

# 自适应特征选择的相关滤波跟踪算法

刘万军<sup>1</sup>, 孙虎<sup>2\*</sup>, 姜文涛<sup>1</sup>

<sup>1</sup> 辽宁工程技术大学软件学院, 辽宁 葫芦岛 125105;

<sup>2</sup> 辽宁工程技术大学研究生院, 辽宁 葫芦岛 125105

**摘要** 针对相关滤波方法对快速运动与快速变形的目标跟踪稳定性较差的问题, 提出一种自适应特征选择的相关滤波跟踪算法。利用位置滤波器和颜色概率模型提取候选区域中的基础特征, 对基础特征以不同的权重分配方式进行融合, 得到多个融合特征。对融合特征进行可信度判定, 选择可信度较高的融合特征作为当前帧的跟踪特征, 估计出目标的候选位置。若最高可信度低于可信度阈值, 启动检测器重新检测目标位置, 否则候选位置即为目标最终位置。与此同时, 对目标模型进行更新, 确保模型对目标描述的准确性。在标准数据集 OTB50 和 OTB100 上进行大量实验, 测试结果表明, 所提出的跟踪方法在运动模糊、光照变化、快速运动等条件下具有较高的跟踪准确率和较好的稳健性。

**关键词** 机器视觉; 目标跟踪; 相关滤波; 颜色统计; 尺度变换; 特征选择

中图分类号 TP391.4

文献标识码 A

doi: 10.3788/AOS201939.0615004

## Correlation Filter Tracking Algorithm for Adaptive Feature Selection

Liu Wanjun<sup>1</sup>, Sun Hu<sup>2\*</sup>, Jiang Wentao<sup>1</sup>

<sup>1</sup> School of Software, Liaoning Technical University, Huludao, Liaoning 125105, China;

<sup>2</sup> Graduate School, Liaoning Technical University, Huludao, Liaoning 125105, China

**Abstract** The conventional correlation filtering methods are known to demonstrate poor tracking stability for fast moving and fast deforming targets. Therefore, this paper proposes a correlation filter tracking algorithm for adaptive feature selection. First, the basic features are extracted in the candidate regions using a position filter and a color probability model and fused in different weight combinations to obtain multiple fusion features. Then, the credibility of the fusion features is determined and the features with relatively high credibility are selected as the tracking features of the current frame to estimate the candidate position of the target. Finally, if the maximum credibility is less than the credibility threshold, the detector is activated to redetect the target position; otherwise, the candidate position is just the final position. Meanwhile, the target model is updated to ensure the accuracy of target description. The experimental results on the standard OTB50 and OTB100 datasets show that the proposed tracking method has relatively high tracking accuracy and good robustness under the conditions of motion blurring, illumination variation, and fast motion.

**Key words** machine vision; target tracking; correlation filter; color statistics; scale transformation; feature selection

**OCIS codes** 150.0155; 100.4999; 100.3008; 100.4550

## 1 引 言

目标跟踪是国内外计算机视觉领域的一个研究热点<sup>[1-5]</sup>, 广泛应用于军事领域、民用领域等。现实场景的复杂性(光照变化、形状变化、运动模糊等), 使得目标跟踪的效果不太理想, 因此如何在复杂背

景下提高算法的跟踪准确率是目标跟踪领域的热点问题。

根据目标建模方式的不同, 目前跟踪方法主要分为生成式方法和判别式方法两种。生成式方法主要是计算候选目标与目标模型的相似度, 以相似度最高的候选目标作为目标进行跟踪。生成式方法仅

收稿日期: 2019-01-09; 修回日期: 2019-01-25; 录用日期: 2019-02-19

基金项目: 国家自然科学基金(61172144)、辽宁省自然科学基金(20170540426)、辽宁省教育厅基金(LJYL049, LJ2017QL034, LJ2017ZL003)

\* E-mail: shzxzx@163.com

针对目标本身的外观特征,导致其在处理目标外观快速变化和遇到遮挡时,跟踪效果并不好。判别式方法则是把跟踪看作一个二分类问题,旨在构建一个有效的分类器以区分目标及其背景,从而实现目标的跟踪。相比于生成式方法,判别式方法在建模时将目标与背景进行区分,因此在目标跟踪时的效果通常表现得更为稳健,目前已成为目标跟踪的主流跟踪方式。

