

结合 Boosting 方法与 SVM 的多核学习跟踪算法

曾礼灵¹, 李朝锋^{1,2}

ZENG Liling¹, LI Chaofeng^{1,2}

1. 江南大学 物联网工程学院, 江苏 无锡 214122

2. 江南大学 轻工过程先进控制教育部重点实验室, 江苏 无锡 214122

1. School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China

2. Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education, Jiangnan University, Wuxi, Jiangsu 214122, China

ZENG Liling, LI Chaofeng. Multiple-kernel learning based object tracking algorithm with Boosting and SVM. Computer Engineering and Applications, 2000, 0(0): 1-000.

Abstract: As traditional tracking algorithms fail to track target stably due to the external environment and the target motion caused deformation, a robust multiple kernel learning based algorithm is proposed. By introducing the Boosting method into the multiple kernel learning framework, building a pool of weak classifiers trained with complementary feature set and complementary kernel function set needs less samples comparing to the traditional Multiple-kernel Learning algorithms. Thus a multiple-kernel strong classifier is constructed by combining several weak classifiers selected from the weak classifier pool, which can correctly differentiate the target and background from the candidate patches even when the target is under notable occlusion and background clutters. Results of test on different video sequences show that when the tracked object is in complex environment, the proposed algorithm has higher tracking accuracy compared with OAB algorithm which similarly using the Boosting method and the LOT algorithm which has a high tracking accuracy.

Key words: multiple-kernel learning; object tracking; Boosting; complex environment

摘 要: 针对传统算法在外界环境及目标运动导致外形变化的影响下跟踪效果不稳定的问题, 提出一种鲁棒的多核学习跟踪算法, 将 Boosting 提升方法引入到多核学习框架中, 用比传统多核学习算法更少的样本训练, 构建出基于互补性特征集和核函数集的弱分类器池, 从中将多个单核的弱分类器组合出一个多核的强分类器, 从而能够在出现的较强背景干扰、目标被遮挡的情况下仍能正确地对候选图块中的背景和目标进行分类。对不同视频序列的测试结果表明, 与同样采用 Boosting 方法的 OAB 算法及近年跟踪精度高的 LOT 算法相比, 能够在复杂环境下更准确地跟踪到目标。

关键词: 多核学习; 目标跟踪; 提升方法; 复杂环境

文献标志码: A 中图分类号: TP391 doi: 10.3778/j.issn.1002-8331.1703-0004

1 引言

目标跟踪是机器视觉领域研究的重要问题之一, 近年来已有大量的相关跟踪算法及其改进^[1-2]。但在

跟踪过程中由于目标外观变化、外界遮挡等因素的影响, 进行鲁棒的目标跟踪仍然面对很大的挑战。根据算法与目标检测的关系的不同, 近年来提出的跟踪算法可分为生成式与判别式 2 种。判别式跟踪

作者简介: 曾礼灵(1991—), 男, 硕士研究生, 研究领域为机器学习, 模式识别; 李朝锋(1971—), 男, 博士研究生, 教授, 研究领域为图像处理, 模式识别, E-mail: 6141921013@vip.jiangnan.edu.cn。

万方数据

方法将跟踪问题看作前景与背景的二分类问题,通过学习分类器,在当前帧的候选区域中,用分类的方法找出与背景区分度最大的前景区域。^[2]其代表算法有 OAB (online Adaboost)^[3]、TLD (Tracking-Learning-Detection)^[4]、LOT(locally object tracker)^[5]等。

