文章编号:1671-4598(2018)09-0130-04 DOI:10.16526/j.cnki.11—4762/tp.2018.09.028 中图分类号:TP3 文献标识码:A

基于 CUDA 加速的 TLD 算法实现

张旭标,卢 迪,林尤添,梁金伟
(哈尔滨理工大学 电气与电子工程学院,哈尔滨 150000)

摘要:TLD(Tracking—Learning—Detection)是一种稳定可靠的单目标长时间在线跟踪算法;该算法能够有效的解决目标物体在被跟踪过程中发生的形变、遮挡等问题;将 TLD 算法移植到嵌入式开发板中并对其实时性进行了改进:对目标模型进行了优化存储,使系统初始化速度更快;对方差滤波器使用 CUDA 进行并行计算,缩短计算时间;对集合分类器使用距离聚类算法,减少了不必要资源浪费;此外,搭建了基于 P2P 的服务器客户端模型,能够实时的对跟踪目标进行远程监控。

关键词:TLD;CUDA;目标跟踪

TLD Algorithm Implementation Based on CUDA Accelerated

Zhang Xubiao, Lu Di, Lin Youtian, Liang Jinwei
(Harbin University of Science And Technology, Harbin 150000, China)

Abstract: TLD (Tracking—Learning—Detection) is a stable and reliable single—target tracking algorithm for long—term online tracking. The algorithm can effectively solve the problems such as deformation and occlusion of the target object in the process of being tracked. In this paper, the TLD algorithm is transplanted into the embedded development board and its real—time performance is improved: the target model is optimized in order to make the system initialize faster. In order to shorten the computation time , using CUDA to perform parallel computation in the variance classifier. The collection classifier uses distance clustering algorithms to reduce unnecessary waste of resources. In addition, building a server based on a P2P model, which can remote monitor the target in real time.

Keywords: TLD; CUDA; target tracking

0 引言

目标跟踪在导航制导、医学诊断、智能监控等领域有着广泛的应用,目标跟踪算法[1-4]通常面临着光照变化、外观变形、背景相似干扰、快速移动和运动模糊等[5-7]几大难点。Kass 等人提出的主动轮廓模型(Snake 模型)[8]不仅考虑图像的灰度信息,而且考虑了整体轮廓的几何信息,跟踪可靠性更强,但是 Snake 模型计算量较大且对于形变较大或快速移动的物体跟踪效果不够理想;Yizong Cheng 改善的均值漂移算法(Meanshift 算法)[9]利用了梯度优化方法实现快速目标定位,对目标的变形、旋转等运动有较好的适用性,但在跟踪过程中没有利用目标在空间中的运动方向和运动速度信息,当周围环境存在干扰时(如光线、遮挡),容易丢失目标[10]。Zdenek Kalal 提出的 TLD 算法[11-16],通过将传统的跟踪算法与检测算法相结合,并通过在线学习机制不断更新跟踪模块的显著特征点和检测模块的目标模型与相关参数,从而有效的解决目标物体发生形变、光照变化、尺度变化以及遮挡等问题,使得跟踪效果鲁棒性更好。

本文对传统的 TLD 算法进行了嵌入式设备的移植,加入了离线目标模型的初始化操作,并使用 CUDA(Compute Unified Device Architecture)并行化编程方式对方差滤波器进行并行计算,缩减了原来的 CPU 单线程的计算时间;对通过集合分类器的样本使用聚类算法,将远离目标的图像框排除,提高跟踪鲁棒性。在保证跟踪效果的前提下,减少了资源的占有率,缩短了运行时间。

1 TLD 跟踪算法原理

TLD 算法主要由追踪、检测和学习 3 个部分组成,TLD 算法的运行流程图如图 1 所示。用第一帧图像中手动圈定需要跟踪的目标物体对跟踪模块、检测模块进行初始化操作,从第二帧图像开始,跟踪—检测—学习 3 个模块互相补充,不断的对错误进行评估,从而实现目标在发生形变或被遮挡的情况下仍然能准确追踪的效果。

跟踪模块采用中值流跟踪法(Median Flow),该算法本质上是基于前后误差估计(forward—backward error)的金字塔 LK 光流法。其基本原理为:在上一帧的目标框中选择若干个像素点作为特征点,利用金字塔 LK 光流法在下一帧中寻找上一帧的特征点在当前帧中的对应位置,然后将这若干个特征点在相邻两帧之间的位移变换进行排序,得到位移变化的中值,将小于中值的 50%的特征点作为下一帧的特征点,并依次进行下去,从而实现动态更新特征点

