN Zhe, LIU Chuan-Cai

(College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: Tracking-Learning-Detection (TLD) is a kind of long-term visual tracking algorithm which received wide attention in recent years. In order to improve the running speed of this algorithm, a novel algorithm named Accelerated TLD (ATLD) is proposed in this paper. Two aspects of improvements were made in original TLD algorithm. The improvement includes as follows: using a grey prediction model in the detection module for estimating the location of the target and setting a detection area; applying an image indexing method based on normalized cross correlation (NCC) distance to manage the positive and negative sample set. And on this basis, the multiple targets tracking algorithm is realized. Through experiments, the ATLD algorithm, the original TLD algorithm and other two recent improved TLD algorithm are compared. The experimental results show that the ATLD algorithm runs faster on the premise of ensuring the accuracy.

Key words: target tracking; tracking-learning-detection; target location prediction; image index

目标跟踪技术是计算机视觉的一个重要分支, 各国学者提出了众多的目标跟踪算法, 较全面的综述参见文献[1]. 一般的跟踪算法可分为两类: 基于跟踪的算法(Tracking-Based Algorithms)和基于检测的算法(Detection-Based Algorithms). 基于跟踪的算法, 如光流法^[2], 以先前帧中目标的信息为先验知识进行逐帧跟踪, 并假设跟踪过程中不会出现完全遮挡或目标消失等情况. 这方向的研究一般关注于算法的运行速度和精确度, 或者采用一些可靠方法来延长跟踪的时间^[3]. 基于检测的算法则是在已知的目标信息中提取某

些特征, 并在当前帧图像中检测目标位置从而达到跟踪的效果, 如 mean-shift 算法^[4]、粒子滤波算法^[5]. 基于跟踪的算法, 前端需要跟检测模块相互配合, 当检测到被跟踪目标之后, 就进入跟踪模块, 此后, 检测模块不会再介入到跟踪过程中. 这种方法有一个缺陷: 即当被跟踪目标存在消失或遮挡时, 跟踪就很容易失败, 因此, 对于长时间跟踪, 更多人采用基于检测的方法. 然而, 该方法虽然在某些情况下可以改进跟踪效果, 但它需要一个离线的学习过程. 即: 在检测之前, 需要挑选大量的被跟踪目标的样本进行学习和训

^① 基金项目:国家自然科学基金(61373063)

收稿时间:2015-10-25;收到修改稿时间:2015-11-19 [doi:10.15888/j.cnki.csa.005222]

练. 并且, 训练样本要涵盖被跟踪目标可能发生的各种形变和各种尺度、姿态变化和光照变化的情况, 否则, 跟踪的鲁棒性就难以保证.

考虑到单纯的跟踪和检测算法都无法达到较好的长时间跟踪效果, Kalal 等人将两者结合起来, 并加入了一种在线学习机制, 提出了一种跟踪-学习-检测算法(Tracking-Learning-Detection, TLD)^[6-9]. TLD 算法解决了目标在消失或被遮挡后无法重新捕获的问题, 不仅如此它在效率和准确性方面都表现得较为出色, 还能够较好地适应目标的外观变化, 在给定较少先验知识的情况下, 能够迅速地学习目标特征并进行有效跟踪.

自 TLD 算法提出以来得到了广泛的关注, 已经有学者提出了改进和完善的方法. Gemignani 等人^[10]在跟踪模块引入了 MCMC 粒子滤波方法改善了跟踪模块的精确度. Guo 等人提出了一种自适应的 TLD^[11], 用粒子滤波方法预测目标尺寸变化, 提高了检测模块的效率. 龚小彪在 TLD 框架中加入了 kalman 滤波等算法^[12], 使跟踪算法能够较好地处理遮挡问题, 甚至目标被完全遮挡, 跟踪器也能给出预测位置. Nebehay 等人对 TLD 的整体框架进行了改进, 如图 1 所示, 此外他们还对融合器和学习模块的部分内容做了改进并得到了较好的效果, 详细的内容可参见文献[13].