TLD 算法采用跟踪和检测相结合的策略进行跟踪,首先在第一帧图像上使用像素比较的方法得到目标的特征,将特征输入到随机森林机器学习算法中训练得到一个目标检测器。跟踪器利用目标的运动信息来跟踪目标,这个过程基于 Lucas-Kanade 光流法。跟踪器和检测器同时运行,估计目标的位置。但是在目标被遮挡后,该算法易出现跟踪漂移的问题^[6],导致目标丢失。SVT (Support Vector Tracking) 算法成功地将事先用大量的样本训练好单核支持向量机 (SVM, support-vector machine) 用于车辆的跟踪,并取得了较好的效果^[7]。由于 SVM 在目标跟踪方面已取得较好的效果,近年来有不少算法将其用于处理目标跟踪的问题。Grabner 等人提出将 SVM 与 Boosting 方法结合的 OAB 算法,该算法将 Boosting 方法运于到特征的筛选和组合过程中,通过在线学习、在线选择特征实现对目标的跟踪,而无需像文献[7]那样事先用大量样本训练好 SVM 才能用于跟踪。然而由于该算法用于组合的弱分类器个数为固定值,最终构建的强分类器分类能力受到限制^[3]。Oron 等提出的 LOT 算法将局部无序匹配 (LOM, Locally Orderless Matching) 方法引入到目标跟踪算法中,其具体做法是利用粒子滤波法选取得到候选粒子,再使用 LOM 计算各粒子的置信度。该算法在计算置信度时使用了超像素 (superpixel) 方法,^[5]虽然其跟踪精度高、鲁棒性好,但其处理过程较为复杂,计算量非常大,处理跟踪问题时效率不高。

针对上述算法存在的问题,本文提出一种基于多核学习框架^[8-9]的跟踪算法,根据 Boosting 的加权分类思想,利用不同特征的样本集和核函数集构建出具有差异性的单核弱分类器池,再通过 Boosting 方法对单核弱分类器进行筛选,组合出分类能力更强的多核分类器,将所得强的分类器用于对目标状态的预测,从而实现稳定、准确的跟踪。

2 问题分析

如上所述,跟踪问题可视为前景与背景的二分类问题,判别式跟踪算法的跟踪效果很大程度上取决于算法分类能力的强弱。为此,将多核学习引入到跟踪算法中,根据集成学习的思想,通过组合多个分类器^[10]获得分类能力更强的分类器,从而实现更精确的跟踪。

在分类问题中,选取合适的数据表示方式直接影响到分类能力。SVM 巧妙地采用了核函数 $K(x, xi)$ 来表示数据,这里的核函数 $K(\cdot, \cdot)$ 是在希伯尔空间内正定的函数。虽然 SVM 在分类问题中被认为是强分类器,在机器视觉方面有出色的表现^[7],但使用单个 SVM 分类器分类能力终究有限,能正确分类的数据类型也受到限制。文献[8-9]将 SVM 与多核学习框架相结合用于数据分类,实验结果表明,多核学习能增强 SVM 决策函数的解释能力,提高其分类能力。文献[11]将多核分类器用于的大数据分类问题上,结果表明多核分类器分类能力明显优于任何一个单核分类器。

在生成集成成员 SVM 方面, Bagging、Boosting 等是研究和使用的最多、影响最大的重要技术。其中 Bagging 对稳定的学习算法效果不明显,有时甚至使预测精确度降低^[12]; Boosting 方法则使用了确定性方法保证训练集中含有更难分类的样本以形成分类^[13],大量的实验也证明, Boosting 不但可以提高学习精度,而且不易过配^[14]。

3 多核学习目标跟踪算法的实现

3.1 样本的选取

在本文算法中有训练样本和候选样本 2 种样本。训练样本用于训练分类器,包括正样本和负样本。样本块 i 的状态用 p_i 表示,记为 $p_i = (x, y, w, h)$, 分别为该样本块的中心横坐标、中心纵坐标、宽、高。

正样本 p^+ 为目标图块,只在目标周围很小的范围内选取,其选取规则为:

$$p^+ = \{p_i^+ | 0 \leq d(p_i^+ - t) < r_+\}_{i=1}^{n_+} \quad (1)$$

式中的 t 表示目标状态, $d(p_i^+ - t)$ 为样本 p_i^+ 与目标 t 的欧氏距离, n_+ 为正样本的数量。

负样本 p^- 为背景图块,在目标周围的背景范围内选取,类似地,其选取规则为:

$$p^- = \{p_i^- | r_{\beta-} \leq d(p_i^- - t) < r_{\alpha-}\}_{i=1}^{n_-} \quad (2)$$

候选样本在预测当前帧目标所在位置时选取。针对目标可能运动剧烈导致跟踪失败的问题,与文献[15]“由粗到精”确定目标位置的思想类似,本文采用了两步采样方法:首先,在较大范围 d_1 内进行一次“粗采样”,在上一帧目标位置周围随机撒点,得到的 i 个候选块,从中选出分类器输出响应最高的候选样本所在位置。然后在此位置周围更小的范围 d_2 内进行第二次采样,得到 j 个候选样本。

