

基于簇相似的多分类器目标跟踪算法

李 康,何发智,潘一腾,孙 航
(武汉大学计算机学院,湖北武汉 430072)

摘 要: 由于跟踪过程中目标和背景的变化,传统的单分类器跟踪算法学习到大量的非目标信息而导致跟踪精度降低. 针对该问题,本文提出使用树形结构保存历史分类器. 在每一帧,根据树中路径距离选择分类器集对测试样本分类. 提出了一种基于簇相似性比较的分类算法. 通过建立以方差为尺度的特征空间,比较测试样本到簇中心的距离计算相似度,快速计算出目标样本. 实验表明本算法能够在复杂条件下实现对目标的鲁棒跟踪.

关键词: 目标跟踪; 多分类器; 在线学习; 类哈尔特征

中图分类号: TP391. 41 **文献标识码:** A **文章编号:** 0372-2112 (2016)04-0821-05

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2016.04.010

Multi-Classifer Object Tracking Based on Cluster Similarity

LI Kang, HE Fa-zhi, PAN Yi-teng, SUN Hang

(School of Computer Science, Wuhan University, Wuhan, Hubei 430072, China)

Abstract: Due to the changes of target and background during tracking, traditional single classifier tracking algorithms learn a lot of non-target information which result in the decrease of tracking accuracy. In this paper, we propose to use tree structure to save former classifiers as a set. In each frame, a subset of classifiers are chosen according to the path in the tree to classify test samples. We propose a classification algorithm based on cluster similarity comparison. A normalized feature space is established according to the variance of the cluster. The target in a new frame could be got by computing the distance between test samples and the center of the cluster. Experiments show that our algorithm could achieve the goal of robust tracking under complicated conditions.

Key words: object tracking; multiple classifiers; online learning; Haar-like feature

1 引言

目标跟踪是计算机视觉中最为活跃的领域之一,其研究成果被广泛用于运动分析,实时监控和人机交互等领域^[1,2]. 近年来有大量的目标跟踪算法被开发出来,但是由于目标本身的变化,目标信息不足,以及光照变化和遮挡变化等外部因素的影响,开发出鲁棒性的目标跟踪算法仍旧是一个难题.

经典的跟踪系统由外观模型,运动模型和搜索策略三部分组成. 其中,外观模型最为重要,它直接影响了整个跟踪系统的算法和结构. 与静态外观模型相比,自适应外观模型算法^[3,4]在跟踪过程中不断进行更新,适应目标新的变化,具有更好的鲁棒性.

根据外观模型的不同,自适应模型跟踪算法可以分为生成模型^[5,6]和判别模型^[7,8]两类. 基于分类的判别模

型算法从目标和背景中采集正样本和负样本,对这些样本提取特征后训练分类器,最后使用分类器对目标可能存在区域的采样样本进行分类得到目标的位置. 文献[8]提出了一种在线 Boosting 算法,取得了良好的效果. 但是由于 Boosting 算法在每一帧都必须迭代地对弱分类器进行选择,极大的降低了跟踪速度. 近年来,压缩感知理论被广泛用于包括目标跟踪等图像处理的各个领域. 文献[7]根据信号的稀疏特性,使用随机采样的方式从图像中快速提取目标的类哈尔特征,并用这些相互独立的特征组合训练贝叶斯分类器,然后使用滑动窗口在目标可能存在的区域采集测试样本进行分类,得到目标位置.

在上述所有自适应算法中,分类器随着跟踪的进行而实时更新,但是当目标本身或其周围环境发生剧烈变化时,最新的分类器学习到了大量的非目标信息,这时若目标恢复原状或遮挡物从目标移开,最新的分

类器则不能对目标进行准确分类. 本研究通过构建分类器树保存历史分类器信息, 并根据树中路径距离选择出分类器集用于对测试样本分类, 有效地解决了这个问题. 为了保证跟踪速度, 本研究提出了一种基于簇相似性比较确定目标样本的方法. 该方法具有计算速度快, 分类能力强, 分类器易于保存, 更新速度快等优点.