近年来,基于相关滤波的判别式方法已广泛应用于目标跟踪领域,并取得了显著的效果。2010年Bolme等<sup>[6]</sup>提出将相关滤波(CF)应用于目标跟踪领域,该算法基于输出结果的最小均方误差(MOSSE)训练相关滤波器,通过响应图像的值判断候选位置与初始化目标的相关性。响应值越大,则相关性越大,从而提高了滤波器的准确度。同时该算法的计算从时域转换到频域,跟踪速率超过了600 frame/s。此后,很多算法都在其基础上进行改进。Henriques等<sup>[7]</sup>于2012年提出核循环结构(CSK)算法,该算法针对相关滤波中样本数量不足对滤波器的影响,使用循环密集采样的方法,有效地利用了整张图片的特征。为了提升算法性能,Henriques等<sup>[8]</sup>在相关滤波中引入核空间,提出了核相关滤波算法(KCF),在算法中采用多通道梯度方向直方图(HOG)特征,通过核函数将线性空间的岭回归问题映射到非线性空间,提高了高维特征空间中样本分类的速度。Danelljan等<sup>[9]</sup>针对尺度变化问题,基于MOSSE跟踪方法提出了多尺度空间滤波跟踪算法(DSST),该算法分为位置滤波器(TF)和尺度滤波器(SF)两部分,采用判别相关滤波器来确定位置,同时在文中提出一种精准的尺度估计方法以确定尺度信息。利用该算法进行跟踪的过程中,两个滤波器独立工作,分别进行目标定位和尺度评估,取得了优异的表现。Bertinetto等<sup>[10]</sup>从特征融合入手,提出一种特征互补的跟踪方法,将HOG特征与颜色特征相融合,有效地解决了变形情况下HOG特征稳健性较差和光照变化下颜色特征表现效果较差的问题,提高了跟踪的准确度,但两种特征融合增加了计算复杂度。除融合特征外,针对人为选择的核函数未必是最优函数的问题,Tang等<sup>[11]</sup>提出了多核相关滤波器(MKCF),即在核化滤波器(KCF)基础上引入多个核函数,并提出了优化求解的方法。Zhang等<sup>[12]</sup>将多个特征进行融合,同时在计算时引入多个核函数,再将多个滤波器进行整合,并通过最后的跟踪结果调整核的权重,从而使

得核有自适应选择的能力。

虽然相关滤波算法凭借其速度的优势和良好的精度得到广泛应用,但也存在不足之处:1)相关滤波方法属于模板类方法,对于快速变形和快速运动的情况跟踪效果较差,易发生跟踪漂移现象;2)在提取目标特征的过程中,提取单一特征或固定权重融合特征不能够全面地描述目标,导致跟踪的准确率较低;3)为了提高跟踪的准确率,一些跟踪方法采用多个滤波器相互融合来确定目标位置,时间复杂度较高。

针对以上问题,本文在相关滤波跟踪的基础上,提出一种自适应特征选择的相关滤波跟踪算法。该方法对HOG特征相关滤波器的响应值与颜色统计特征的响应值分配不同的权重,进行线性加权运算,得到多个具有不同偏向性的融合特征,通过对融合特征进行可信度判定,选取得分最高的融合特征作为跟踪特征,保证了对目标特征描述的准确性。然后对确定的目标位置进行尺度估计和模型更新,实现对目标的稳定跟踪。在OTB50和OTB100的标准数据序列上进行实验,实验结果验证了所提方法的有效性和稳定性。

## 2 总体框架设计

快速多尺度空间滤波跟踪(fDSST)算法<sup>[13]</sup>在跟踪效果和速度上都表现得较好,因此本文算法采用fDSST算法作为基本框架,同时为了更有效、全面地描述目标以及方便后续与其余跟踪算法的比较,采用判别能力较强的HOG特征与颜色统计特征作为基础特征进行融合。跟踪框架主要分为建模、跟踪和更新三个阶段,整体框架如图1所示。

1) 建模阶段。根据手动标定的第一帧目标位置信息,训练用来估计目标位置的相关滤波器,同时训练颜色概率模型和用来估计目标尺度的相关滤波器,最终建立目标模型。

2) 跟踪阶段。首先在当前帧的候选区域提取HOG特征和颜色特征<sup>[14]</sup>,使用前一帧确定的目标模型对两种基础特征进行计算。然后对计算结果分配不同的权重并进行融合,对融合结果进行可信度判断,选取可信度最高的融合特征作为当前帧的跟踪特征,并根据该特征估计出目标的候选位置。对于最高可信度仍低于阈值的情况,根据两帧之间特征点的匹配结果,判断是否启动检测器。若匹配程度较低,启动检测器重新检测和定位目标的位置,否则候选位置即为目标最终位置。最后使用尺度滤波