其中, d_1 由最近 C 帧目标位置变化情况决定,设当前为第 $f+1$ 帧,则 d_1 由以下公式确定:

$$d_1 = d_f * \max(\text{std}(\{h_p\}_{f-C}^f), \theta) \quad (3)$$

式中的 h_p 为本文算法计算的目标历史状态, d_f 、 C 均为常数, θ 为常数向量,用于防止目标在多帧静止后突然运动,算得 d_1 为0。 $\text{std}(\{h_p\}_{f-C}^f)$ 表示最近 C 帧目标位置的标准差。

第二次采样的范围 d_2 由以下公式计算:

$$d_2 = d_f * d_1 \quad (4)$$

目标的状态 p 由第二次采样中所有样本 p_i 的状态决定,其计算公式为:

$$p = \frac{\sum_{i=1}^j p_i F(p_i)}{\sum_{i=1}^j F(p_i)} \quad (5)$$

式中的函数 $F(x)$ 为由多个单核 SVM 分类器组合而成的多核分类器的决策函数, $F(p_i)$ 为样本 p_i 的输入到该分类器中得到的决策值。

3.2 SVM 与多核学习框架

本文参照文献[8-9]的多核学习框架,将其用于构建一个多核的二分类器 $F(x)$,用它来对未分类的样本(即候选样本)进行分类。对于给定的训练样本集 $\{x_i, y_i\}_{i=1}^D$,其中 x_i 表示第 i 个样本的特征, $y_i = \{\pm 1\}$ 为第 i 个样本对应的标签。设 $\{K_m\}_{m=1}^M$ 为核的集合,多个核组合而成的分类器可表示为:

$$K(x, x_i) = \sum_{m=1}^M \beta_m K_m(x, x_i) \quad (6)$$

公式中的 β_m 为第 m 个核的权重, $\beta_m > 0$ 且满足 $\sum_{m=1}^M \beta_m = 1$; K_m 为第 m 个核函数,且可以是类型相同、参数不同,或特征不同的核。这样,算法主要任务就变成了计算出各个核的权重 $\{\beta_m\}_{m=1}^M$ 。最终得到的强分类器可表示为:

$$F(x) = \sum_{i=1}^D \alpha_i y_i \sum_{m=1}^M \beta_m K_m(x, x_i) + b \quad (7)$$

公式中的 $\{\alpha_i\}_{i=1}^D$ 和 b 分别是拉格朗日乘子、SVM 的标准偏置。

虽然文献[8]的多核学习框架能在数据分类问

题中取得良好的效果,但是由于以下几个因素的影响,其并不能直接应用于目标跟踪。首先,参数优化过程需要大量的训练样本才能收敛,而样本太多会降低跟踪的效率;其次,在做分类识别时,经过训练后的核权重是固定的,而在目标跟踪过程中,由于前景变化和背景干扰,核的权重需要自适应地进行调整。现有的多核学习提升方法着重于加强在大量样本的基础上提高识别分类的精度,而未加强对特定目标的定位,因此在跟踪上未能做到高效。本文提出的是基于特征选择的多核 Boosting 提升算法,从而减少了计算量,并提高了跟踪精度。

对于输入样本 x ,第 m 个 SVM 核的响应输出用 $K_m(x)$ 表示, $K_m(x)$ 为样本 x 与其他样本 $\{x_i, y_i\}_{i=1}^D$ 的核内积,记为

$$K_m(x) = [K_m(x, x_1), K_m(x, x_2), \dots, K_m(x, x_D)]^T \quad (8)$$

则公式(6)可改写为

$$F(x) = \sum_{m=1}^M \beta_m \sum_{i=1}^D \alpha_i y_i K_m(x, x_i) + b = \sum_{m=1}^M \beta_m (K_m(x)^T \alpha + b_m) \quad (9)$$