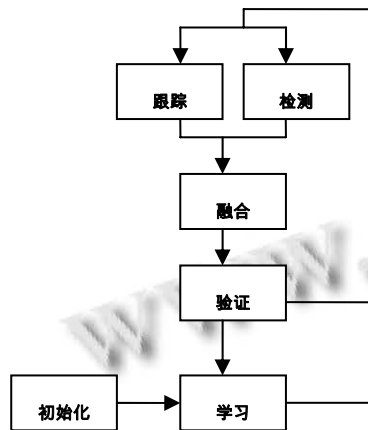


图 1 改进的 TLD 算法框架图

本文参考 Nebehay 等人的工作, 在此基础上进一步改进以求获得更高的运行效率. 本文的工作主要包括如下三个方面: (1) 采用灰色预测模型估计目标当前的位置, 并为检测模块设置一个检测区域, 仅对与检测区域有重叠的子窗口进行处理, 提高检测模块的

效率. (2) 采用一种基于 NCC 距离的图像检索方法来管理正负样本集, 以提高样本查找的效率, 从而提高检测和学习模块的运算速度. (3) 应用改进方法实现了多目标跟踪.

1 基于灰色预测模型的检测区域设置

TLD 算法的检测模块是基于窗口扫描的, 该模块需要独立的对每一个窗口进行处理, 从而判断窗口内容是否包含目标. 这些窗口以很小的间隔遍布于整张图像, 并且窗口还需要有不同的尺寸以适应目标的尺度变化, 以一帧 VGA(640×480)的图像为例, 需要设置 5 万至 20 万个窗口. 然而这些窗口绝大部分分布于背景区域而并不包含目标内容, 因此造成了计算资源的极大浪费. 为此, 本文考虑采用灰色预测模型来预估目标位置^[14], 并以此位置设置一个较小的检测区域以缩小检测模块的检测范围.

1.1 灰色预测模型

灰色预测模型 $GM(1,1)$ 的一般形式为:

$\frac{dx^{(1)}}{dt} + ax^{(1)} = u$ 用于预测目标位置时, 分别对目标中心点的 x 和 y 坐标值进行预测即可估计目标的位置. 这里以 x 坐标的预测为例说明, y 坐标同理, 方法如下:

(1) 获得目标前 n 帧位置的 x 坐标序列:

$$X = \{x_1, x_2, \dots, x_n\}$$

(2) 对 X 序列作一阶累加, 生成一阶序列:

$$X^{(1)} = \{x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}\}$$

(3) 用 X 序列及 $X^{(1)}$ 序列构建矩阵 B 和 X_n :

$$B = \begin{bmatrix} -0.5(x_1^{(1)} + x_2^{(1)}) & 1 \\ -0.5(x_2^{(1)} + x_3^{(1)}) & 1 \\ \vdots & \vdots \\ -0.5(x_{n-1}^{(1)} + x_n^{(1)}) & 1 \end{bmatrix}, \quad X_n = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

(4) 用最小二乘法计算参数 a 和 u 的值:

$$\begin{bmatrix} a \\ u \end{bmatrix} = (B^T B)^{-1} B^T X_n$$

(5) 代入参数 a 、 u 得到预测模型的通解:

$$x_{t+1}^{(1)} = (x_1 - \frac{u}{a}) \cdot e^{-at} + \frac{u}{a}$$

并求得预测一阶序列:

$$\hat{X}^{(1)} = \{\hat{x}_1^{(1)}, \hat{x}_2^{(1)}, \dots, \hat{x}_{n+1}^{(1)}\}$$

(6) 最后运用公式 $\hat{x}_{t+1} = \hat{x}_{t+1}^{(1)} - \hat{x}_t^{(1)}$ 求得预测序列:

$$\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{n+1}\}$$

其中 \hat{x}_{n+1} 即为 x 当前的预测值.