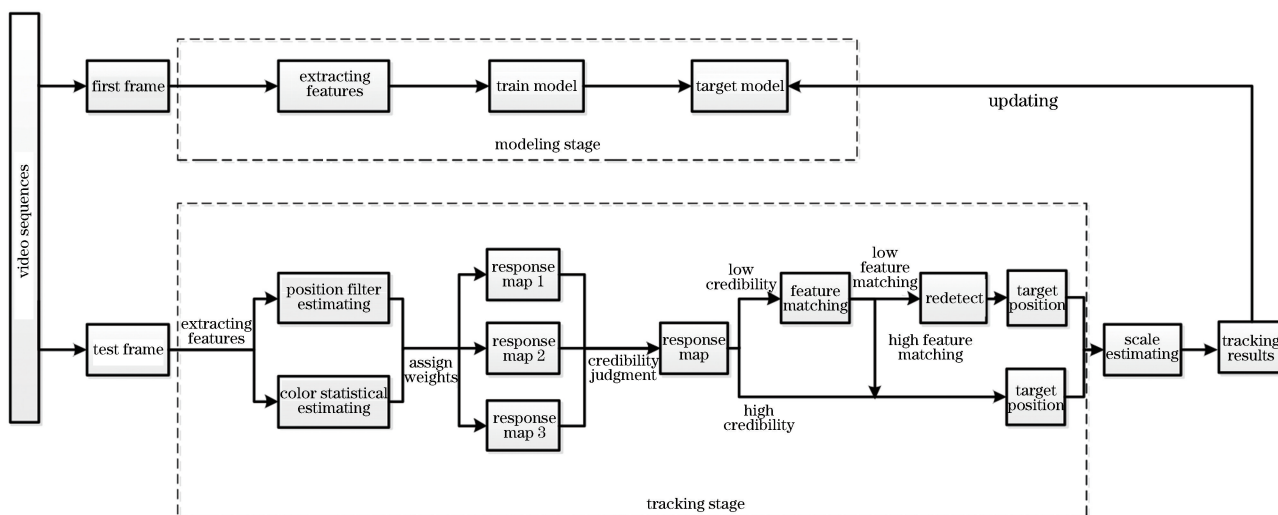


图 1 整体框架示意图

Fig. 1 Schematic of overall frame

器对目标的尺度进行估计,从而完成跟踪过程。

3) 更新阶段。利用每帧跟踪阶段确定的目标位置,对位置滤波器、颜色概率模型和尺度滤波器进行更新。

### 3 自适应特征选择的跟踪算法

相关滤波方法最早在文献[6]中提出,可应用在跟踪领域,是近年来跟踪领域的主流方法。其基本思想是:越是相似的两个目标,其相关值越大,即图片中的目标与初始化目标越相似,滤波器得到的响应值越大,因此将滤波器响应最大的位置作为目标的位置,从而实现跟踪。该类方法属于模板类方法,并受边界效应的影响,对于快速运动与快速变形的目标,效果并不好。文献[10]提出融合两种互补特征的方法,一定程度上改善了相关滤波方法对快速运动和快速变形目标跟踪的情况,同时对光照变化的情况也能表现出较好的跟踪效果。但是该方法采用固定等权重分配特征的方式融合互补的两个特征,对不断变化的目标以及背景没有选择适应能力,且未对跟踪结果进行可靠性判定。在文献[6]和文献[10]的基础上,本文提出一种自适应特征选择的跟踪算法,通过融合不同权重的特征,不仅有效解决了相关滤波器对于快速运动和快速变形目标效果不好的问题,而且增强了跟踪框架对于变化的目标与背景的适应能力。同时对跟踪效果进行可信度判断,可信度较低时启动检测器进行重新检测,重新定位目标的位置,从而有效地提高了跟踪的准确率。

在本文中,位置滤波器和颜色概率模型用于提

取样本特征,得到相对应的响应图。然后通过对两个响应图以线性加权的方式进行组合,得到最终的目标位置响应图。其组合公式为

$$f_{fuse} = \omega f_{color} + (1 - \omega) f_{cf}, \quad (1)$$

式中: $f_{fuse}$ 为融合特征的得分响应函数; $\omega$ 为权重分配因子; $f_{color}$ 为颜色概率模型的得分响应函数; $f_{cf}$ 为位置滤波器的得分响应函数。通过对两个得分函数进行线性加权,最终得到融合特征的响应函数。

