

基于嵌入式平台的低时间复杂度目标跟踪算法

王向军¹, 郭志翼¹, 王欢欢²

(1. 天津大学 精密测试技术及仪器国家重点实验室, 天津 300072;

2. 天津科技大学 机械工程学院, 天津 300222)

摘要: 针对嵌入式平台往往算力受限的应用背景, 提出了一种低时间复杂度的、适用于复杂场景的目标跟踪算法——CTSTC 算法。算法由自适应更新的时空上下文目标跟踪环节和自适应更新的压缩感知目标辅助定位环节两部分构成, 当时空上下文跟踪结果不可靠时, 启动压缩感知目标辅助定位环节, 如果辅助定位后的结果可靠, 则采用辅助定位结果校正时空上下文跟踪环节。算法运行速度与时空上下文算法(STC)接近, I5CPU 下测试可达每秒 1 577 帧, 远高于其他常用算法, 是一种运算速度极高的目标跟踪算法, 但算法在复杂环境下的鲁棒性却有所提升。使用 OTB2013 数据集进行测试, 较 STC 算法, CTSTC 精度提升 12.8%, 成功率提升 27.5%。算法在以 DM6437 为核心的小型目标跟踪系统上进行测试, 可以实现实时稳定跟踪。

关键词: 目标跟踪; 低时间复杂度; 嵌入式; 辅助定位

中图分类号: TP391.4 文献标志码: A DOI: 10.3788/IRLA201948.1226001

Low time complexity target tracking algorithms based on embedded platform

Wang Xiangjun¹, Guo Zhiyi¹, Wang Huanhuan²

(1. State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China;

2. College of Mechanical Eng., Tianjin University of Science and Technology, Tianjin 300222, China)

Abstract: Aimed at the application background of embedded platform which is often limited in computing power, this paper proposed a low time complexity target tracking algorithm, CTSTC algorithm, which was suitable for complex scenes. The algorithm consisted of two parts: the main part constituted by the spatio-temporal context target tracking based on adaptive update of model and the aided target location part constituted by compressive tracking based on adaptive update of model. When the results of spatio-temporal context tracking were unreliable, the aided location part was activated. If the results of aided location were reliable, the aided location result was used to correct the spatio-temporal context tracking part. The running speed of the algorithm was close to that of the spatio-temporal context learning algorithm (STC). The test on I5CPU can reach 1 577 frames per second, which was much faster than other commonly used algorithms. It was a very fast target tracking

收稿日期: 2019-07-11; 修订日期: 2019-08-21

基金项目: 国家自然科学基金(51575388)

作者简介: 王向军(1955-), 男, 教授, 博士, 主要从事光电传感与测试、计算机视觉与图像分析、微小光机电系统及 MEMS 方面的研究。Email: xdocujw@vip.163.com

algorithm, but the robustness of the algorithm in complex environments was improved. Using OTB2013 data set to test, compared with STC algorithm, CTSTC accuracy increased by 12.8%, success rate increased by 27.5%. The algorithm is tested on a small target tracking system with DM6437 as the core, which can achieve real-time stable tracking.

Key words: target tracking; low time complexity; embedded; aided location

0 引言

目标跟踪是计算机视觉研究的热门领域,在军事与民用方面都有重要的应用价值,如武器导引、智能交通系统。目前人们关注度较高的目标跟踪算法主要有两类,一类是相关滤波类目标跟踪算法,另一类是深度学习类目标跟踪算法。这两类目标跟踪算法发展迅速,涌现了很多跟踪准确而且鲁棒性好的算法。在相关滤波方面,2012年,Henriques等^[1]提出了核检测跟踪算法(CSK),将循环矩阵引入相关滤波目标跟踪框架,大大提升了样本数量,提升了相关滤波类目标跟踪器的性能。2018年,Bhat G等^[2]提出了UPDT算法,分析了深度特征与传统特征的特点,对深度模型与浅层模型分别训练,并在整个模型后期进行融合。在深度学习方面,2016年,Bertinetto L等^[3]提出了基于全连接孪生网络的目标跟踪算法(SiameseFC),将Siamese网络嵌入到跟踪算法中,采用离线学习的方法,速度较快。2018年,Bo Li^[4]和Zheng Zhu^[5]等提出了SiamRPN算法和DaSiamRPN,SiamRPN在SiameseFC算法中引入RPN网络,将跟踪任务构造成局部单样本检测任务,DaSiamRPN是对SiamRPN的改进,通过设计采样策略解决了训练过程中样本不均衡的问题。

人们通过设计更好的图像特征,构造更好的目标函数来提升算法的性能,然而随着算法性能的提升,算法的速度也越来越慢。如果采用这些算法,在很多嵌入式系统,平台的算力将不足以支持实时目标跟踪的实现。考虑到应用平台的限制,我们需要研究一种低时间复杂度的目标跟踪算法,并使其尽可能实现复杂环境下的稳定跟踪。

目标跟踪算法需要考虑遮挡、形变、背景杂斑、尺度变换、光照、低分辨率、运动模糊、快速运动、超出视野、旋转等多种情况^[6],为了综合评定算法的效果,文中采用OTB2013测试框架对算法进行测试。