收稿日期:2017-11-02; **修回日期**:2018-03-12。

作者简介:张旭标(1995—),男,广东省人,本科生,主要从事物体跟踪方向的研究。

卢 迪(1971-),女,天津市人,教授,主要从事多源数据融合,图像处理方向的研究。

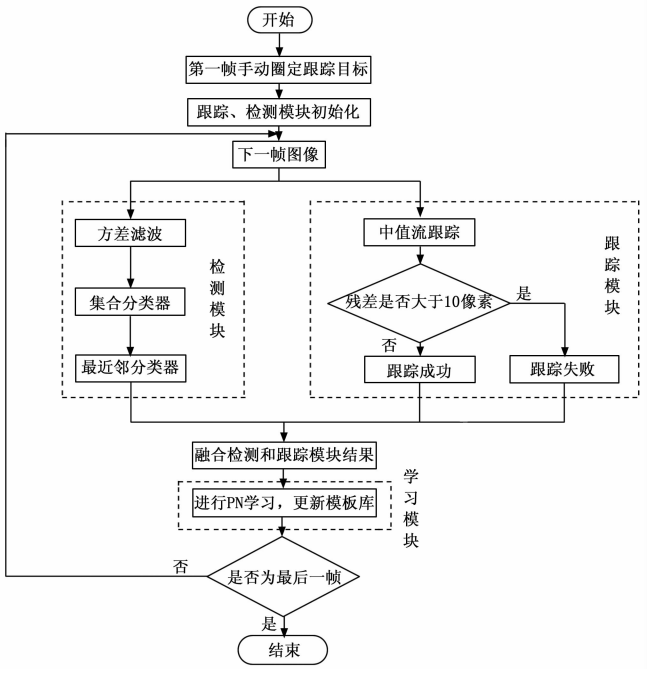

图 1　TLD 算法流程图

的目的。跟踪器是用于跟踪连续帧间的运动，只有在物体始终可见时跟踪器才会有效，跟踪器根据物体在前一帧已知的位置估计在当前帧的位置，同时在估计过程中产生一条目标运动的轨迹作为学习模块的正样本。

检测模块是将方差滤波器、集合分类器以及最近邻分类器级联。方差滤波器是计算图像片像素灰度值的方差，将方差小于原始图像片方差一半的样本标记为负；集合分类器是利用随机森林的形式，对图像片中任意选取的两点进行亮度值的比较，若 A 点的亮度大于 B 点的亮度，则特征值记为 1，否则记为 0，每选择一对新位置，就是一个新的特征值，随机森林的每个节点就是对一对像素点的比较。最近邻分类器是通过设定阈值对新样本进行相对相似度的计算，大于设定的阈值则认为是正样本。通过对每一帧的图像进行全面扫描，找到与目标物体相似的所有外观的位置，从检测产生的结果中生成正样本和负样本，传到学习模块。

学习模块采用一种半监督的机器学习算法 P－N 学习（P－N Learning），根据追踪模块和检测模块产生的正负样本，使用 P 专家（P－expert）检出漏检（正样本误分为负样本）的正样本，使用 N 专家（N－expert）改正误检（负样本误分为正样本）的正样本。由于每一帧图像内物体最多只可能出现在一个位置，且相邻帧间物体的运动是连续的，所以连续帧的位置可以构成一条较平滑的轨迹。P 专家的作用是寻找数据在时间上的结构性，它利用追踪器的结果预测物体在 $t+1$ 帧的位置。如果这个位置（包围框）被检测器分类为负，P 专家就把这个位置改为正。也就是说 P 专家要保证物体在连续帧上出现的位置可以构成连续的轨迹；N 专家的作用是寻找数据在空间上的结构性，它把检测器产生的和 P 专家产生的所有正样本进行比较，选择出一个最可信的位置，保证物体最多只出现在一个位置上，把这个位置作为 TLD 算法的追踪结果。同时这个位置也用来重新初始化追踪器。

