

一种检测范围自适应的目标跟踪算法

黄元捷, 赵杰煜*, 程婷婷, 陈普强

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: 为处理目标的消失重现、形变及环境变化等问题, 要求跟踪算法有一定的检测与学习能力. 针对全局检测方法因冗余检测而造成检测效率低下的问题, 在基于 P-N 学习的跟踪框架的基础上, 提出一种自适应生成检测范围的目标跟踪算法. 通过引入卡尔曼滤波器(Kalman filter)对目标位置、尺度以及两者的变化速度进行预估, 在检测前根据预估信息自适应生成检测范围, 提高检测效率. 在公开的 CoGD 数据集上进行实验, 结果证明该算法较原始算法在准确度基本不变的基础上, 速度得到显著改善.

关键词: 目标跟踪; 目标运动预估; 卡尔曼滤波器; P-N 学习

中图分类号: TP391.4

文献标志码: A

文章编号: 1001-5132 (2015) 04-0036-06

视频目标跟踪是机器视觉研究中的核心内容之一, 已经被广泛应用于各种领域, 如智能视频监控、场景分析、智慧交通等. 然而, 视频中光照变化、运动模糊、目标非刚体、目标消失与重现等问题增加了跟踪的难度. 为处理目标的消失与重现、形变及环境变化等问题, 要求跟踪算法有一定的检测与学习能力. 目标消失与重现的解决方法是使跟踪器具备一定的检测能力, 当跟踪失败时可以检测目标以重启跟踪器. Tracking-by-detection 方法^[1]与跟踪检测集成方法^[2-3]中的跟踪器均具备检测能力, 然而这些方法均需要在跟踪之前对检测器进行离线训练, 并且检测器在跟踪过程中保持不变. 但是目标的形态往往是多变的, 当目标以训练数据中不存在的形态出现时, 检测器会出现错误而导致后续的跟踪错误.

为解决目标形变的问题, 自适应的跟踪法^[4-5]对目标外观进行建模. 但是这类方法仅仅对目标进行建模而忽略了背景的影响, 导致在复杂背景下容易跟踪失败. 为缓解背景对跟踪造成的负面影响, 最近提出的跟踪算法^[6-11]也对背景进行建模. 这些方法可归结为 2 类, 其中一类方法通过背景特征点与目标特征点的运动约束预测目标位置^[6-7].

另一类方法^[8-11]将背景视为一个独立的样本类别, 将跟踪任务转化为二元分类问题, 在跟踪过程中逐帧更新分类器, 使跟踪器能够处理明显的目标形变、短期遮挡以及复杂背景. 然而跟踪结果并不总是正确, 逐帧更新会导致分类器的错误加速积累, 所以这些方法会出现跟踪漂移, 并且若目标消失时间过长, 跟踪很有可能失败. 考虑到逐帧更新的缺点, 文献^[12-14]通过限制分类器更新的办法提高跟踪准确度, 如在初始帧训练一个辅助分类器^[14]或引入“共训练”方式^[12-13]. Kalal 等^[15]对上述问题进行了总结, 提出跟踪、学习与检测相结合的单目标长时间跟踪算法(Tracking Learning Detection, TLD), 较好地解决了目标在长时间跟踪过程中发生的消失重现与形态变化的问题, 同时具有较强的相似目标与复杂场景辨别能力.

TLD 算法在每一帧都进行全屏检测, 以达到当目标消失后重现时的鲁棒跟踪的目的, 但是却消耗了过多的计算资源. 在实际环境中, 目标大多数时间处于可视跟踪状态, 目标运动轨迹与尺度连续变化, 可以通过对目标运动预估, 合理缩小检测范围以提高算法性能. 笔者针对 TLD 算法在跟踪目标过程中全局检测而产生大量冗余检测的这

收稿日期: 2015-02-06.

宁波大学学报(理工版) 网址: <http://journal.g.nbu.edu.cn/>

基金项目: 国家自然科学基金(61175026); 科技部国际科技合作专项(2013DFG12810); 浙江省国际科技合作专项(2013C24027).