经OTB2013数据集测试,时空上下文算法(STC)执行速度远超其他算法,文中考虑以STC算法为基础,研究适用于嵌入式平台的低时间复杂度的目标跟踪算法,并使算法在复杂场景下尽可能稳定跟踪。因为STC算法的跟踪依赖于前一帧的结果^[7],所以在场景复杂时容易丢失目标。压缩感知目标跟踪算法(CT)可在一定区域内搜索目标^[8],其速度也名列前茅。为了在STC算法跟踪不可靠时重新锁定目标,文中采用CT算法进行辅助定位,提出了CTSTC算法,它是一种鲁棒性较强的低时间复杂度的目标跟踪算法。文中在OTB2013上对所提出的算法进行了测试,并将算法移植到DSP平台实现了工程应用。

1 文中算法

文中算法在三个方面对现有算法进行了改善:(1)提出了一种STC算法与CT算法的融合框架,满足了高速稳定的目标跟踪的需求;(2)设计了STC算法的自适应更新策略,降低了STC模型漂移的可能;(3)设计了CT算法更新策略,满足了CT算法辅助定位的应用需求。

CTSTC算法的主要框架如图1所示。算法包括自适应更新的STC目标跟踪环节与CT辅助定位环节两部分。当STC算法跟踪结果不可靠时,使用CT算法进行辅助定位,确认目标位置。具体的辅助定位过程如图1所示,取可靠的时空上下文模型与当前的图像信息计算评估置信图,当求得的评估置信图的最大响应小于设定阈值时,算法启动辅助定位环节,使用CT算法,以可靠的目标图像为模板,搜索目标,以搜索到的目标位置重新计算评估置信图,如果最大响应超过数倍的原最大响应,则采用CT算法重定位后的坐标作为目标位置,否则仍以STC算法计算的位置作为目标位置。

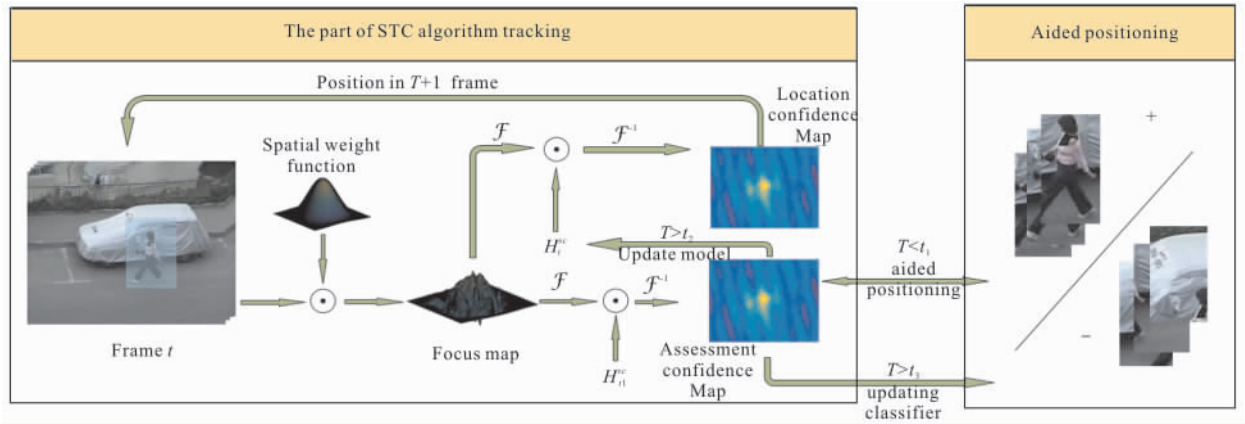


图 1 CTSTC 算法示意图

Fig.1 Schematic diagram of CTSTC algorithm

1.1 自适应更新的时空上下文目标跟踪

文中使用时空上下文算法作为主跟踪环节的算法,但在实物实验中发现 STC 算法的尺度变换部分并不能很好地体现目标的尺度变换,反而会降低跟踪效果。从 STC 算法的理论分析,可以得到同样的结论,STC 算法尺度信息的更新依赖于两帧间置信图最大响应值的比值的平方根,在某些场景下,该处理方式确实可以较好地反映尺度变换,但在某些场景下,背景干扰会使该比值无法正确反映尺度变换。经 OTB2013 数据集测试,取消尺度变换的 STC 算法精度比原算法提升了 1.5%,所以 CTSTC 算法去除了 STC 算法的尺度变换部分。而采样尺度滤波器^[9]的方法会提高算法的时间复杂度,为了保证性能,CTSTC 算法将尺度变换部分剔除。