2 算法描述

2.1 基于检测的跟踪流程

假设已知 t 时刻图像 F_t 中的目标为 x_t^* , 其位置为 $l(x_t^*)$. 首先从目标周围半径 r_p 内提取目标簇 $\chi_t^+ = \{x \mid \|l(x) - l(x_t^*)\| \leq r_p\}$. 其次在目标周围半径 r_{in} 和 r_{om} 之间采集背景簇 $\chi_t^- = \{x \mid r_{in} \leq \|l(x) - l(x_t^*)\| \leq r_{om}\}$. 目标簇和背景簇中的样本用于训练跟踪系统维持的分类器. 当第 $t+1$ 帧到来时, 使用滑动窗口在第 $t+1$ 帧目标周围采集测试样本集 $\chi_{t+1}^o = \{x \mid \|l(x) - l(x_t^*)\| \leq r_s\}$. 最后使用 t 时刻训练得到的分类器对测试样本集进行分类, 得到 $t+1$ 时刻的目标 x_{t+1}^* .

2.2 特征提取

本实验中使用随机生成的均值类哈尔特征模板提取样本的特征. 样本 x 的第 i 个特征可以表示为:

$$f_i(x) = \sum_j^n w_j \frac{\text{sum}(x_j')}{\text{pixn}(x_j')} \quad (1)$$

其中 x_j' 表示在样本 x 内部随机选择的矩形图像块, $\text{sum}(x_j')$ 表示 x_j' 内所有像素灰度值之和, $\text{pixn}(x_j')$ 表示图像块中像素的个数, w_j 表示 0 到 1 之间的随机权值, n_h 表示图像块的个数. 样本 x 的特征可以表示为 $f(x) = (f_1(x), \dots, f_{n_h}(x))$. 在跟踪进行前, 需要对类哈尔特征模板进行初始化, 模板在整个跟踪过程中保持不变.

2.3 分类器的构建

本文根据样本到样本簇之间的距离来定义它们之间的相似度. 定义第 i 维特征空间中样本 x 与簇之间的距离关系为:

$$d_i(x, \chi) \propto |f_i(x) - \mu_i| \quad (2)$$

其中 $\mu_i = \frac{1}{n} \sum_{j=1}^n f_i(x^{(j)})$, $x^{(j)} \in \chi$ 表示簇 χ 的中心位置, n 是簇中样本的个数. 为了在归一化的特征空间对相似度进行度量, 参考马氏距离定义的思想^[9], 定义一个与方差成反比的权重, 则在空间的第 i 维上, 样本到簇之间的距离定义为:

$$d_i(x, \chi) = \frac{|f_i(x) - \mu_i|}{\alpha(\sigma_i)^\beta} \quad (3)$$

其中 $\sigma_i = \sqrt{\frac{1}{n} \sum_{j=1}^n (f_i(x^{(j)}) - \mu_i)^2}$, $x^{(j)} \in \chi$ 为簇在第 i 维特征空间上的标准差, α, β 为控制方差影响的参数.

当 $\alpha = \beta = 1$ 时为样本到中心点的标准欧几里得距离. 加入了方差参数后, 原特征空间就变成了以簇方差为尺度的特征空间. 由于特征空间中的维度之间相互独立, 定义样本 x 到簇 χ 之间的距离为:

$$d(x, \chi) = \sum_i d_i(x, \chi) \quad (4)$$

样本与簇之间的相似度定义为与它们之间的距离成反比:

$$S(x, \chi) = e^{-d(x, \chi)} \quad (5)$$

样本与簇越相似, 则 $S(x, \chi)$ 越大. 定义目标函数 $J(x)$ 描述样本是目标的相对可能性, 即若 $J(x^{(1)}) > J(x^{(2)})$, 则样本 $x^{(1)}$ 更有可能是目标. 由于目标应该与目标簇相似, 而要与背景簇不相似, 因此目标函数 $J(x)$ 定义为与样本到目标簇之间的相似度成正比, 与样本到背景簇之间的相似度成反比:

$$J(x) = \frac{S(x, \chi^+)}{S(x, \chi^-)} \quad (6)$$

2.4 测试样本分类

设 $A_t = \{a_i\}_{i=1}^t$ 表示至 t 时刻为止所有历史分类器的集合, 其中元素 $a_i = (\mu_i^+, \sigma_i^+, \mu_i^-, \sigma_i^-)$ 表示 t 时刻目标簇和背景簇的统计特征参数的四元组, 即 t 时刻的分类器. 设 $C_t^k = \{c \mid c \in A_t, \text{dis}(c, a_t) \leq k\}$ 表示 t 时刻选出的用于分类 $t+1$ 时刻测试样本的分类器集合, 其元素 $c = (\mu_c^+, \sigma_c^+, \mu_c^-, \sigma_c^-)$, 分类器的选择和更新方法在第 2.5 节描述. 则目标在 $t+1$ 时刻的状态以及所使用的分类器可以通过式(7)得出:

$$(x_{t+1}^*, c^*) = \arg \max_{x \in \chi_{t+1}^o, c \in C_t^k} J(x, c) \quad (7)$$

其中 χ_c^+ 和 χ_c^- 表示分类器 c 对应时刻的目标簇和背景簇, 下标 i 表示特征空间的维度.

2.5 分类器的更新与选择

图 1 描述了多分类器的选择与更新策略. 在 t 时刻, 系统维持了一个树形数据结构表示分类器之间的派生关系. 在 $t+1$ 帧, 从分类器集合 A_t 选择一个子集 $C_t^k = \{c \mid c \in A_t, \text{dis}(c, a_t) \leq k\}$ 对 $t+1$ 帧的测试样本进行分类, 其中 $\text{dis}(c, a_t) \leq k$ 表示距离 a_t 路径长度小于或等于 k . 图 1 中虚线框内节点为距离 a_t 路径长度小于 2 的节点. 通过式(7)对 $t+1$ 帧测试样本分类后, 得出 x_{t+1}^* 和 c^* . 由于在跟踪过程中目标形变, 光照, 遮挡等因素的影响, 目标簇和背景簇的参数会发生相应的变化, 为了保证跟踪的鲁棒性和准确性, 需要对这些参数进行在线更新. 本文参考文献[8]中参数更新的方法, 在求出第 $t+1$ 帧目标的位置 x_{t+1}^* 后, 采集第 $t+1$ 帧的目标簇 χ_{t+1}^+ 和背景簇 χ_{t+1}^- , 并计算出它们的统计特征参数 $\mu_{t+1}^+, \sigma_{t+1}^+, \mu_{t+1}^-$ 和 σ_{t+1}^- . 则 $t+1$ 时刻的分类器 $a_{t+1} = (\mu_{t+1}^+, \sigma_{t+1}^+, \mu_{t+1}^-, \sigma_{t+1}^-)$ 可以表示为:

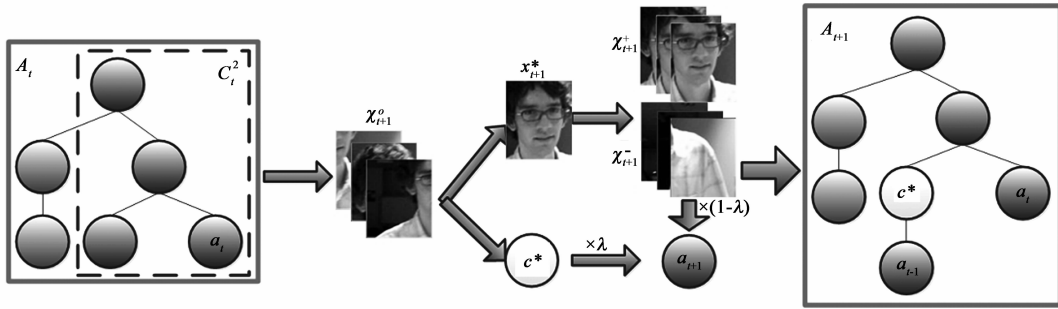


图1 多分类器的选择与更新策略

$$\begin{cases}
 \mu_{t+1}^+ = \lambda \mu_{c^*}^+ + (1 - \lambda) \mu_{t+1}^+ \\
 \sigma_{t+1}^+ = \lambda \sigma_{c^*}^+ + (1 - \lambda) \sigma_{t+1}^+ \\
 \mu_{t+1}^- = \lambda \mu_{c^*}^- + (1 - \lambda) \mu_{t+1}^- \\
 \sigma_{t+1}^- = \lambda \sigma_{c^*}^- + (1 - \lambda) \sigma_{t+1}^-
 \end{cases} \quad (8)$$

其中 λ 是学习率. 最后将 a_{t+1} 作为 c^* 的叶子节点加入到分类器树中.