第一作者: 黄元捷(1988-), 男, 浙江宁波人, 在读硕士研究生, 主要研究方向: 计算机视觉. E-mail: yuangyuanjie@msn.com

*通信作者: 赵杰煜(1965-), 男, 浙江宁波人, 博士/教授, 主要研究方向: 计算机视觉与机器学习. E-mail: zhao_jieyu@nbu.edu.cn

个缺点, 提出一种检测范围自适应的目标跟踪算法. 当上一帧目标可视时, 通过卡尔曼滤波器对目标运动及尺度变化进行预估, 并对每一帧预估得到的信息进行处理, 根据目标的运动速度与预设的不确定度生成检测范围. 这种措施不仅缩小了检测范围, 明显提升了检测效率, 而且检测区域随目标运动自适应变化, 目标不容易滑出检测区域, 降低了误检率.

1 TLD 目标跟踪算法框架

TLD 目标跟踪算法框架包含 4 个主要部分: 跟踪器、检测器、整合器和学习模块. 跟踪器为一个能检测跟踪错误的中流跟踪器^[16], 在目标可视时, 跟踪器负责逐帧跟踪目标并在错误大于一定值时停止追踪. 检测器为一个级联分类器, 负责在每一帧中检测目标, 并在必要时重启跟踪器. 整合器根据目标模型评估跟踪器与检测器的性能, 并对两者的结果作融合处理. 学习模块根据整合器的结果生成训练样本, 更新检测器, 提高检测器下次检测时的性能. 当前实现的原始 TLD 算法, 学习模块只更新检测器而跟踪器保持固定^[15]. TLD 算法的框架结构如图 1 所示.

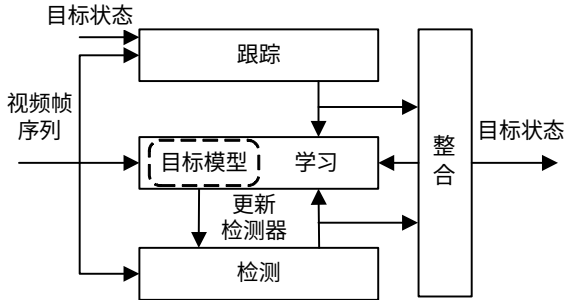


图 1 TLD 算法框架

在起始帧, 用户指定一个边界框表示目标的位置与大小, TLD 算法以此边界框初始化. 对于后续帧, 跟踪器与检测器共同进行处理. 跟踪器根据上一帧目标的位置与大小估计当前帧目标的状态. 检测器以滑动窗口方式对当前帧作完全扫描, 输出一个或者数个可能包含目标的窗口. 整合器对跟踪器与检测器的结果进行处理, 输出当前帧的最终跟踪结果. 最后学习模块根据整合器结果更新检测器.

在 TLD 算法初始化时, 检测器会以用户给予

的边界框为模板, 以其不同尺度建立扫描窗口. 检测器在检测过程中将各扫描窗口对应的 patch 依次通过方差过滤器、组合分类器、最近邻分类器, 完全通过级联分类器的 patch 即为检测结果, 如图 2 所示.

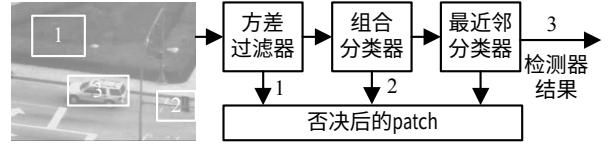


图 2 检测器框图

检测器在每一帧都会对每个扫描窗口进行检测, 而大部分窗口并不包含目标, 造成了大量的冗余检测. 检测器与跟踪器不同, 跟踪器只对固定数量的点进行跟踪, 几乎不受视频分辨率大小的影响. 随着视频分辨率的增加, 扫描窗口的数量会成倍增加, 如当视频分辨率为 320×240 时, 约有 5 万个扫描窗口, 而当分辨率为 640×480 时, 扫描窗口的数量可能超过 20 万. 当视频分辨率较大时, 检测器的处理速度会是整个算法的瓶颈. 在实际运行过程中, TLD 算法的速度很大程度上取决于检测器的处理速度.

2 基于卡尔曼滤波的检测框过滤器

在实际视频序列中, 相邻两帧间的时间间隔很短, 可以认为目标在相邻两帧间做匀速运动, 系统为线性模型. 通过使用卡尔曼滤波器对目标运动进行预估, 从而得到目标在当前帧的位置与大小. 然而当目标运动不规律时卡尔曼滤波器得到的预估信息是不准确的. 本研究利用目标的预估速度预测目标可能出现的区域, 将之作为检测范围. 通过滤去不在生成检测范围内的检测框, 提高检测效率.