1.1.1 空间上下文模型

时空上下文算法将目标跟踪视为求解目标在图像各个位置的概率问题^[7]。利用贝叶斯公式,将该概率问题分解为求解先验概率与条件概率这两个子问题,如公式(1)所示。

$$c(x) = \sum_{c(z) \in X^c} P(x|c(z), o)P(c(z)|o) \quad (1)$$

式中: $x \in R^2$ 为目标位置; $X^c = \{c(z) = (I(z), z) | z \in \Omega_c(x^*)\}$ 是上下文特征集, $I(z)$ 表示图像中位置 z 处的灰度值, $\Omega_c(x^*)$ 是 x^* 的邻域; o 表示目标出现在视场; $c(x)$ 为置信图,表示目标位于位置 x 的可能性; $P(x|c(z), o)$ 是条件概率部分; $P(c(z)|o)$ 是先验概率部分。

条件概率部分反映的是目标位置与周围背景的上下文关系,可以建立空间上下文模型,如公式(2)

所示,该模型可以反应目标位置 x 与点 z 之间的关系。并且由于 $(x-z)$ 部分可以区分 x 相对 z 的方向。

$$P(c(z)|o) = h^{sc}(x-z) \quad (2)$$

先验概率部分反映的是目标的背景信息,可以建立上下文先验模型,如公式(3)所示, $I(z)$ 表示 z 处图像的灰度值, ω_σ 是权重函数,其意义是距离目标越近的地方,权重越大,对于跟踪目标越重要。

$$P(c(z)|o) = I(z)\omega_\sigma(z-x^*) \quad (3)$$

$$\omega_\sigma(z) = ae^{-\frac{|z-x^*|^2}{\sigma^2}} \quad (4)$$

式中: a 为归一化常数; σ 为尺度参数。

对于表示目标所在位置可能性的置信图,目标实际的位置应为概率最大的点,距离目标越远的位置应该概率越低,因此可以构造置信图的模型如公式(5)所示。

$$c(z) = be^{-\left|\frac{x-x^*}{\alpha}\right|^\beta} \quad (5)$$

式中: b 为归一化常数; α 和 β 为模型参数。

联立公式 (1)~(3)、(5) 求解空间上下文模型,并根据卷积定理,引入傅里叶变换加速计算,可得到空间上下文模型的解析式如公式(6)所示。

$$h^{sc}(x-z) = F^{-1} \left\{ \frac{F \left(be^{-\left|\frac{x-x^*}{\alpha}\right|^\beta} \right)}{F(I(z)\omega_\sigma(z-x^*))} \right\} \quad (6)$$

式中: F 为傅里叶变换; F^{-1} 为逆傅里叶变换。

1.1.2 自适应更新策略及目标定位

在跟踪过程中,STC 算法每处理一帧图像便更新一次模型,当遇到遮挡、快速移动等情况时便会发

生模型漂移直至丢失目标,而更新不及时则会使算法无法及时适应目标变形的情况,进而丢失目标。为了实现更好的跟踪效果,需要自适应更新模型,避免更新过快或不及时的情况,为此笔者需要采用一种评价准则,评价当前跟踪的可靠性,在可靠时更新模型,不可靠时停止更新。Lukezic A 等^[10]用峰值旁瓣比的方法来判定不同通道跟踪的可靠性,Bhat G^[21]等人提出了同时反映准确性和鲁棒性的置信图质量评估方法,然而 Bhat G 等的方法计算量过大,不适合应用于本算法。跟踪失败意味着丢失目标,本算法的目标在于在目标丢失前通过辅助定位得到更高的跟踪精度和目标丢失后通过辅助定位重新锁定目标,其中重新锁定目标是本算法着重关注的功能,为此,辅助定位评价方法必须能够反映丢失目标的情况,为此,可以选取典型的能够体现稳定跟踪与丢失两种情况的图像序列进行测试,以 Biker 与 Jogging 序列测试结果为例,在 Biker 序列的前段,STC 算法可以稳定跟踪,在 66 帧时,跟踪失败。在 Jogging 序列的 69 帧前,算法可以稳定跟踪,第 69 帧时,算法丢失目标。计算处理 Biker 序列与 Jogging 序列的峰值旁瓣比,得到变化曲线如图 2 所示,由图可知 STC 算法求得的置信图的峰值旁瓣比波动较大,丢失的情况并未

很好地体现在曲线中,所以峰值旁瓣比评价准则不能很好体现跟踪的效果,不适合应用于本算法。

为了确保运算速度,文中算法也不适合采用 Bhat G 等人的方法。在跟踪环节中,定位置信图反映的是当前场景下目标出现在各个像素点的概率,为了发挥时空上下文算法抗背景干扰的优势,更好地评估跟踪情况,文中使用输入的图像信息与可靠的时空上下文模型计算评估置信图。由于置信图表示的是目标出现在各点的概率,最大响应值本身便具备着可靠性信息,为了避免额外的计算,所提算法尝试直接取评估置信图的最大响应值作为评价准则,同样采用 Biker 序列和 Jogging 序列进行测试,得到最大响应值变化曲线如图 3 所示,曲线能够反映丢失的情况,能够较好地反映跟踪效果,故文中提出采用算法中求得