然而, 灰色预测模型得到的预测结果可能存在误差, 因此需要进行检验与修正. 记 $Q = \{q_1, q_2, \dots, q_n\}$ 表示预测值 \hat{x} 与原值 x 的残差, 即 $q_i = x_i - \hat{x}_i$. 记序列 X 的平均值为 \bar{x} , 标准差为 S_x , 序列 Q 的平均值为 \bar{q} , 标准差为 S_q , 检验预测模型精度的两个指标为方差比 C 和小频误差 p , 其中 $C = \frac{S_q}{S_x}$, $p = P\{|q_i - \bar{q}| < 0.6745S_x\}$. 若 $C \leq 0.35$ 且 $p \geq 0.95$ [15] 则认为预测值是可靠的, 否则预测值需要修正, 修正方法为: $\hat{x}_{n+1}^{(1)} = \hat{x}_{n+1}^{(0)} + \hat{q}_{n+1}^{(1)}$,

$$\text{其中 } \hat{q}_{n+1}^{(1)} = (q_1 - \frac{u_q}{a_q}) \cdot e^{-a_q n} + \frac{u_q}{a_q}.$$

1.2 检测区域设置

运用灰色预测模型得到目标的预测位置后, 本文以预测位置为中心划定一个矩形区域, 此区域的长宽比与目标包围框一致, 而长与宽设置为包围框的 2 倍. 仅将与此区域有重叠的子窗口送入检测模块进行处理, 从而减少需要处理的子窗口数目. 如图 2 所示, 其中大矩形表示检测区域, 而 A、B 和 C 三个小矩形表示子窗口, 其中 A 和 B 与检测区域有重叠将被送入检测模块处理, 而 C 与检测区域没有重叠将不被处理.

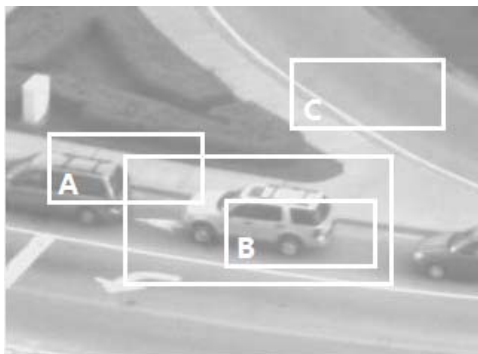


图 2 检测区域示意图

2 基于NCC距离的图像检索算法

TLD 算法维护着正、负两个样本集, 检测、学习模块都需要将获得的子窗口图像与两个样本集中的每一个样本依次进行 NCC 运算, 得到距离最近的样本, 然后进一步计算该窗口内容包含目标的概率. TLD 算法在运行过程中正、负样本集将随着学习算法的执行而不断的扩大, 而检测模块和学习模块的运行时间与样本集的大小成正相关, 因此导致 TLD 算法的效率随运行时间增长呈下降趋势. 为改善这种情况, 本文使用两种策略来维护正负样本集: 对于负样本集, 本文采用固定样本集大小的方法, 由于应用了 P-N 学习算

法 [8] 负样本集保存的是与目标较为相似的背景子窗口图像, 而背景会随目标运动而变化, 因此本文保存最晚加入的负样本而当负样本数超过固定值时移除最早加入的负样本; 而正样本集保存的是不同形态的目标窗口图像, 不能轻易移除. 另外本文在 B-树和 R-树 [16] 的基础上提出了一种基于 NCC 距离的图像检索方法同时用于正负样本, 该方法通过一种 m 阶平衡树来存储和管理样本集, 以提高运算效率. 该平衡树的主要特性如下:

- (1) 所有叶结点都在树的同一层上, 并且叶结点包含了样本集的所有样本.
- (2) 任一结点的孩子结点数介于 1 与 m 之间, 非叶结点的数据域与最早成为其孩子的结点相同.
- (3) NCC 距离最近的结点才能被放入同一父结点之下.

本算法在保证找到 NCC 距离最小的样本的基础上提高了运算的效率. 对于一个样本数为 n 的样本集, 原始查找算法的时间复杂度为 $O(n)$, 引入该检索算法后的时间复杂度为 $O(m \cdot \log_m n)$. 在本文的实验中 m 的值选取 4.

2.1 算法描述

基于 NCC 距离的图像检索算法主要涉及的算法包括图像的查找和插入算法, 算法描述如下:

查找算法

输入: 待查找的图像 E .

输出: 样本集中与 E 的 NCC 距离最近的图像 N ; 它们的 NCC 距离 d .

- (1) 将 N 设置为树的根结点.
- (2) 计算 N 的所有孩子结点与 E 的 NCC 距离. 并将 N 设置为 NCC 距离最近的一个孩子结点.
- (3) 如果 N 为叶子节点则执行步骤(4), 否则执行步骤(2).
- (4) 算法结束, 输出 N 以及 N 与 E 的 NCC 距离 d .