式中的 $\alpha = [\alpha_1 y_1, \alpha_2 y_2, \dots, \alpha_D y_D]$, $b = \sum_{m=1}^M b_m$ 。这样,标准多核学习公式就变成了 M 个独立的 SVM 分类器的线性组合形式: $K_m(x)\alpha + b_m$ 。在训练 M 个独立的 SVM 分类器时,每个分类器均使用不同的核参数 $\{\alpha_m, \beta_m\}$,训练完成后再计算其权重 $\{\beta_m\}_{m=1}^M$ 。令 $h_m(x)$ 为 SVM 分类器的响应,则 $h_m(x) = K_m(x)\alpha_m + b_m$,表示每个核都有一个独立的系数 α ,而标准的多核学习算法[8]中所有核的系数均为同一系数,这也是本算法与其算法的不同之处。

在计算 $\{\beta_m\}_{m=1}^M$ 时,本文采用了 Boosting 提升方法(详见 3.3 节),则生成的强分类器的决策函数 $F(x)$ 表示为:

$$F(x) = \sum_{i=1}^L \beta_i h_i(x) \quad (10)$$

式中的 L 为 Boosting 筛选单核分类器时的迭代次数。因筛选过程中无需进行复杂的优化计算,从而提高了效率和精度。

如图 1 所示,为从正样本(目标图块)、负样本(背景图块)中提取的特征集合 $\{f_1, f_2, \dots, f_N\}$,这些特征分别输入到 M 个不同的单核 SVM 分类器中训练,每种特征和每个核视为一个组合,这样就得到了 $M \times N$ 个分类器。

3.3 Boosting 方法筛选弱分类器

训练所得的 $M \times N$ 个单核分类器并不会全部整合到强分类器中, 经 Boosting 方法筛选后, 只有不多于 L 个分类器会被选中。

在筛选前, 需要计算所有单核 SVM 分类器的分类误差 ε , 记为:

$$\varepsilon = \frac{\sum_{i=1}^D w(i) \cdot |h(x_i)| \cdot U(-y_i h(x_i))}{\sum_{i=1}^D w(i) \cdot |h(x_i)|} \quad (11)$$

式中的 $U(x)$ 为单位阶跃函数, 输入为正输出为 1, 否则输出为 0。 $w(i)$ 为样本的权重, $h(x_i)$ 为 SVM 对样本 x_i 的分类结果, 取值范围为 1 或 -1。

在迭代的初始阶段, 为每个样本分配的权重 $w_l(i)$ 是一样的, 均为 $w_l(i) = 1/D$, D 为样本总数。迭代开始, 在每次迭代过程中, 利用公式(11)计算各分类器误差, 选取一个训练误差最小的分类器, 通过下式计算该分类器的权重 β_l :

$$\beta_l = \frac{1}{2} \log \frac{1 - \varepsilon_l}{\varepsilon_l} \quad (12)$$

如果权重 β_l 小于 0, 则其误差 $\varepsilon_l > 0.5$, 表示此分类器的分类效果比随机分类的效果还差, 不能对样本进行正确的分类预测, 则立即停止此次循环; 否则, 将选中的分类器被选中。在进行下一次迭代之前, 为上一次被错误分类的样本分配以更大的权重:

$$w_{l+1}(i) = \frac{w_l(i)}{z_l} e^{-\beta_l y_i h_l(x)} \quad (13)$$

式中的 y_i 为分类器 $h_l(x)$ 的分类结果 (1 或 -1)。进行下一次迭代, 将新的样本权值代入到公式(11)中重新计算各分类器误差, 如此循环直至 L 次后结束。

最后, 将所有被选中的分类器组合起来, 得到强分类器决策函数 $F(x)$:

$$F(x) = \sum_{l=1}^L \beta_l h_l(x) \quad (14)$$

值得注意的是, 本文中提出的算法不同于文献[16]的集成跟踪算法那样, 将多个同类的分类器在同一个特征空间进行组合。虽然用于组合的分类器越多, 效果可能会越好, 但由于如果特征空间受到限制, 弱分类器数目再多, 最终集成所得的强分类器的分类能力也是有限的。而本文算法训练弱 SVM 分类器时使用了多种特征, 故所得的弱分类器亦是在不同的特征空间中训练所得, 从而保证了分类器之间的差异性。为了降低运算成本, 本文算法并不引入新的分类器, 也不淘汰旧的分类器, 而是在每次训练时只用给定范围内的参数来训练, 在使用 Boosting 提升方法获得强分类器过程中, 从上述训