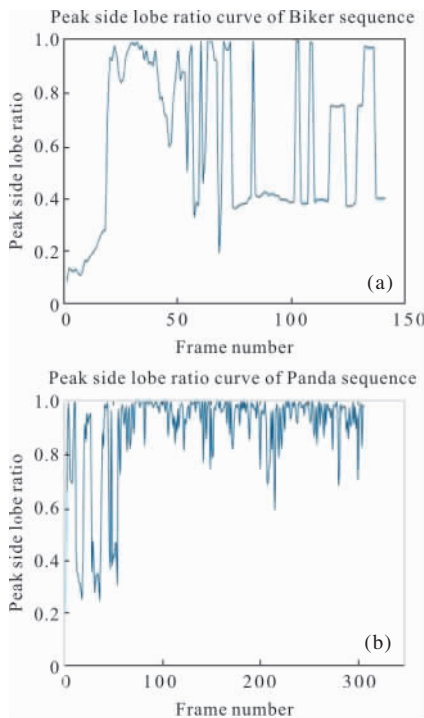


图 2 峰值旁瓣比曲线

Fig.2 Peak side lobe ratio curve

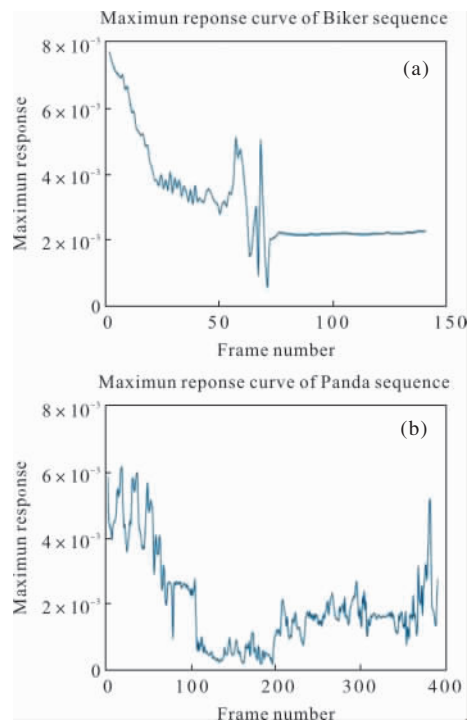


图 3 最大响应值曲线

Fig.3 Maximum response curve

响应超过阈值时才对模型进行更新,更新方法如公式(7)所示。模型更新的阈值可以在权衡算法在实际应用场景的实验数据后进行设置,增大阈值可以提升算法执行效率和增强算法的抗干扰能力,但会使模型更新缓慢,从而使算法无法适应目标快速变化

的情况,减小阈值可以适应目标快速变化的场景,但会使模型对目标更新更为敏感,从而降低算法的抗干扰能力和执行效率。

$$\begin{cases} H_{t+1}^{stc}=(1-\rho)H_t^{stc}+\rho h_t^{sc} & T>t_2 \\ H_{t+1}^{stc}=H_t^{stc} & T\leq t_2 \end{cases} \quad (7)$$

式中: H_t^{stc} 为第 t 帧的时空上下文模型; h_t^{sc} 为第 t 帧的空间上下文模型。第 1 帧的时空上下文模型等于第 1 帧的空间上下文模型。式中, $T=\max(c_{t+1})$ 是置信图的最大响应。

已知第 $t+1$ 帧的时空上下文模型和图像信息及第 t 帧的目标位置,便可求出第 $t+1$ 帧的置信图,置信图中的最大值所在位置即为目标位置,如公式(8)所示。

$$x_{t+1}^* = \operatorname{argmax}\left(F^{-1}\left(F\left(H_{t+1}^{stc}(x)\right)\otimes F\left(I_{t+1}(x)\omega_\sigma(x-x_t^*)\right)\right)\right) \quad (8)$$

式中: x_{t+1}^* 为第 $t+1$ 帧的目标位置; x_t^* 为第 t 帧的目标位置。

1.2 压缩感知辅助定位

1.2.1 辅助定位策略

在跟踪不准确时,需要启动辅助定位环节,重定位目标位置。为了实现重定位的效果,辅助定位环节必须保留可靠的目标信息,并且在区域内有较强的搜索能力。

辅助定位环节可采用更新缓慢的 STC 算法、SVM 算法、CT 算法等,但采用 STC 算法辅助定位的效果不佳,无法完成重定位任务,如典型的反映目标遮挡的 jogging-1 序列,当目标重新出现时,STC 算法并不能重定位成功。SVM 算法典型的应用是 Struck 算法,跟踪效果较好,但 SVM 算法计算量较大,会影响算法速度,无法满足需求。CT 算法是经典的稀疏表示目标跟踪算法,计算量较小,在区域内的搜索能力也较强,为此,文中使用 CT 算法进行辅助定位,因为 CT 算法既要保留可靠的目标模型,又要适应目标物本身的变化,所以 CT 算法的模型需要自适应更新。