插入算法

输入: 待插入的图像 E .

输出: 插入新结点后的树.

- (1) 若根结点的孩子结点数为 0 (即样本集为空), 将 E 插入到根结点下, 执行步骤(5).
- (2) 执行查找算法得到样本集中与 E 的 NCC 距离最近的结点, 记为 N , 将 N 的父结点设置为待

插入结点, 记为 F .

- (3) 若 F 的孩子结点数小于 m , 将 E 插入到 F 结点下, 执行步骤(5); 若 F 的孩子结点数等于 m , 执行步骤(4).
- (4) 分裂 F 结点: 将 F 的孩子结点中与 F 的 NCC 距离最远的结点分裂出来, 插入 F 的父结点下. 若 F 的父节点的孩子结点数等于 m , 则以相同方法分裂直至根结点. 若根结点也被分裂则树增加一层.
- (5) 算法结束, 输出插入新结点后的树.

3 多目标跟踪

本文还将 TLD 算法应用到多目标的场景, 实现了 TLD 的多目标跟踪算法, 并且将前面所述的改进方法应用到其中, 如图 3 所示. 适用于多目标的各个模块的改变如下:

① 跟踪模块:

跟踪模块的算法与原算法相同, 需要中值流跟踪器^[9]逐个的对每个目标进行跟踪, 但由于跟踪时每个目标都处于同一帧中, 因此金字塔 LK 算法中的图像金字塔无需重复计算, 每帧计算一次后可由多个目标共用.

② 检测模块:

由于每个目标窗口的尺寸及宽高比各不相同, 无法共用一扫描窗口集合, 因此需要为每一个目标设置一个扫描窗口集合. 同样的, 由于每个目标的运动都可能不同, 需要为每个目标建立灰色预测系统并设置检测区域. 方差分类器只需要对每一帧图像计算一次积分图像, 多个目标可公用地计算窗口的方差, 但需要分别为每个目标保存方差的阈值.

③ 学习模块:

集合分类器需要为每一个目标建立正负样本集合以及后验概率表. 最近邻分类器需要为每个目标建立正样本集合, 但可以共用一个负样本集合, 如前文所述负样本集合中保存的主要是背景图像块, 各目标存在于同一帧中因此背景相同. 由于目标数的增加样本数量也增大, 使得图像检索方法对运算效率的提升更为显著.



图 3 多目标跟踪运行截图

4 对比实验及分析

在 TLD 算法的基础上, 本文对其检测和学习模块做了两部分的改进, 除此以外并无其他修改, 按照 Nebehay 的论文^[13]所述方法实现, 记本文算法为 ATLD 算法. 除了原始 TLD 算法以外, 本文还与 Gemignani 等人提出的 BTLD 算法^[10]以及 Guo 等人提出的 AA-TLD 算法^[11]进行了对比实验. 本文以 Visual Studio 2012 和 OpenCV 2.4.5 为平台编程实现了上述所有算法用于实验对比, 并且实验均在 CPU 为 Intel Core i5-3317U 1.70GHz, 内存为 4G 的计算机上进行.

4.1 实验数据

实验的数据包含了多种严苛的场景以测试算法的性能, 包含了光照变化、形态变化、尺寸变化、遮挡、消失等等. 实验用的视频是 6 段网上公开的常被用于跟踪测试的视频, 包括 Car、Mhyang、Car Chase、Jogging、David 和 David3. 表 1 中列出了 6 段视频的信息, 其中目标存在帧数表示目标在视频中未发生消失或被遮挡的情况的帧数. 图 4 是 6 段视频的截图, 其中白色框包围的区域表示目标区域. 6 段视频中 1、3 号视频的目标为车辆, 2、5 号视频的目标为人脸, 4、6 号视频的目标为人. 并且, 1、2、5 号视频包含了光照变化, 2、4、5 包含了尺寸变化, 1、4、5、6 包含了形态变化, 1、3、4、6 目标有消失或遮挡现象.