练过程中得到的 $M \times N$ 个弱分类器中挑选出分类效果最好的 n 个进行组合, 从而得到新的强分类器。

在跟踪过程中, 由于目标自身变化、外界环境变化等原因, 某些分类器可能无法继续正确地对候选块分类。这时就需要对分类器重新训练, 并重新挑选一组分类器用于组合出新的强分类器。为了达到更高的分类精度, 每次更新时, 组合的分类器的数目也会自适应地变动。文献[16]的集成跟踪算法 (ensemble tracking) 虽然也使用多种特征训练弱分类器, 但其用于组合的分类器数目却是固定的, 经过 Boosting 提升得到的强分类器分类能力并未得到很大的提升, 因此未能实现更好的跟踪效果。

3.4 本文算法主要步骤

第一帧为算法的初始化阶段, 人工给定目标的初始状态 (坐标及大小)。根据初始状态提取目标、背景样本的 3 种特征, 将所得的 N 个特征向量输入到 M 个不同核参数或核类型的 SVM 分类器中进行训练, 得到 $M \times N$ 个单核分类器。利用 Boosting 方法, 筛选出 L 个分类器并整合成一个强分类器。

进入第 2 帧或后续帧中, 在上一帧目标所在位置经过两步采样选取候选样本并提取特征, 将特征输入到所得强分类器中, 预测出目标在当前帧中的位置。

达到一定帧数后, 利用新的目标产生正负样本重新训练、重新组合出新的强分类器, 即更新分类器。否则进入下一帧。

如此循环迭代, 直至完成整个序列的跟踪。

4 仿真结果和分析

为了验证本文算法的性能及时间复杂度, 本文在版本号为 R2012b 的 Matlab 软件上对 4 个算法进行测试对比, 实验所使用的电脑操作系统为 32 位 Win7 旗舰版, CPU 处理器型号为 Intel(R) Core(TM) i3-2130, 主频为 3.4GHz, 内存为 4GB。本文实验从两个方面进行, 分别为单核与多种核的对比、多核与其他算法的对比。SVM 核类型采用了径向基 (Radial Basis Function) 核、多项式 (Polynomial) 核、Sigmoid 核, 共 3 种。

在特征选取方面, 本文选取了 RGB 直方图、方向梯度直方图 HoG^[17]、SIFT^[18]3 种不同类型的特征。

4.1 实验数据集与评价指标

实验数据为来自视觉跟踪公开库^[1]中具有代表性的 3 个序列: 序列 Bolt 分辨率为 640 × 360, 其主要挑战为遮挡、非刚体形变; 序列 Basketball 分辨率为 576 × 432, 其主要挑战为光照变化、遮挡、非刚体形变、背景杂波干扰; 序列 Skater 分辨率为 320 × 240, 其主要挑战为目标尺度变化、角度旋转。这些测试序列用于测试本文算法在跟踪目标遭受背景杂波干扰、光照变化、物体遮挡时的跟踪精度和性能。

本文采用文献[1]提供的 benchmark 对包括本文在内的 4 个算法进行了比较, 跟踪性能采用坐标精度、准确率 2 个指标来衡量。其中坐标精度为坐标误差精度, 即跟踪算法测得的目标中心坐标与目标中心的真实坐标之间的欧氏距离的平均值, 用来衡量跟踪算法的跟踪精度, 误差精度越小, 表明算法跟踪得越准确。

准确率用来衡量算法的稳定性, 每帧的准确率计算公式为 $S_t = \frac{|r_t \cap r_a|}{|r_t \cup r_a|}$ 。式中的 r_t 为跟踪算法预测的目标区域, r_a 为目标真实的区域, $|r|$ 表示区域 r 内包含的像素点的个数。若 S_t 大于覆盖率阈值 t_o , 则视为成功跟踪到目标, 否则为失败。所有成功跟踪到的帧数与总帧数的比值即为准确率。由此可绘制出准确率曲线, 各算法准确率曲线的 AUC (Area under Curve) 数值即为各算法的最终得分, 得分越高, 表明算法的稳定性越好。