1.2.2 压缩感知定位

压缩感知目标跟踪算法先使用测量矩阵对原始图像特征进行降维,如公式(9)所示,然后将降维后的特征输入朴素贝叶斯分类器^[8]。由分类器可以得

到样本是目标的概率,其中概率最大的就是下一帧目标所在的位置。

$$V=RX \quad (9)$$

式中: $R\in R^{m\times n}$ ($k\ll n$)为测量矩阵,需满足压缩感知的 RIP 条件; $X\in R^n$ 为图像的 Haar-like 特征; $V\in R^{k\times 1}$ 为压缩后的特征值。根据 Johnson-Lindenstrauss 引理, R 可取以下形式:

$$R_{ij}=\sqrt{s}\times\begin{cases} 1 & P=\frac{1}{2s} \\ 0 & P=1-\frac{1}{s} \\ -1 & P=\frac{1}{2s} \end{cases} \quad (10)$$

当 $s=O(n)$ 时,测量矩阵渐进满足正太分布,当 $s=n/\log(n)$ 时,测量矩阵几乎和传统的随机投影矩阵一样准确,文中取 $s=n/4$ 。这意味着每一行中仅提取 c 个数据(c 个 4),得到的特征数据仅有 ck 个,将大大减少计算量,最终得到的特征为 $V=\{v_1, \dots, v_k\}^T$ 。

由于 Diaconis 等人证明高维随机向量的随机投影几乎是高斯分布的^[11],所以算法中的朴素贝叶斯分类器可表示为公式(11)。

$$H(v)=\log\frac{\prod_{i=1}^n p(v_i|y=1)p(y=1)}{\prod_{i=1}^n p(v_i|y=0)p(y=0)} = \sum_{i=1}^n \log\left(\frac{p(v_i|y=1)}{\prod_{i=1}^n p(v_i|y=0)}\right) \quad (11)$$

$$\text{其中,}\begin{cases} p(v_i|y=1)\sim N(\mu_i^1, \sigma_i^1) \\ p(v_i|y=0)\sim N(\mu_i^0, \sigma_i^0) \end{cases}。$$

压缩算法的模型需要自适应更新,为了不增加额外的计算,可以利用 STC 算法目标跟踪环节求得的评估置信图的最大值,当最大值大于阈值时更新压缩感知辅助定位环节,更新公式如公式(12)、(13)所示。

$$\begin{cases} \mu_{t+1}^1=\lambda\mu_t^1+(1-\lambda)\mu^1 & T>t_3 \\ \mu_{t+1}^1=\mu_t^1 & T\leq t_3 \\ \mu_{t+1}^1=\sqrt{\lambda(\sigma_t^1)^2+(1-\lambda)(\sigma^1)^2+\lambda(1-\lambda)(\mu_t^1-\mu^1)^2} & T>t_3 \\ \mu_{t+1}^1=\sigma_t^1 & T\leq t_3 \end{cases} \quad (12)$$

$$\begin{cases} \mu^1 = \frac{1}{n} \sum_{k=0}^{n-1} v_i(k) \\ \sigma_i^1 = \sqrt{\frac{1}{n} \sum_{k=0}^{n-1} (v_i(k) - \mu^1)^2} \end{cases} \quad (13)$$

式中： μ_i^1 为正样本的第 i 帧的均值； σ_i^1 为正样本的第 i 帧的方差； T 为置信图的最大响应； t_3 为更新阈值。

辅助定位环节以压缩感知目标跟踪为主干算法，辅助定位的结果不一定可靠，为此，需对跟踪结果进行检验，为了发挥时空上下文算法抗背景干扰的优势，可以对辅助定位位置重新计算评估置信图，最大响应值大于数倍的原最大响应值，说明辅助定位明显改善了跟踪结果，目标位于辅助定位位置的概率较大，可采用辅助定位结果进行辅助定位。否则，仍采用自适应更新的 STC 算法的跟踪结果。

算法为了平衡在各个场景中的性能，难免在某些场景下出现辅助定位结果不可靠的情况，如对 deer 序列的实验，实验结果如图 4 所示。其中，蓝线为加可靠性判断的跟踪结果，灰线为无可靠性判断

跟踪结果，橘色线为真实位置，黄线为可靠性判断结果。图中黄线值为 125 时表示辅助定位未触发，值为 200 时表示辅助定位可靠，值为 50 时表示辅助定位不可靠。由图 4 可知，当采用可靠性判断时，算法避免了误判导致跟踪失败的情况。

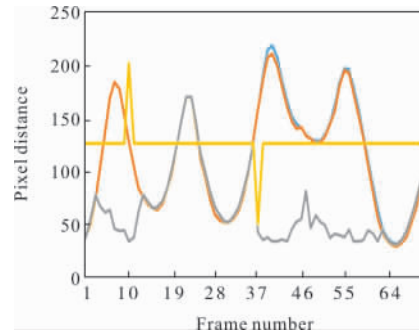


图 4 实验结果图

Fig.4 Figure of experimental results

1.3 算法流程

文中的 1.1 节与 1.2 节分别介绍了自适应更新的时空上下文目标跟踪环节和压缩感知辅助定位环节，CTSTC 算法正是由这两个环节融合得到，算法的具体流程如图 5 所示。