2.6 跟踪算法步骤

本文跟踪算法的基本步骤如下:

初始化:第 1 帧的目标 x_1^* ; 第 1 帧的分类器 a_1 为簇 χ_1^+ 和 χ_1^- 的统计特征, 令 $A_1 = a_1$ 作为分类器树的根节点; 令 $t = 1$.

如果第 $t + 1$ 帧不是最后一帧, 则进入循环:

输入: t 时刻的目标位置 x_t^* ; 至 t 时刻的分类器集合 A_t 及对应的分类器树; 第 $t + 1$ 帧的图像.

步骤 1: 通过 $C_t^k = \{c | c \in A_t, dis(c, a_t) \leq k\}$ 选择出分类器集合.

步骤 2: 在第 $t + 1$ 帧采集测试样本集 χ_{t+1}^o .

步骤 3: 通过式(7)求得 $t + 1$ 时刻的目标位置 x_{t+1}^* 和对应的分类器 c^* .

步骤 4: 采集 $t + 1$ 帧的目标簇 χ_{t+1}^+ 和背景簇 χ_{t+1}^- , 通过式(8)更新 $t + 1$ 时刻的分类器 a_{t+1} .

步骤 5: 将 a_{t+1} 作为 c^* 的叶子节点加入到分类器树中.

输出: 第 $t + 1$ 帧的目标位置 x_{t+1}^* ; 更新后的分类器集合 A_{t+1} 以及新的分类器树.

3 实验结果与分析

实验与当前主流的 3 种算法在 3 个视频序列上进行对比. 参与对比的算法有: CT^[7], DFT^[10], TLD^[11], 本文算法简称 MCCS. 视频序列包括了实时跟踪中常遇到的光照变化, 形变, 遮挡, 复杂背景, 运动模糊等影响因素. 对比视频序列包含了人工标定的目标位置. 视频序列数据以及其它算法的实验数据可以从文献[12]得到.

3.1 参数设置

本实验中使用矩形框标记跟踪结果, 目标在第 1 帧中的位置通过人工标定. 试验参数设置如下: 目标簇采

集半径为 $r_p = 4$ 个像素, 背景簇采集内半径为 $r_{in} = 6$ 个像素, 外半径 $r_{on} = 15$ 个像素, 负样本采集数量为 50 个. 测试样本采集半径为 $r_s = 25$ 个像素. 类哈尔特征数 $n_f = 50$ 个. 类哈尔特征中图像块的个数 n_b 为 1 到 4 之间的随机整数. 多分类器选择中的节点距离参数 $k = 10$. 学习率 $\lambda = 0.9$. 实验环境为: Intel (R) Core (TM) i3-2120 CPU 3.30GHz, 8 GB RAM, 64 位 Windows 操作系统, 算法平均速度为 35 帧/秒.

3.2 评价方法

本实验中采用文献[12]中提出的重合度成功率作为评判方法. 定义为 $S = \frac{area(R \cap G)}{area(R \cup G)}$ 为重合度, 其中 R 表示算法跟踪的结果区域, G 表示人工标定的目标区域, \cap 和 \cup 表示两种区域的交集和并集, 函数 $area$ 表示区域中像素的个数. 若 S 大于某个阈值, 则认为该帧的跟踪是成功的.

3.3 实验结果分析

图 2 显示了在各种不同重合率阈值的情况下各种算法跟踪成功率. 其中横轴表示重合率阈值, 纵轴表示在该阈值下跟踪成功的帧数占视频总帧数的比例, 算法后面的数字为 AUC 值 (Area Under Curve, 即曲线与坐标轴围成区域的面积). 从各个测试视频的 AUC 值中可以看出, 本文的算法跟踪准确度更高. 部分实例跟踪结果如图 3 所示. 下面针对各个测试视频序列进行结果分析:

(1) Basketball: 该视频序列中存在的大量与目标相似的背景以及目标本身的跑动产生的形变都会干扰跟踪的结果. 在第 389, 485 帧目标发生了形变和遮挡, CT, TLD, DFT 算法跟踪到了非目标队员上. 第 650 帧时发生了强烈的光照变化, 只有 MCCS 跟踪准确. 说明本算法能够应对跟踪中遇到的遮挡, 形变以及光照变化等情况.