本文各算法的 AUC 值均为在 OPE (one-pass evaluation) ^[1]方式下测得的。

4.2 结果和分析

单种核与多种核的对比实验采用了相同的参数, 对文献[1]库中的 55 个序列进行测试, 并计算出各种情况下跟踪的平均坐标误差。实验结果如表 1 所示。

表 1 单核与多核的坐标精度对比 单位: 像素

核类型	径向基核	多项式核	Sigmoid 核	多核
平均误差	132.6	128.3	162.8	116.4

由表 1 可知, 根据集成学习的思想^[10], 用于组合的弱分类器差异性越高, 组成的强分类器分类效果越好, 而多种核的组合差异性比单种核的差异性高, 故将多种核进行组合的效果比单种核的组合效果好。

在与其他经典算法对比的实验中, 选取了 OAB、TLD、LOT 三种算法用于对比。其中, TLD 算法为被广泛使用的经典跟踪算法之一; OAB 算法与本文一样, 都采用 Boosting 提升方法组合强分类器、采用新样本更新分类器; LOT 算法为 2015 年发表的具有高精度的算法之一。

表 2 跟踪算法的坐标精度对比 单位: 像素

序列	TLD	OAB	LOT	本文算法
Bolt	88.4	258.1	13.4	8.9
Basketball	66.4	161.4	9.3	14.1
Skater	20.2	26.0	19.5	16.2
平均	58.3	148.5	14.1	13.1

4 个算法的跟踪坐标精度如表 2 所示, 加粗的数字表示误差最小, 即跟踪精度最高。TLD 算法的结果中有部分数据为 NaN, 为无效数据, 为进行比较, 将结果为 NaN 的数据用前一帧的数据代替。由表 2 可以看出, 本文算法跟踪效果比较稳定, 在 Bolt、Skater 序列上测试均取得最小的坐标误差; 在 Basketball 序列上测试坐标精度低于 LOT 算法。

在图 2 中的 Basketball 序列的第 490 帧前后, 被跟踪目标周围出现部分遮挡及背景干扰的情况, TLD 算法开始逐渐偏离目标, 跟踪误差越来越大; OAB 算法则在跟踪开始后不久就因目标被部分遮挡, 分类器无法正确对背景和目標分类, 因此逐渐偏离了目标, 直到此序列结束都未能再次跟踪到原来的目标。本文算法虽然受到一定的影响, 但因采用“由粗到精”的样本选取方法限制了样本选取范围, 没有将周围与目标相似的运动目标错误地识别为被跟踪目标, 故依然能够持续地对目标进行跟踪, 预测出目标的位置。

图 3 所示的 Bolt 序列中, 有 2 处存在较大干扰, 一是第 20 帧前后背景中出现与目标颜色相近的区域, 二是第 140 帧前后目标快速运动产生运动模糊,

所有算法均因此受到影响, OAB、TLD 算法分别在 20、140 帧左右, 因分类器未能正确识别目标, 开始丢失目标。本文算法虽然也因背景干扰出现了偏移现象, 但除颜色特征外, 还采用了其他特征对目标进行描述, 故仍能再次识别到目标, 虽受到影响, 却仍能以较低的误差跟踪到目标。

图 4 所示的 Skater 序列中, 因目标面积较大, 其运动幅度与其面积相比较小, 所有算法均未丢失目标, 但因运动发生尺度变化、角度旋转, 各算法还是受到了一定的影响, 其中 OAB、LOT、TLD 算法跟踪误差波动较大, 而本文算法在选取候选样本时选取的范围没有盲目扩大, 故跟踪误差波动要小得多, 最终的坐标误差最小。

由图 5 的跟踪算法准确率曲线可知, 本文算法 AUC 值为 0.596, 性能优于 TLD、OAB 算法。LOT 算法性能与本文算法接近, 但仍小于本文算法。受样本选取方法的影响, 在目标运动幅度较小时, 本文算法对目标的跟踪反而出现小幅的波动, 所以 LOT 算法在覆盖率阈值 t_0 小于 0.3 时, 准确率略高于本文算法, 这也是本文存在的尚需改进之处。不过在覆盖率阈值 t_0 大于 0.3 的情况下, 本文算法准确率都是高于 LOT 算法的。

4 个算法的速度对比如表 3 所示, 表格中数据为算法运行速度, 数值越大则表示时间复杂度越低。由表 3 及上述跟踪结果可知, 与本文性能接近的 LOT 算法处理过程较为复杂, 计算量非常大, 虽然也表现出很高的精准度, 但速度慢于本文算法。本文算法因在集成多个 SVM 弱分类器时增加了运算量, 比 OAB、TLD 算法慢, 但因此获得了精度上的提高。