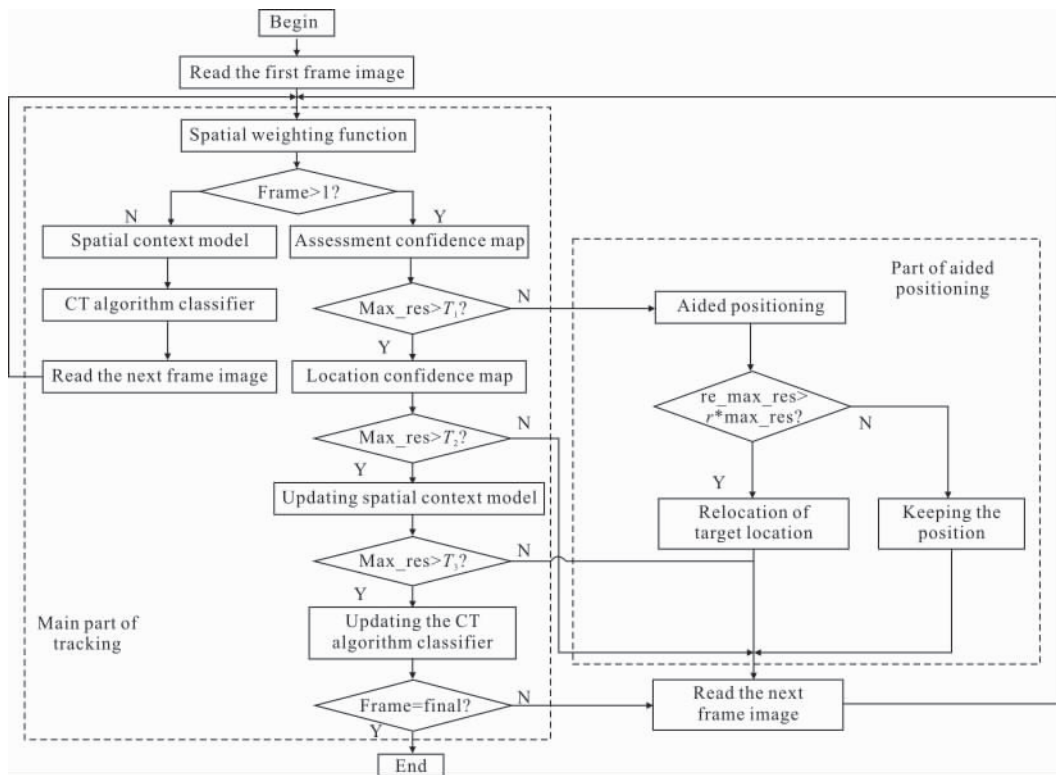


图 5 CTSTC 算法流程图

Fig.5 Flow chart of CTSTC algorithm

2 实验

2.1 实验平台

文中算法的最终应用平台是嵌入式系统,但在嵌入式系统无法严格评估算法的效果,而基于 PC 平台的目标跟踪算法评估较为成熟,有 OTB2013 等目标跟踪数据集,可以反映算法在不同场景下的跟踪效果。为此,文中算法先在 PC 平台进行仿真实验,评估算法的跟踪效果和时间复杂度,然后移植到嵌入式系统。

仿真实验使用的 PC 平台为 XIAOXIN CHAO7000-13 计算机,处理器为 Inter (R) Core (TM) i5-8250U CPU@1.60 GHz 1.80 GHz,内存为 8 GB,编程环境为 MATLAB R2016a。

2.2 算法参数设置

CTSTC 算法涉及的辅助定位阈值 T1 为 0.001、空间上下文模型更新阈值 T2 为 0.0019、CT 算法贝叶斯分类器更新阈值 T3 为 0.0071,CT 算法正样本的范围 trparams.init_postrainrad 为 2.0,STC 算法 alpha 值为 2.02。

2.3 实验框架

算法采用 OTB2013 进行测试,该数据集包含 51 个视频序列,从光照变化、平面外旋转、尺度变化、遮挡、形变、模糊、快速移动、平面内旋转、超出视场、背景杂斑、低分辨率等几个场景对算法展开评估。评价指标包括准确度图与成功率图。

2.4 实验结果

该算法的应用平台是算力不充裕的嵌入式系统,算法时间复杂度是必须重点考虑的指标,文中采用帧率来进行衡量,对几个典型的图像序列的处理帧率进行了对比,如表 1 所示。

表 1 运算帧率对比表(单位:fps)

Tab.1 Operational frame rate contrast table (Unit: fps)

	CTSTC	CSK	STC	CT
CarDark	1 577	855	1 620	258
Girl	1 030	467	1 003	198
Coke	766	428	745	120
Deer	387	236	410	164
Jogging-2	551	264	522	240

由表 1 可知,对于相同图像序列,CTSTC 算法的处理帧率远高于 CSK 算法,是一种低时间复杂度的目标跟踪算法。在保证算法低时间复杂度的同时,本算法对复杂环境的鲁棒性较 CSK、STC、CT 等算法有明显提升,Jogging1 是典型的能够体现目标遮挡、形变、平面外旋转的图像序列。图 6 的 Jogging1-(a) 中,CTSTC、CSK、STC、CT 算法均可跟踪目标;图 6 的 Jogging1-(b)中,目标被障碍物完全遮挡,四种算法均丢失目标;图 6 的 Jogging1-(c)中,目标物重新出现,CTSTC 算法重定位成功,CSK、STC、CT 算法丢失





motorRolling-(a) motorRolling-(b) motorRolling-(c) motorRolling-(d)