表 1 实验视频信息

序号	视频名	总帧数	目标存在帧数
1	Car	945	891
2	Mhyang	500	500
3	Car Chase	396	375
4	Jogging	307	293
5	David	761	761
6	David3	252	240

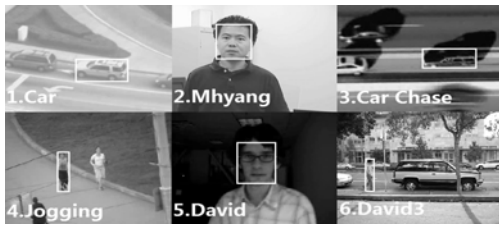


图 4 视频截图

4.2 实验结果与分析

4.2.1 跟踪精度比较

TLD 视频跟踪算法是一种优秀的长期跟踪算法,它具有较高的准确性和较强的鲁棒性,能够适用于多种严峻的场景. 实验中分别将 ATLD、BTLD、AA-TLD 以及原始 TLD 算法在 6 个视频上执行,进行对比实验. 表 2 中记录了各个算法在 6 段视频上执行时正确跟踪的帧数. 从表中数据可以看出四种算法都具有较高的正确率,能够较好的跟踪目标. 本文提出的 ATLD 算法的准确率与原始算法及 AA-TLD 算法接近,证明本文的改进没有影响算法的跟踪准确性,而 BTLD 算法由于对跟踪模块的改进,准确率略高于其他. 从表中数据还可以看出四种算法在处理无遮挡或消失的目标时的准确率要高于有遮挡的情况,这是因为当目标被部分遮挡及部分消失时可能无法正确的被识别为目标而导致的误差.

表 2 各算法正确跟踪帧数

序 号	目标存 在帧数	正确跟踪帧数/正确率			
		TLD[9]	ATLD	BTLD[10]	AA-TLD[11]
1	891	876/0.98	877/0.98	889/0.99	875/0.98
2	500	500/1.0	500/1.0	500/1.0	500/1.0

表 3 各算法各模块在 6 段视频上的平均运行时间

序 号	每帧平均运行时间/ms															
	TLD[9]				ATLD				BTLD[10]				AA-TLD[11]			
	跟 踪	检 测	学 习	总 时 间	跟 踪	检 测	学 习	总 时 间	跟 踪	检 测	学 习	总 时 间	跟 踪	检 测	学 习	总 时 间
1	16.4	552.6	1221.8	1694.0	16.8	218.7	464.8	673.4	1096.8	539.8	991.4	2327.9	17.8	145.1	710.3	818.3
2	16.6	292.0	610.6	863.4	16.7	134.4	416.8	549.9	406.2	310.6	543.4	1282.0	16.7	170.7	414.4	611.6
3	16.8	302.6	1321.7	1550.8	17.1	150.0	864.8	910.7	974.6	319.4	1347.1	2099.3	17.4	113.7	918.5	1065.5
4	20.2	470.1	1105.6	1551.8	20.1	201.7	762.5	895.4	739.2	468.7	1105.6	1954.8	20.1	306.6	765.0	1010.7
5	17.0	531.2	3665.8	4262.0	17.2	126.6	764.1	928.5	2143.9	514.3	3873.8	5537.7	17.0	214.2	3371.9	3743.5
6	17.2	351.7	1010.2	1381.6	16.9	202.7	752.9	943.1	705.5	321.8	1033.4	1741.3	17.1	314.8	772.9	1137.7

5 结语

TLD 算法是近年提出的一种高性能的长时间目标

3	375	369/0.98	368/0.98	372/0.99	369/0.98
4	293	285/0.97	285/0.97	292/0.99	283/0.96
5	761	761/1.0	761/1.0	761/1.0	761/1.0
6	240	233/0.97	234/0.97	239/0.99	233/0.97