表 3 跟踪算法的时间复杂度对比 单位: 帧/秒

序列	TLD	OAB	LOT	本文算法
Bolt	25.88	1.97	0.27	0.56
Basketball	15.16	3.42	0.27	0.57
Skater	35.69	3.31	0.35	0.59
平均	25.58	2.9	0.3	0.57

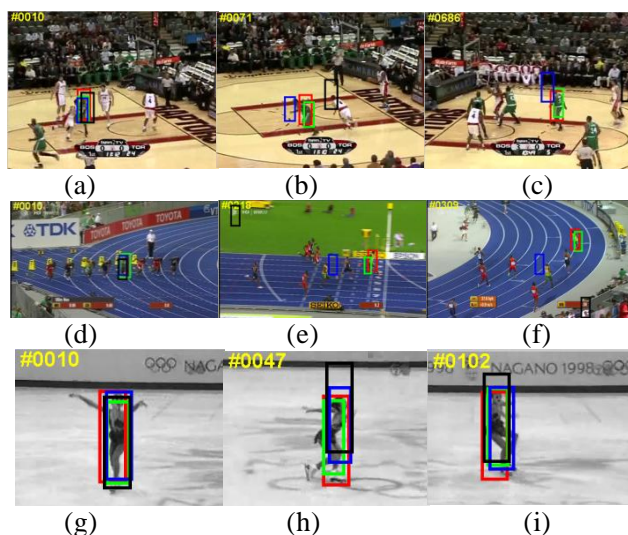


图 1 不同序列的跟踪效果截图
(a) (b) (c) 为 Basketball 序列实验截图, (d) (e) (f) 为 Bolt 序列实验截图, (g) (h) (i) 为 Skater 序列实验截图