图 6 跟踪效果图,绿色框、红色框、蓝色框、黄色框分别为 CTSTC 算法、STC 算法、CSK 算法、CT 算法的跟踪效果

Fig.6 Figure of tracking effect, green box, red box, blue box and yellow box are CTSTC algorithm, STC algorithm, CSK algorithm and CT algorithm respectively

目标;图 6 的 Jogging1-(d)中,CTSTC 算法开始稳定跟踪目标。Coke 图像序列体现了光照变化、变形、遮挡、旋转等场景,CTSTC 算法可以稳定跟踪。Woman 序列体现了变形、遮挡、平面外旋转、尺度变化等场景,CTSTC 可以全程稳定跟踪,而 STC、CT、CSK 算法均丢失目标。motorRolling 序列体现了模糊、旋转、快速移动,形变、尺度变换等场景,CTSTC 算法也出现了丢失目标的情况,但在后续的跟踪中多次重定位成功。

为体现算法在不同场景下的性能,文中采用 OTB2013 测试框架进行测试,测试指标包括精度(Precision)和成功率(Success),精度以欧式距离的方式衡量预测位置中心点与实际中心的的偏离程度,成功率衡量预测目标与实际目标的重合程度,包含位置与尺度信息。OTB2013 数据集测试结果如图 7 所示,

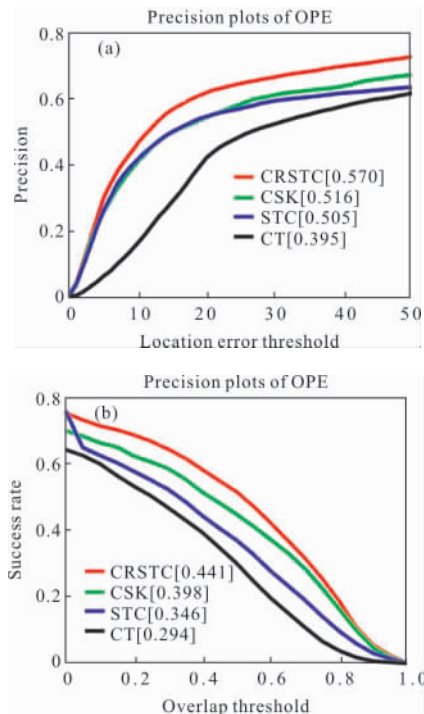


图 7 精度与成功率测试结果

Fig.7 Test results of precision and success rate

从图中可以看出,CTSTC 算法性能有显著改善。

3 嵌入式平台下算法的实现及测试

文中所使用的嵌入式平台是以 DM6437 芯片为核心的小型化目标跟踪系统,该芯片是 TI 公司生产的达芬奇系列 32 位定点数字信号处理器,主频 600 MHz,具有低成本、低功耗的特点,是一款具有代表性的 DSP 芯片。系统的硬件系统包括 DM6437、FPGA、DDR2、FLASH、CMOS、上位机等部分,FPGA 接受来自 CMOS 的图像数据,并对图像进行预处理,消除光照变化的影响,然后将处理过的图像传给 DM6437。DM6437 接收上位机通过 422 传来的目标坐标,对 FPGA 传入的图像进行处理,进行目标跟踪,跟踪结果经 422 传给上位机。硬件部分结构框图如图 8 所示。

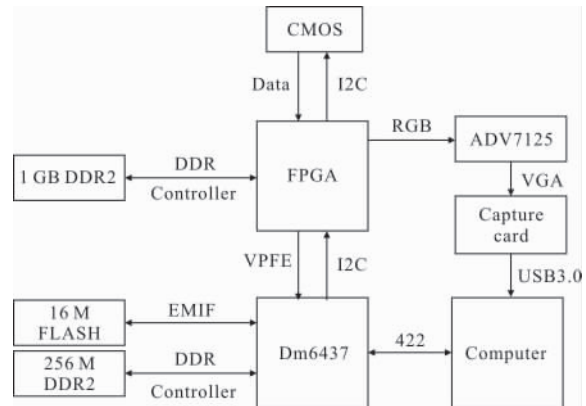


图 8 硬件结构图

Fig.8 Hardware architecture

算法移植时需要结合 DM6437 芯片的特点对程序进行修改,DM6437 是定点芯片,进行浮点计算时会大大增加计算时间,在移植时,需将浮点数转化为 IQ 类型,这样便可基于 TI 公司开发的 IQmath 库进

行计算,其速度大大提升。但对于 IQ 类型的的数据而言,取值范围越大,精度越低,因此需要在移植时综合考虑算法中变量的取值范围与需要的精度,保证算法效果。在 CT 算法辅助定位环节,需要计算全图的积分图,文中使用 TI 提供的 VLIB 视觉代码库,高效地完成计算。