2.1 卡尔曼滤波提取目标运动信息

采用卡尔曼滤波对目标运动建模, 扩展文献[17]的方法以获取生成检测范围所需运动信息. 卡尔曼滤波是一种高效率的自回归滤波器, 它能够从一系列的不完全及包含噪声的测量中, 估计动态系统的状态^[18].

卡尔曼滤波的数学描述为:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k, \quad (1)$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \quad (2)$$

其中: x_k 与 z_k 分别为 k 时刻的系统状态向量与观测向量; A 与 H 分别为系统状态转移矩阵与观测矩阵; w_k 与 v_k 为 2 个独立的均值为 0, 协方差分别为 Q 与 R 的高斯白噪声因素.

以视频帧中目标的中心位置和尺度大小构建运动模型和观测模型, 即系统状态向量为:

$$x_k = [p_x \ p_y \ s \ v_x \ v_y \ v_s]^T, \quad (3)$$

观测向量为:

$$z_k = [p_x \ p_y \ s]^T, \quad (4)$$

其中: p_x 与 v_x 分别为目标中心位置在水平方向上的分量与速度; p_y 与 v_y 分别为目标中心位置在垂直方向上的分量和速度; s 与 v_s 分别为目标尺度大小以及其变化速度. 目标的尺度大小 s 定义为:

$$s = \sqrt{S/S_{\text{init}}}, \quad (5)$$

其中: S 为目标边界框面积大小, S_{init} 为初始边界框面积大小, 目标初始化时尺度大小设置为 1.

系统状态转移矩阵为:

$$A = \begin{bmatrix} 1 & 0 & 0 & t & 0 & 0 \\ 0 & 1 & 0 & 0 & t & 0 \\ 0 & 0 & 1 & 0 & 0 & t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

观测矩阵为:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad (7)$$

其中: t 为两帧间隔时间.

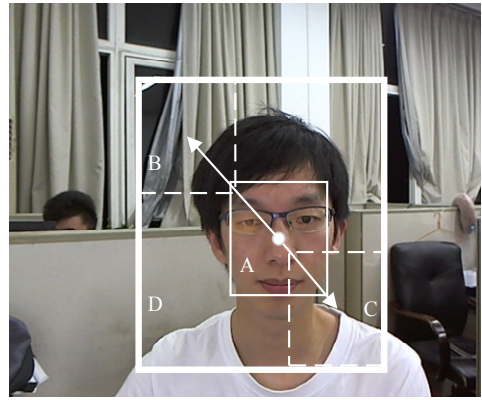
文中 Q 与 R 固定, 分别为 $10^{-5}I_6$ 与 $10^{-1}I_3$, 其中 I_6 与 I_3 分别为 6×6 和 3×3 的单位矩阵.

2.2 利用预估信息自适应生成检测范围

在已知初始边界框信息的基础上, 利用卡尔曼滤波可以得到当前帧目标边界框的预估位置及大小, 但该预估位置与大小不准确, 尤其当目标运动变化较大时, 预估位置与目标位置偏差很大. 故提出了一种基于卡尔曼滤波预估信息自适应生成检测范围的方法, 通过引入 2 个预设度量值 m 与 n 描述目标在预估运动方向以及其相反方向上的运动不确定性, 具体检测区域生成方式为: (1)在保持预估尺度不变的情况下, 预估边界框以预估移动

速度 (v_x, v_y) 的 m 倍移动, 计算其在下一帧到达的位置. (2)在保持预估尺度不变的情况下, 预估边界框以预估移动速度 (v_x, v_y) 的 $-n$ 倍移动, 计算其在下一帧到达的位置. (3)取包含上述 2 个边界框的最小矩形为检测区域.

图 3 给出了构建检测区域的示意图, 框 A 为卡尔曼滤波预估得到的边界框, 框 B 和框 C 分别为(1)和(2)步骤所得, 框 D 为最终得到的检测区域. m 与 n 可根据场景不同进行设置, 一般情况下取 m 大于 n , 本文中 m 和 n 分别取 4 和 2.