2　基于 CUDA 加速的 TLD 改进算法

将图 1 所示的 TLD 算法移植至嵌入式设备时，由于其串行运算特点，运算时间较长，不能满足跟踪系统的实时性要求。而目前随着 GPU 发展的越来越强大，其计算能力已经超越了通用的 CPU，将 CPU 与 GPU 并用，进行协同处理，是计算行业发展大趋势，基于此 NVIDIA 发明 CUDA 运算平台。本文利用 CUDA 运算平台，采用多线程方式计算大量数据，缩短运算时间，以满足嵌入式设备运动目标跟踪系统的实时性要求。此外，本文在初始化操作中加入了离线目标模型模块，增强了 TLD 算法目标跟踪的自适应性能；采用基于 K 均值的聚类算法对经过集合分类器后的样本数据进行了聚类，降低最近邻分类器的阈值，从而提高检测率，最后将跟踪目标传输到 C/S 服务器的上位机端进行显示。基于 CUDA 加速的 TLD 算法框图如图 2 所示，其中虚线部分即为改进部分。

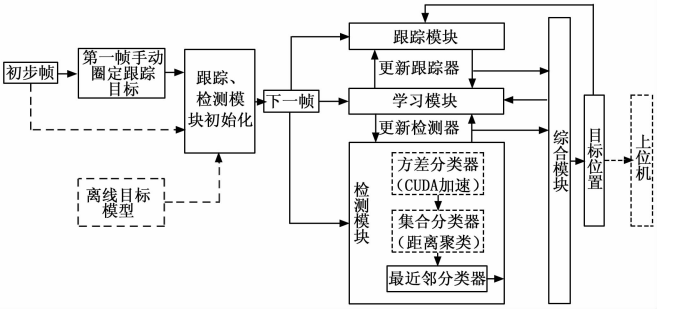

图 2　基于 CUDA 平台的 TLD 算法框图

2.1　离线目标模型的初始化

图 1 所示 TLD 算法在每次启动时都需要手动选取跟踪目标，本文在此基础上加入离线目标模型的初始化操作过程。对于已圈定的跟踪目标，在第一次启动系统时进行目标的选取操作后，将目标模型进行保存，下次启动系统时，只需读入目标模型即可实现对跟踪模块和检测模块的初始化，增强了 TLD 算法的自适应性能。

2.2　CUDA 平台下方差滤波器算法

在 TLD 算法中，检测模块功能是在视频当前帧中定位与目标模型匹配的目标图像，因此要构建扫描窗口扫描当前帧图像，并判断当前扫描窗口内是否存在目标图像。对图像使用不同尺寸的扫描窗（scanning grid）进行逐行扫描时，就会在每一个位置产生一个包围框（bounding box），包围框中确定出的图像区域即为图像片（patch），所有图像片的集合进入到学习部分就成为了一个样本。经过扫描窗口产生的样本是无标签的样本，需要使用分类器对无标签

的样本进行分类，从而确定其标签。

假设视频的分辨率为 $m\times n$，步长为 k 个像素，尺度缩放系数为 l，则扫描过程中会产生 $m\times n\times l/k$ 个图像片。图 1 所示 TLD 算法中采用的步长为 10 像素、尺度缩放系数有 21 种的扫描窗口策略，需要计算每一个扫描窗口与输入目标的重叠度，由此带来了大量的样本数据进行同样的计算。由于采用 CPU 单线程依次计算，势必导致了大量的时间损耗。

采用 NVIDIA 推出的 CUDA 运算平台，运用多线程方式计算大量的样本数据，具体步骤如下：

1）设置视频分辨率为 240×180，选用尺度缩放系数有{ 0.16151f， 0.19381f， 0.23257f， 0.27908f， 0.33490f， 0.40188f， 0.48225f， 0.57870f， 0.69444f， 0.83333f， 1.0f， 1.20000f， 1.44000f， 1.72800f， 2.07360f， 2.48832f， 2.98598f， 3.58318f， 4.29982f， 5.15978f， 6.19174f} 共 21 种，步长为 10 像素。则产生 240×180×21÷10＝90720 个图像，记为 N；

2）在 CPU 端计算帧图像积分以及帧图像平方的积分，分别记为 I 和 II；

3）GPU 每一个线程块中设置 256 个线程，则线程网络中产生（N＋255）/256 个线程块；

4）将图像片数据量 N、阈值 T、帧图像积分 I、帧图像平方的积分 II 传入 GPU 显存中；

5）调用核函数，每一个线程块下的每一个线程计算一个图像片的方差，并与阈值进行比较。大于阈值的是正样本，相应的数组下标的值记为 1，小于阈值的是负样本，相应的数组下标的值记为 0；