由于算法中涉及的循环较多,为了提升循环体的运行速度,移植时添加预处理指令 #pragma MUST_ITERATE()将循环展开。算法中有些数据需要频繁使用,可以结合 DM6437 存储结构的特

点,将这些数据放置在片内,来减少访问时间。此外,对于算法本身,可以将不需重复计算的内容移至循环体外,尽量不在循环语句中嵌套判断语句,使用指针传递数据等方式改善算法的结构。

经过优化,自适应更新时空上下文跟踪部分消耗时钟数为 11.36 M,压缩感知辅助定位部分消耗时钟数为 13.48 M。在频繁辅助定位时,帧率为 24.84 帧,满足实时性的要求。算法移植后,进行了目标跟踪实验,实验图像如图 9 所示,实验结果显示,该算法可以实现稳定跟踪。

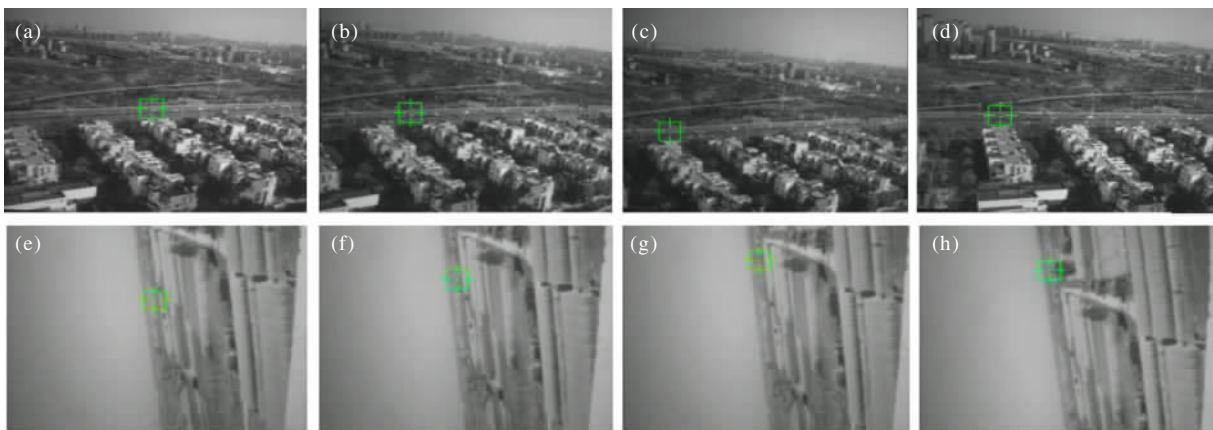


图 9 嵌入式平台跟踪效果图

Fig.9 Tracking effect diagram of embedded platform

4 结 论

文中提出了一种低时间复杂度的鲁棒性较强的目标跟踪算法,可以在模型漂移、目标丢失后重新定位目标,实现稳定跟踪。算法在 OTB2013 benchmark 框架下进行测试,在保证帧率的前提下,效果优于 CSK、STC 等算法,算法移植到嵌入式系统后,可以实现稳定的跟踪。虽然本算法的效果仍距当前最出色的算法如 ECO、UPDT 等存在一定差距,但算法的运算速度却是远远超过这些算法的,在一些低成本的嵌入式平台下本算法有着广泛的应用前景。

参考文献:

[1] Henriques J F, Caseiro R, Martins P, et al. Exploiting the circulant structure of tracking-by-detection with kernels [C]//European Conference on Computer Vision.

Berlin: Springer, 2012: 702-715.

[2] Bhat G, Johnander J, Danelljan M, et al. Unveiling the power of deep tracking[C]//Proceedings of the European Conference on Computer Vision (ECCV), 2018: 483-498.

[3] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional siamese networks for object tracking[C]//European Conference on Computer Vision. Cham: Springer, 2016: 850-865.

[4] Li B, Yan J, Wu W, et al. High performance visual tracking with siamese region proposal network [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 8971-8980.

[5] Zhu Z, Wang Q, Li B, et al. Distractor-aware siamese networks for visual object tracking [C]//Proceedings of the European Conference on Computer Vision (ECCV), 2018: 101-117.

[6] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark [C]//Proceedings of the IEEE Conference on

- Computer Vision and Pattern Recognition, 2013: 2411–2418.
- [7] Zhang K, Zhang L, Liu Q, et al. Fast visual tracking via dense spatio-temporal context learning [C]//European Conference on Computer Vision. Cham: Springer, 2014: 127–141.
- [8] Zhang K, Zhang L, Yang M H. Real-time compressive tracking [C]//European Conference on Computer Vision. Berlin: Springer, 2012: 864–877.
- [9] Danelljan M, Häger G, Khan F S, et al. Discriminative scale space tracking [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(8): 1561–1575.
- [10] Lukešič A, Vojíř T, Zajc L Č, et al. Discriminative correlation filter tracker with channel and spatial reliability [J]. *International Journal of Computer Vision*, 2018, 126(7): 671–688.
- [11] Diaconis P, Freedman D. Asymptotics of graphical projection pursuit[J]. *The Annals of Statistics*, 1984, 12(3): 793–815.