框 A 为卡尔曼滤波器预估框, 框 B 和 C 为预估移动位置, 框 D 为最终预估检测区域

图 3 构建检测区域

预估尺度大小范围时, 使用类似的计算方式预估边界框大小范围. (1)预估边界框以预估尺度变化速度的 m_s 倍变化, 计算其在下一帧的尺度大小值 s_m . (2)预估边界框以预估尺度变化速度的 n_s 倍变化, 计算其在下一帧的尺度大小值 s_n . (3)取尺度大小在 s_m 与 s_n 之间的检测框作为有效检测框. 为提高有效检测框搜索效率, 可在算法初始化建立扫描窗口时, 按照不同尺度对扫描窗口进行划分, 并按照尺度大小进行排序. 本文 m_s 与 n_s 的值分别取 2 和 1.

2.3 改进后的检测器

在检测前, 首先将检测框与 2.2 节中生成的检测区域比较, 若检测框与生成检测区域不重叠或者尺度不在预估大小范围内, 该检测框被否决从而不进入后续的级联分类器. 这样减少了进入级联分类器的检测框数量, 加快了检测的速度.

在初始化时, 使用初始边界框的中心位置与尺度, 并且以各速度为 0 初始化卡尔曼滤波器, 并

在目标可视时, 以当前帧的最终跟踪结果更新卡尔曼滤波器状态. 由于目标消失后重现的位置不可确定, 算法在目标消失后依旧采用全局检测, 并且当检测器重新检测到目标时, 利用检测到的目标位置及尺度, 并且以各速度为 0 重新初始化卡尔曼滤波器.

增加检测框过滤器后, 检测器的结构如图 4 所示, 图 4 中虚线框为使用本文方法自适应生成的检测区域(预估尺度范围未标出), 框 1、2、3 为检测框. 在增加检测框过滤器后, 检测框 1、2 由于与生成的检测区域不重叠, 会直接被检测框过滤器否决, 不会进入后续的级联分类器进行分类, 节省了计算资源.

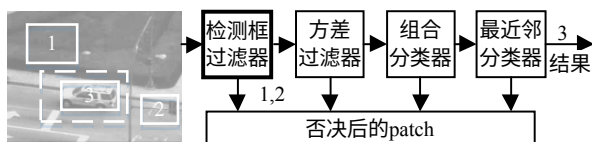


图4 改进后的检测器框图

3 实验结果及分析

以 Visual Studio 2013 与 OpenCV 2.47 为开发平台实现了本文算法, 算法流程为:

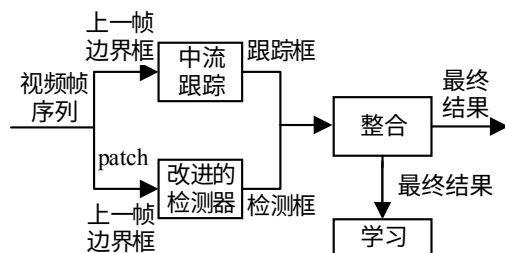


图5 算法流程图

在起始帧, 用户指定一个边界框表示目标的位置与大小, 算法以此边界框初始化. 从下一帧开始执行以下算法: 若上一帧目标可视, 使用中流跟踪算法预测目标在当前帧的位置; 改进后的检测器进行目标检测: (1)卡尔曼滤波器提取目标运动信息; (2)利用预估信息生成检测范围; (3)对在生成检测范围内的 patch 级联分类, 输出检测结果.

整合(1)、(2)步结果, 输出当前帧的最终结果, 并根据结果的置信度, 对检测器进行更新. 当前帧跟踪结束. 重复(1)、(2)、(3)步, 直到跟踪结束.

接着我们在 CoGD 数据集^[13]上对本文方法的

性能进行评估, 并对自适应检测范围生成的有效性进行验证. 实验在 CPU 为 Intel Core i7-2600K 3.40 GHz, 内存为 16 G 的计算机上进行.

3.1 测试数据集说明

采用公开测试集 CoGD^[13]作为测试数据集, 可以分别测试算法在遮挡、形变、光照变化及相似目标等方面的性能. 测试视频的有关信息见表 1, 相关视频截图如图 6 所示, 其中框表示所要跟踪的目标. 评价标准使用 f-measure, 计算公式为:

$$F = 2PR / (P + R), \quad (8)$$

其中: P 表示精确度, R 表示召回率.