6）将显存中的数组数据从显存传回到内存中，将正负样本过滤分离，传入集合分类器。

2.3 集合分类器的聚类

方差滤波器过滤后的图像片传入集合分类器中，集合分类器中采用随机森林计算后验概率，根据阈值判定出需要进入最近邻分类器的正负样本。随机森林包含了 10 棵树，每棵树为完全二叉树，每棵树都包含 13 个节点，每个节点作为一个特征，每个节点是一个像素对的比较，每个像素对的采点位置是通过第一帧初始化随机生成的，并且互相垂直，将 13 个点的值统计做成一个 13 位的二进制数字从而进行后验概率的统计计算。若该图像区域的后验概率平均值大于设定的阈值，则判定为正样本，否则判定为负样本。

在实际情况中，集合分类器所表示的是目标的大概位置，能很好的表示物体的位置，最近邻分类器利用保守相似度和相关相似度去判断是否能通过级联分类器，从而保证成功获得目标。为了提高精度，提高识别度，在集合分类器后进行 K 均值聚类，对通过集合分类器的相关性大的图像片进行聚合，将远离目标的图像片去除，降低最近邻分类器阈值，在综合出新的目标图像片后传入到最近邻分类器中进行评估。

具体步骤如下：

1）提取出所有通过集合分类器的每个样本的中心坐标点；

2）将当前样本的中心坐标点与其他的样本中心坐标点做差得到一系列的距离值；

3）对每个样本求得的所有距离值取平均，作为该样本的参考值；

4）将参考值由小到大进行排序，取出前 1/2 的样本进行聚类，若样本数量超过 50 个，则只取前 50 个样本；

5）降低最近邻分类器的阈值，对新目标进行分类。

2.4 客户端/服务器的设计

客户端/服务器（C/S）部分采用 socket 网络编程技术，通过 TCP/IP 协议编写基于 P2P 模型的客户端和服务器精灵进程，实时的向指定路径写入和读取数据。

在实际情况中，保证系统服务器平台的鲁棒性，系统采用了如下几种技术细节保证高效性及扩展性：

1）采用 Reactor 模式事件处理机制，利用主进程作为监听网络事件，如果有事件发生，则通过主进程利用管道技术通知并分发到各个处理进程。相比于 Select 的 I/O 复用技术，为了突破网络端口的监听上数量上的限制，采用 Epoll 模型的边缘触发的 ET 模式提高系统的鲁棒性。

2）在满足多目标跟踪的设计下，利用单例模式创建进程池，提高扩张性，单例设计模式保证进程池的唯一性，进程池的维护减少了进程创建和销毁所带来的 CPU 负荷和内存的是使用情况。

3）系统在设计时，利用多进程技术和加权 Round Robin 算法对子线程的分发进行处理提高服务器的处理能力，完成了 Reactor 的主线程与 Worker 线程均衡分发，增加了理解性和简化了调试。Round Robin 算法可以有效的解决单进程的任务负载过重时，保证均衡负载。

4）使用管道统一事件源机制保证父子进程信号及信号处理函数尽可能快的被执行完毕，有效的防止服务器的父进程程序的提前退出的情况活宕机而导致产生僵尸进程。利用统一事件源管理机制可以有效的解决该问题。

5）本系统设计了采用高性能定时器，基于时间堆的心跳定时器，在对方心跳超时或者宕机的情况下，则自动杀死该进程，重新启动，保证了系统正常的运行。对于传统的降序链表定时器，升序链表定时器和基于 hash 的时间轮来说，最小堆定时器在空间上位 O（1），相较于 O（N）的 hash 时间轮的空间复杂度优越的多，当系统连接数很大的时候，最小堆的添加时间复杂度位 O（lgn），相较于升降序链表定时器的 O（n）时间复杂度要优越。对于执行效率来说，升序链表和最小堆定时器都是 O（1）比时间轮定时器要好，为此系统采用最小堆位时间定时器。

6）本系统设计的日志系统，采用 4 级日志实时输出系统日志，降低了调试难度，提高了程序的可靠性。

表 1　传统 TLD 算法与本文优化算法各个过程时间对比(ms)

耗时 / 算法	跟踪过程	学习过程	检测过程			总时间
			方差分类器	集合分类器	最近邻分类器	
传统 TLD	11.4056	24.7692	183.461	22.1017	181.586	423.3235
本文优化算法	10.0828	1.88427	47.9121	7.78709	53.4198	121.0861

7）为了使不同读写格式或不同平台能互相兼容、保证数据传输的完整性、提高传输的准确性，使用 der 编码的方式，对图像进行 der 格式的编码，转成统一的二进制格式。客户端通过读取到数据后进行 der 编码传输，服务器端接受到数据后进行 der 解码并显示。