4.2.2 运算时间比较

表 3 中给出了各算法处理不同视频时每帧图像的总平均处理时间并且给出了跟踪、检测、学习三个模块各自的运行时间. 由表中数据可知 ATLD 算法的运算效率最高其次是 AA-TLD 算法, BTLD 效率最低且低于原始 TLD 算法. 这是由于 BTLD 算法的跟踪模块中使用了 *Sr* 度量参数, 需要计算窗口图像的 NCC 距离, 这是非常耗时的, 可以从表中看出 BTLD 算法跟踪模块的运行时间与学习模块接近, 这是该算法提高精度付出的效率代价. AA-TLD 算法引入了粒子滤波算法用于预估目标的尺寸变化, 不需要设置多种尺度的扫描窗口, 较大程度上减少了需要处理的窗口数目, 从而提高了检测模块与学习模块的效率, 并且当目标窗口占一帧图像的面积越大效果越明显. 另外由表中数据可知, 学习模块(以及 BTLD 算法的跟踪模块)是整个 TLD 算法中耗时最久的模块, 其次是检测模块, 这是因为这些模块都需要计算图像的 NCC 距离, 这种计算是大量且耗时的, 而学习模块有大量的待学习正、负样本因此运行时间较检测模块更长. ATLD 算法提出的检测区域设置以及图像检索方法有效的减少了扫描窗口的数目并较大的提高了图像检索的效率, 因此, 如表中所示 ATLD 算法对检测模块和学习模块的效率都有明显提高, 并且对学习模块效率的提高尤为显著.

跟踪算法, 受到了广泛的关注. 该算法仅需较少的先验知识就能实现对目标的长时间跟踪, 能够较好的适

用于目标出现遮挡、消失、形态变化、尺寸变化等多种情况的场景. 本文在原始 TLD 算法的基础上提出了 ATLD 算法, 对原算法的检测模块和学习模块进行了改进, 提高了算法的运行速度, 并且将改进应用到了多目标跟踪的实现中. 实验的结果也显示, ATLD 算法的跟踪精度较原算法并未下降, 在可接受范围内, 而运行速度得到了较大提高. 下一步的研究考虑对算法的跟踪精度提出进一步的改进.

参考文献

- 1 Yilmaz A, Javed O, Shah M. Object tracking: A survey. *ACM Computing Surveys*, 2006, 38(4): 131-145.
- 2 Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. Menlo Park. *Proc. of International Joint Conference on Artificial Intelligence*. California: AAAI Press, 1981: 674-679.
- 3 Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003, 25(5): 564-577.
- 4 Comaniciu D, Meer P. Mean shift analysis and applications. *Proc. of the Seventh IEEE International Conference on Computer Vision*. IEEE. 1999, 2. 1197-1203.
- 5 Ristic B, Arulampalam S, Gordon N. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwood, MA: Artech House, 2004.
- 6 Kalal Z, Matas J, Mikolajczyk k. Online learning of robust object detectors during unstable tracking. *Proc. of the IEEE On-line Learning for Computer Vision Workshop*. New York: IEEE Press. 2009. 1417-1424.
- 7 Kalal Z, Mikolajczyk K, Matas J. Forward-backward error: automatic detection of tracking failures. *Proc. of International Conference on Pattern Recognition*. New York: IEEE Press. 2010. 23-26.
- 8 Kalal Z, Matas J, Mikolajczyk k. P-N learning: Bootstrapping binary classifiers by structural constraints. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. New York: IEEE Press. 2010. 49-56.
- 9 Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012, 34(7): 1409-1422.
- 10 Gemignani G, Choi W, Ferone A, et al. A Bayesian approach to tracking learning detection. *Image Analysis and Processing-ICIAP 2013*. Springer Berlin Heidelberg. 2013. 803-812.
- 11 Guo P, Li X, Ding S, et al. Adaptive and accelerated tracking-learning-detection. *ISPD 2013-Fifth International Symposium on Photoelectronic Detection and Imaging*. International Society for Optics and Photonics. 2013. 89082H-89082H-10.
- 12 龚小彪. 基于 TLD 框架的目标跟踪算法研究[硕士学位论文]. 成都: 西南交通大学, 2014.
- 13 Nebehay G. Robust object tracking based on tracking-learning-detection[Master thesis]. Vienna: Faculty of Informatics, Vienna University of Technology, 2012.
- 14 李连, 蒋文涛, 张庆敏. 传感器网络中基于灰色预测的目标跟踪算法. *软件学报*, 2012, 23(1): 149-158.
- 15 Liu S, Dang Y, Fang Z. Grey system theory and its application. Science, Beijing, 2004: 150-195.
- 16 Guttman A. R-tree: A dynamic index structure for spatial searching. *Proc. of the 1984 ACM SIGMOD International Conference on Management of Data*, 1984, 14(2): 47-57.