(2) Jumping: 在该视频序列中, 目标由于自身跳动以及相机的抖动产生了大量的模糊图片. 在第 59 帧中, DFT 和 CT 就丢失了目标. 从图 2 的统计数据中可以看出, 本算法处理运动模糊的效果优于其他算法.

(3) Shaking: 本视频序列主要用于测试算法应对灯

光变化的能力. 如图 3 所示, 在第 60 帧, 舞台灯光突然增强, 除本算法之外的所有算法都产生了不同程度的漂移. 在灯光强度减弱后, CT 和 TLD 由于学习到了大量的灯光信息而不能对目标进行准确定位. 在第 266 帧, 由于

光照强度变化, CT 和 TLD 发生了漂移, 本算法仍能够准确地跟踪目标位置. 从图 2 的数据中可以看出, 本算法在应对光照变化情况时能够保证跟踪精度.

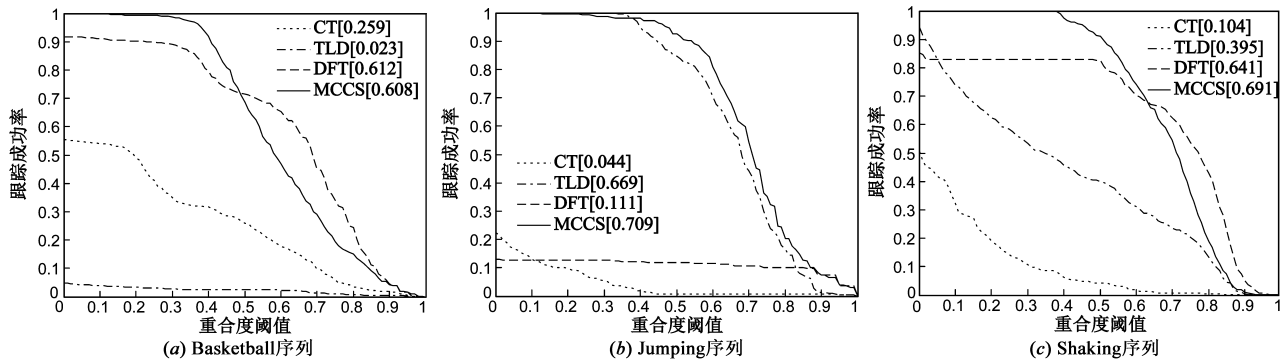


图2 跟踪算法成功率

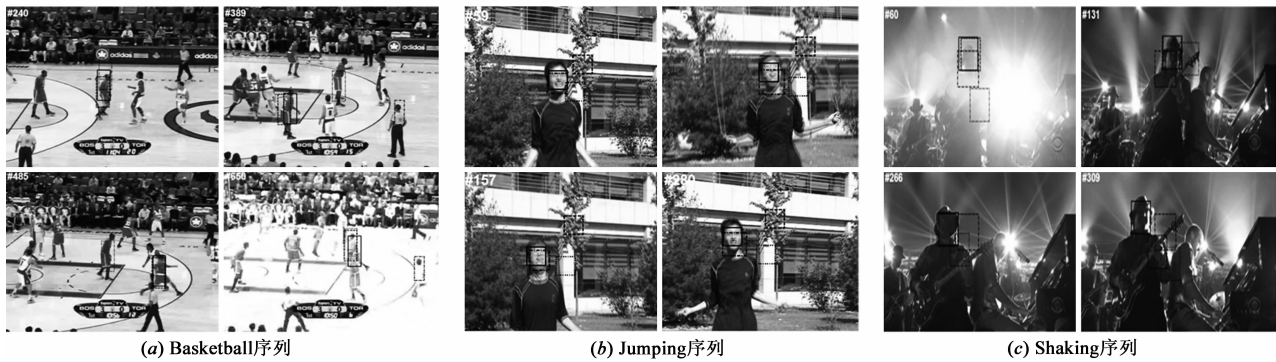


图3 部分跟踪结果

4 结束语

在自适应目标跟踪算法中, 由于目标被遮挡或背景的剧烈变化而使得分类器学习到了非目标信息, 从而减低了跟踪算法的性能. 本文针对该问题提出了一种多分类器的跟踪策略, 通过建立树形结构的分类器集合, 在每帧挑选出多个分类器对测试样本进行分类, 并挑选分类结果最好的分类器进行更新, 避免了因分类器学习过快而产生的跟踪漂移问题. 为了能够达到实时跟踪, 提出了一种基于簇相似性比较的快速分类算法, 通过在以方差为尺度的特征空间中比较样本与簇之间的相似性确定目标样本. 在与当前流行的跟踪算法的比较中体现了本方法的优越性.

参考文献

- [1] Yang H, Shao L, Zheng F, et al. Recent advances and trends in visual tracking: A review [J]. *Neurocomputing*, 2011, 74(18): 3823–3831.
- [2] Yilmaz A, Javed O, Shah M. Object tracking: A survey [J]. *ACM Computing Surveys*, 2006, 38(4): 13.
- [3] 权伟, 陈锦雄, 余南阳. 在线学习多重检测的可视对象跟踪方法[J]. *电子学报*, 2014, 42(5): 875–882.
QUAN Wei, CHEN Jin-xiong, YU Nan-yang. Online learning of multiple detectors for visual object tracking [J]. *Acta Electronica Sinica*, 2014, 42(5): 875–882. (in Chinese)
- [4] 姜明新, 王洪玉, 王洁, 王彪. 基于 ML 和 L2 范数的视频目标跟踪算法 [J]. *电子学报*, 2013, 41(11): 2307–2313.
JIANG Ming-xin, WANG Hong-yu, WANG Jie, WANG Biao. Visual object tracking algorithm based on ML and L2-norm [J]. *Acta Electronica Sinica*, 2013, 41(11): 2307–2313. (in Chinese)