表1 测试视频描述

视频名	帧数	摄像头移动	部分遮挡	完全遮挡	形态变化	光照变化	尺度变化	相似物体
David	761	√	√	×	√	√	√	×
Jumping	313	√	×	×	×	×	×	×
Pedestrian1	140	√	×	×	×	×	×	×
Pedestrian2	338	√	√	√	×	×	×	√
Pedestrian3	184	√	√	√	×	×	×	√
Car	945	√	√	√	×	×	×	√



图6 测试视频截图

3.2 跟踪准确度

实验将本文算法与原始 TLD 作比较, 说明本文算法的高准确性. TLD 方法的结果直接取自文献 [15]. 实验结果见表 2, 结果格式为 Precision/Recall/f-measure, 粗体表示最好结果. 从表 2 可看出本文

表2 视频序列测试结果

视频名	帧数	TLD ^[15]	本文算法
David	761	1.00/1.00/1.00	1.00/1.00/1.00
Jumping	313	1.00/1.00/1.00	1.00/1.00/1.00
Pedestrian1	140	1.00/1.00/1.00	1.00/1.00/1.00
Pedestrian2	338	0.89/0.92/0.91	1.00/0.94/0.97
Pedestrian3	184	0.99/1.00/0.99	0.99/1.00/0.99
Car	945	0.92/0.97/0.94	0.93/0.96/0.94

算法与原始 TLD 的算法准确度基本持平, 平均准确度高出原始 TLD 算法约 1.0%。关于 TLD 方法与其他非 TLD 方法的测试比较, 可参见文献[15]。

3.3 处理速度

原始 TLD 算法为 Matlab+C 实现, 而本文算法是由 C++实现, 由于编程语言的效率差异, 两者没有可比性。为了得到合理的实验结果, 使用 C++重新实现了原始 TLD 算法, 用于测试。

采用视频 1、4、6 来测试算法的运算速度, 并且简单地在 640×480 的摄像头下测试较高分辨率下算法的处理速度。每个算法均使用同一位置初始化, 结果用帧率表示, 粗体表示最好结果, 实验结果见表 3。

表 3 处理速度测试结果(帧率)

视频名	帧数	TLD(C++)	本文算法
David	761	30.0	50.2
Pedestrian2	338	48.5	56.3
Car	945	25.6	58.8
摄像头(640×480)	~2 000	~16	~31

在 1、4、6 号视频中, 本文算法胜出 TLD 约 57.1%, 而在较高分辨率下, 胜出 TLD 将近 1 倍。从实验结果可以看出, 增加卡尔曼检测框过滤器后, 本文算法的速度与原始 TLD 相比有大幅提高, 并且随着视频分辨率的提高, 本文算法的速度优势更加明显。

3.4 自适应生成检测区域

本文提出了一种方法使预估的检测区域能随着目标运动的预估速度自适应变化。在这部分实验中, 将检测区域预测框可视化, 以证明该策略的有效性。该实验以视频 6 为例, 结果可见图 7。图 7 中白框即本文根据卡尔曼滤波器预估信息生成的检测框, 黑框为实际跟踪结果。在第 56 帧, 目标消失后重新出现, 重新初始化卡尔曼滤波器; 在第 74 帧目标开始水平移动, 检测区域变大; 在第 385 帧, 目标运动趋于稳定, 检测区域开始缩小; 在第 447 帧, 目标运动继续稳定, 检测区域接近实际跟踪结果; 在第 515 帧, 目标消失, 卡尔曼滤波器停止工作; 在第 569 帧, 目标重新出现, 重新初始化卡尔曼滤波器, 生成检测区域。

由实验结果可以看出, 卡尔曼检测框过滤器正常工作, 预估检测位置并限制检测区域, 随着目

标运动, 检测区域自适应变化。

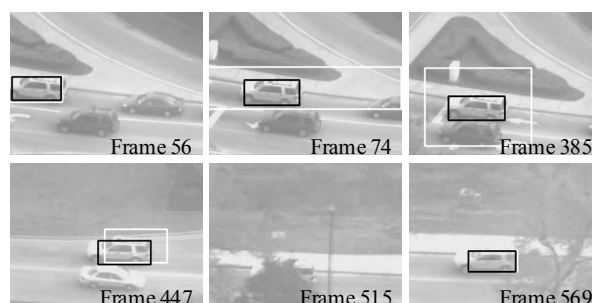


图 7 卡尔曼滤波预估检测区域示例

4 结论

针对原始 TLD 算法跟踪目标过程中全局检测产生大量冗余检测的缺点, 提出一种自适应生成检测范围的目标跟踪算法。通过引入卡尔曼滤波器对目标运动进行预估, 自适应调整检测范围, 滤去不在检测范围内的检测框, 提高检测效率。在公开的 CoGD 数据集上进行实验, 证明本文算法能较好地解决原始 TLD 检测效率低的缺点, 显著提高了算法速度, 使其更具有工程应用价值。在实际使用中, 发现本文算法在长视频中会出现准确度下降的问题, 下一步可考虑提升算法在长视频中的稳定性, 也可以对算法进行扩展, 使其能够应用于多目标跟踪。