3　实验结果

实验硬件平台为 NVIDIA Jetson TK1 嵌入式开发板，软件环境为 Ubuntu 14.04.4＋opencv 2.4.13＋CUDA6.0。服务器端位于局域网内，配置为 ubuntu 64 位操作系统，intel core i7－4710HQ CPU@2.5 GHz，8 G 内存，IP 地址为 192.168.92.134，网络环境如图 3 所示。TK1 作为客户端，采用 intel7260 网卡作为数据传输媒介，使用传输端口为 8000，以二进制形式实时的将图片传送到服务器端，客户端网络环境如图 4 所示。

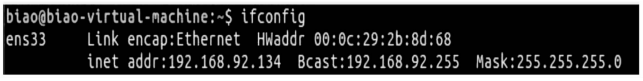

图 3　服务器端网络环境

图 4　客户端网络环境

实验中选用书本作为跟踪目标，与传统 TLD 算法运行时间（ms）对比如表 1 所示，实验结果表明，本文优化的 TLD 算法在保证了传统的 TLD 算法跟踪效果的情况下，大幅的缩短了时间。

4　结论

本文通过对传统 TLD 算法进行分析理解，将其移植到 NVIDIA Jetson TK1 嵌入式开发板中，并分别使用 CUDA 平台和距离聚类算法对方差分类器和集合分类器进行优化。实验结果表明，相比于传统的 TLD 算法，本文的优化算法的实时性更强。此外，本文设计了 C/S 服务器用于远程监控目标跟踪效果，配合嵌入式开发板，使得整个系统的应用范围更加广泛。

参考文献：

[1] 王　鑫，徐立中. 图像目标跟踪技术［M］. 北京：人民邮电出版社，2012.

[2] 梁　敏，刘贵喜，基于自适应混合滤波器的多目标跟踪算法［J］. 光学学报，2010，30（9）：2554－2561.

[3] 王坤峰，李镇江，糖淑明. 基于多特征融合的视频交通数据采集方法［J］. 自动化学报，2011，37（3）：322－330.

[4] Kalal Z，Mikolajczyk，Matas J. Tracking learning dectection［J］. IEEE transactions on Pattern Analysis and Machine Intelligence，2012，34（7）：1409－1422.

[5] 陈义军，沈　航，张进明，等，基于视频传感器网络的目标跟踪模型研究［J］. 计算机工程与设计，2011，32（12）：3933－3937.

[6] 王相海，方玲玲，丛志环. 基于 MSPF 的实时监控多目标跟踪算法研究［J］. 自动化学报，2012，38（1）：139－142.

[7] 孔　军，汤心溢，蒋　敏，等. 基于多尺度特征提取的 kalman 滤波跟踪［J］. 红外与毫米波学报，2011，30（5）：446－450.

[8] Michael Kass，Andrew Witkin，Demetri Terzopoulos. Snakes：Active contour models［J］. International Journal of Computer Vision. 1988（4）.

[9] Cheng Y. Mean shift，mode seeking and clustering［J］. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1995.

[10] 唐峥远，赵佳佳，杨　杰，等. 基于稀疏表示模型的红外目标跟踪算法［J］. 红外与激光工程，2012，41（5）：1389－1395.

[11] Zdenek Kalal，Krystian Mikolajczyk，Jiri Matas. Tracking－Learning－Detection［J］. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2012.

[12] Kalal Z，Mikolajczyk K，Matas J. Face－TLD：Tracking－Learning－Detection applied to faces［A］. IEEE International Conference on Image Processing［C］. IEEE，2010：3789－3792.

[13] Kalal Z，Mikolajczyk K，Matas J. Forward－Backward Error：Automatic Detection of Tracking Failures［A］. International Conference on Pattern Recognition［C］. IEEE Computer Society，2010：2756－2759.

[14] Kalal Z，Matas J，Mikolajczyk K. Weighted Sampling for Large－Scale Boosting［A］. British Machine Vision Conference 2008［C］. Leeds，September. DBLP，2008.

[15] Kalal Z，Matas J，Mikolajczyk K. P－N learning：Bootstrapping binary classifiers by structural constraints［J］. Computer Vision and Pattern Recognition. IEEE，2014：49－56.

[16] Kalal，Z，Matas，J，Mikolajczyk，K. Online learning of robust object detectors during unstable tracking［A］. IEEE，International Conference on Computer Vision Workshops［C］. IEEE，2009：1417－1424.