- [5] Ross D A, Lim J, Lin R S, et al. Incremental learning for robust visual tracking [J]. *International Journal of Computer Vision*, 2008, 77(1–3): 125–141.
- [6] Mei X, Ling H. Robust visual tracking using l1 minimization [A]. *IEEE 12th International Conference on Computer Vision [C]*. Kyoto, Japan: IEEE, 2009. 1436–1443.
- [7] Zhang K, Zhang L, Yang M H. Real-time compressive tracking [A]. *Computer Vision-ECCV 2012 [C]*. Berlin Heidelberg: Springer, 2012. 864–877.

- [8] Babenko B, Yang M H, Belongie S. Visual tracking with online multiple instance learning [A]. IEEE Conference on Computer Vision and Pattern Recognition [C]. Florida, USA:IEEE,2009. 983 – 990.
- [9] Mahalanobis P C. On the generalized distance in statistics [J]. Proceedings of the National Institute of Sciences (Calcutta), 1936, 2: 49 – 55.
- [10] Sevilla-Lara L, Learned-Miller E. Distribution fields for tracking [A]. IEEE Conference on Computer Vision and Pattern Recognition [C]. Rhode Island; IEEE, 2012. 1910 – 1917.
- [11] Kalal Z, Matas J, Mikolajczyk K. Pn learning: Bootstrapping binary classifiers by structural constraints [A]. IEEE Conference on Computer Vision and Pattern Recognition [C]. San Francisco; IEEE, 2010. 49 – 56.
- [12] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark [A]. IEEE Conference on Computer Vision and Pattern Recognition [C]. Portland, Oregon; IEEE, 2013. 2411 – 2418.

作者简介



李康 男, 1986 年 9 月出生, 安徽亳州人. 武汉大学在读博士研究生, 主要研究方向: 图像处理、目标跟踪.
E-mail: likang@whu.edu.cn



何发智 (通讯作者) 男, 1968 年 11 月出生, 湖北武汉人. 武汉大学教授、博士生导师. 主要研究方向: 图像处理、计算机协同、计算机图形.
E-mail: fzhe@whu.edu.cn