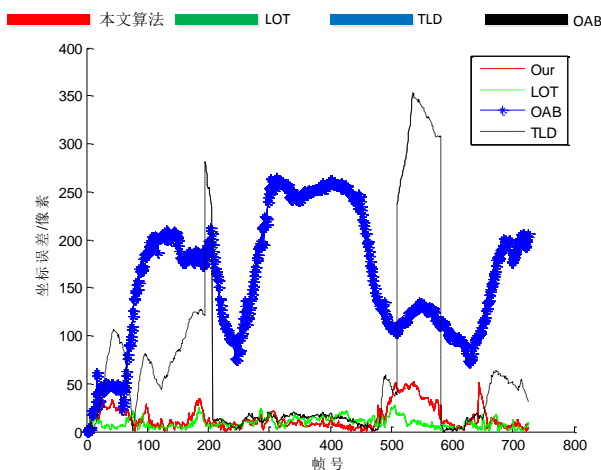


图 2 Basketball 序列逐帧跟踪误差

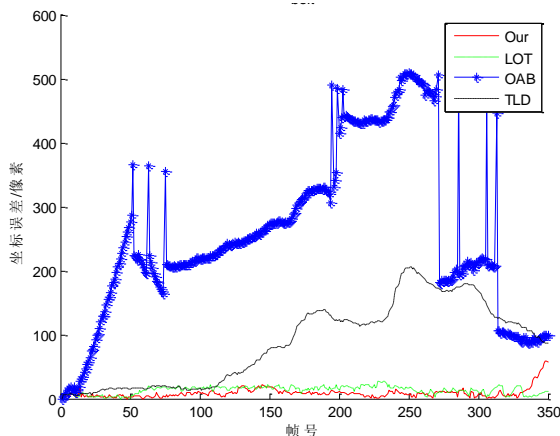


图 3 Bolt 序列逐帧跟踪误差

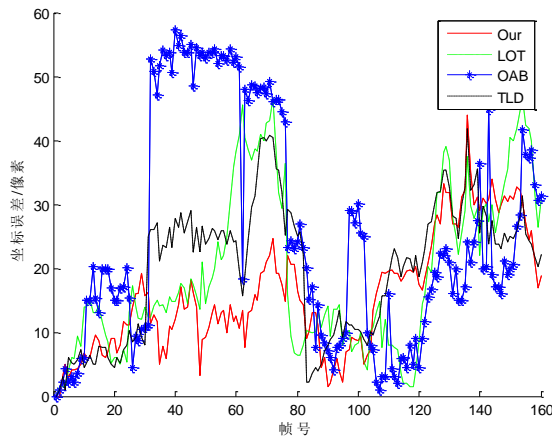


图4 Skater 序列逐帧跟踪误差

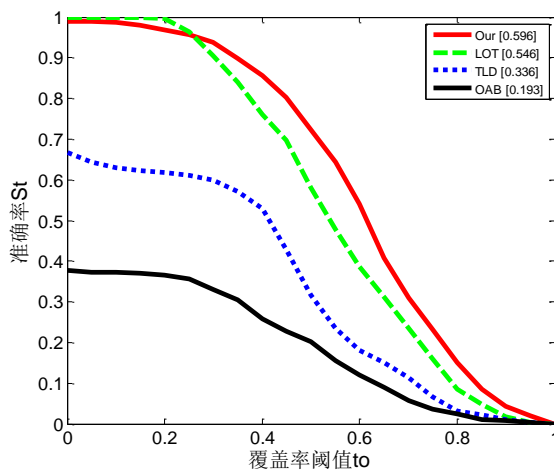


图5 跟踪算法准确率曲线

5 结束语

本文研究了多核学习在目标跟踪中的应用, 通过在存在外界环境影响及目标自身发生形变的序列上的跟踪实验, 测试了本文算法在复杂环境下跟踪效果的性能和精度。本文算法将多核学习引入到跟踪问题中, 通过多种特征和多种核的组合, 保证了合成的分类器具有更好的分类能力, 从而能够在有干扰的情况下正确区分目标与背景, 实现稳定、精确的跟踪。

参考文献

- [1] Yi Wu, Jongwoo Lim, Ming-Hsuan Yang. Online Object Tracking: A Benchmark[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2013: 2411-2418.
- [2] 尹宏鹏, 陈波, 柴毅, 等. 基于视觉的目标检测与跟踪综述[J]. 自动化学报, 2016, 42(10): 1466-1489.
- [3] H. Grabner, H. Bischof. On-line boosting and vision[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2006: 260-267.
- [4] Z. Kalal, K. Mikolajczyk, J. Matas. Tracking-Learning-Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2012, 34(7): 1409-22..
- [5] S. Oron, A. Bar-Hillel, D. Levi, et al. Locally orderless tracking[J]. International Journal of Computer Vision, 2015, 111(2): 213-228.
- [6] 杨萌, 杨阳, 王艺, 等. 一种改进的 TLD 算法[J]. 传感器与微系统, 2014: 130-133.
- [7] S. Avidan. Support vector tracking[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2004, 26: 1064-1072.
- [8] F. Bach, G. Lanckriet, M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm[C]//International Conference on Machine Learning, 2004: 6-13.
- [9] A. Rakotomamonjy, F. Bach, S. Canu, et al. Simplemkl[J]Journal of Machine Learning Research, 2008, 9(3): 2491-2521.
- [10] 卢明, 刘黎辉, 吴亮红. 多核支持向量数据描述分类方法研究[J]. 计算机工程与应用, 2016, 52(18): 68-73.
- [11] 夏利民, 时晓亭. 基于关键帧的复杂人体行为识别[J]. 模式识别与人工智能, 2016, 29(2): 154-162.
- [12] 沈兴华, 周志华, 吴建鑫, 等. Boosting 和 Bagging 综述[J]. 计算机工程与应用, 2000, 36(12): 31-32.
- [13] 王晓丹, 高晓峰, 姚旭, 等. SVM 集成研究与应用[J]. 空军工程大学学报·自然科学版, 2012, 13(2): 84-89.
- [14] 于玲, 吴铁军. 集成学习: Boosting 算法综述[J]. 模式识别与人工智能, 2004, 17(1): 52-59.
- [15] 陈云, 石松等. 基于 SVM 混合集成的信用风险评估模型[J]. 计算机工程与应用, 2016, 52(4): 115-120.
- [16] S. Avidan. Ensemble Tracking[C]//IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005: 494-501. [2007]
- [17] N. Dalal, B. Triggs. Histograms of oriented gradients for human detection[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition, 2005: 886-893.
- [18] D. Lowe. Object recognition from local scale-invariant features[C]//IEEE International Conference on Computer Vision, 1999: 1150-1157.