参考文献:

- [1] Lepetit V, Laguerre P, Fua P. Randomized trees for real-time keypoint recognition[C]//Proceedings of the Computer Vision and Pattern Recognition (CVPR), San Diego CA, 2005:775-781.
- [2] Andriluka M, Roth S, Schiele B. People-tracking-by-detection and people-detection-by-tracking[C]//Proceedings of the Computer Vision and Pattern Recognition (CVPR), Anchorage AK, 2008:1-8.
- [3] Ramanan D, Forsyth D A, Zisserman A. Strike a pose: Tracking people by finding stylized poses[C]//Proceedings of the Computer Vision and Pattern Recognition (CVPR), San Diego CA, 2005:271-278.
- [4] 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.
- [5] Kwon J, Lee K M. Visual tracking decomposition[C]//Proceedings of the Computer Vision and Pattern Recognition (CVPR), San Francisco CA, 2010:1269-

- 1276.
- [6] Yang M, Wu Y, Hua G. Context-aware visual tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(7):1195-1209.
- [7] Grabner H, Matas J, Van G L, et al. Tracking the invisible: Learning where the object might be[C]// Proceedings of the Computer Vision and Pattern Recognition (CVPR), San Francisco CA, 2010:1285-1292.
- [8] Collins R T, Liu Y, Leordeanu M. Online selection of discriminative tracking features[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(10): 1631-1643.
- [9] Avidan S. Ensemble tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(2): 261-271.
- [10] Grabner H, Bischof H. On-line boosting and vision[C]// Proceedings of the Computer Vision and Pattern Recognition (CVPR), New York NY, 2006:260-267.
- [11] Babenko B, Yang M H, Belongie S. Visual tracking with online multiple instance learning[C]//Proceedings of the Computer Vision and Pattern Recognition (CVPR), Miami FL, 2009:983-990.
- [12] Tang F, Brennan S, Zhao Q, et al. Co-tracking using semi-supervised support vector machines[C]//Proceedings of the the 11th International Conference on Computer Vision, Rio de Janeiro, 2007:1-8.
- [13] Yu Q, Dinh T B, Medioni G. Online tracking and reacquisition using co-trained generative and discriminative trackers[M]. Berlin Heidelberg: Springer, 2008:678-691.
- [14] Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking[M]. Berlin Heidelberg: Springer, 2008:234-247.
- [15] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7):1409-1422.
- [16] Kalal Z, Mikolajczyk K, Matas J. Forward-backward error: Automatic detection of tracking failures[C]// Proceedings of the 20th International Conference on Pattern Recognition (ICPR), Istanbul, 2010:2756-2759.
- [17] 周鑫, 钱秋朦, 叶永强, 等. 改进后的TLD视频目标跟踪方法[J]. 中国图象图形学报, 2013, 9:1115-1123.
- [18] Kalman R E. A new approach to linear filtering and prediction problems[J]. Journal of Basic Engineering, 1960, 82(1):35-45.

An Adaptive Approach for Generating Detective Range in Visual Object Tracking

HUANG Yuan-jie, ZHAO Jie-yu*, CHENG Ting-ting, CHEN Pu-qiang

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: Visual object tracking is a highly challenging task due to the fact that it suffers from many intrinsic problems, such as object disappearance, reappearance, deformation and environment variation. A tracking algorithm is required to have the ability of detecting and learning. Given the fact that the global detector would generate certain redundant detections, we propose an approach that adaptively generates the detective range for visual object tracking based on the framework of P-N learning. We introduce a Kalman filter to predict the location, the object scale as well as their changing rate. Then we utilize the predicted information to reduce the scope of detecting. Our experimental results on the pubic CoGD dataset show that our method increases the speed dramatically while without compromising the accuracy.

Key words: visual object tracking; target moving prediction; Kalman filter; P-N learning

(责任编辑 史小丽)