通过使用不同的权重分配因子 $\omega$ ,使融合出的特征具有不同的特征偏向性,以应对变化的目标及背景信息。大量实验证明:若权重分配的偏向性太大,则会使跟踪效果和单一特征的跟踪效果相近,不能体现出融合特征的必要性;若权重分配的偏向性太小,则导致跟踪效果与等权重分配方式的跟踪效果相近,不能体现本文选择特征的有效性。因此本文采用实验中表现最为合适的权重分配方式进行特征融合。融合特征的权重分配方式见表1。

表 1 融合特征多权重分配方式

Table 1 Multi-weight distribution of fusion features

Number	Filter weight	Color weight	Total weight
1	0.7	0.3	1
2	0.3	0.7	1
3	0.5	0.5	1

通过融合不同权重的特征得到不同的融合特征响应图,对融合后得到的响应图的可信度进行判断,取可信度最大的特征作为当前帧的跟踪特征。采用文献[15]提出的平均峰值相关能量(APCE)计算跟踪的可信度,计算公式为

$$E_{APCE} = \frac{|F_{\max} - F_{\min}|^2}{\text{mean} \left[ \sum_{w,h} (F_{w,h} - F_{\min})^2 \right]}, \quad (2)$$

式中： $F_{\max}$ 、 $F_{\min}$ 、 $F_{w,h}$  分别代表特征响应图中的最大值、最小值以及对应  $(w, h)$  位置的元素值； $E_{APCE}$  值反映响应图的波动程度和检测目标的置信水平。 $E_{APCE}$  值越大，表明响应图除峰值之外的区域波动较为平滑，目标的峰值响应相对背景响应越高，目标的可信度越高；反之，目标的可信度越低。实验中使

用不同权重分配方式对特征进行融合，计算每种方式的  $E_{APCE}$  值，取最大值的融合特征作为当前帧的跟踪特征，所选择的跟踪特征以及检测跟踪过程如图 2 所示。图中， $w_{\text{hog}}$  表示位置滤波器的权重， $w_{\text{color}}$  表示颜色概率模型的权重， $P_t$  代表第  $t$  帧的目标的中心位置， $M_1$ 、 $M_2$ 、 $M_3$  表示在不同权重分配方式下的可信度。

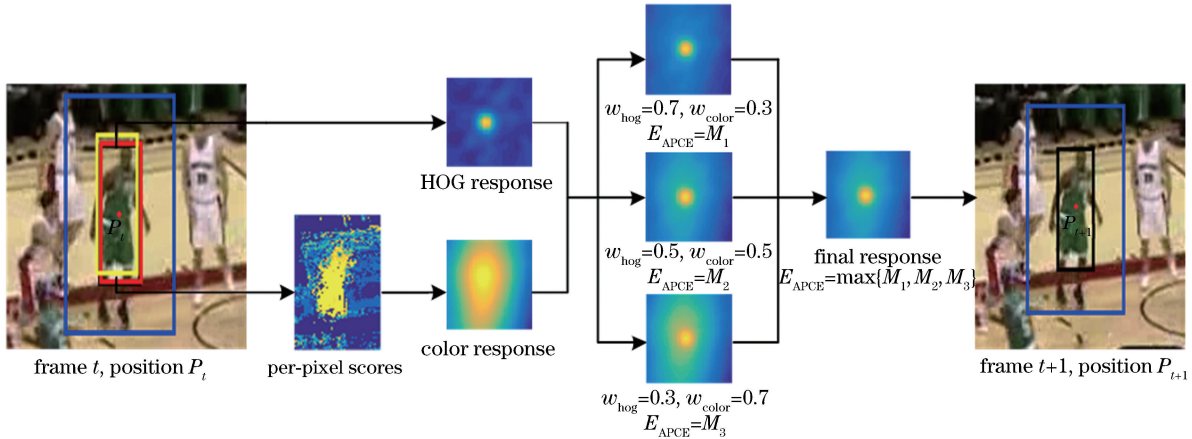


图 2 自适应选择检测跟踪示意图

Fig. 2 Schematic of adaptive selection detection tracking

由于图 2 中的检测跟踪结果可能会因各种因素的影响出现跟踪漂移，实验设定了一个重检测阈值来判断是否需要重新检测跟踪结果。重检测阈值表示可信度历史均值的比例，若重检测阈值设置过大，则会导致目标可信度的可波动范围较小，检测器会误判目标的正常变化，因此需要激活重检测机制，这样就增加了总体跟踪过程的时间复杂度；反之，可信度的可波动范围较大，检测器会漏判目标的异常变化，此时不需要激活重检测机制，这样就降低了整体跟踪的准确率。本文设定重检测阈值  $\theta_{APCE} = 0.6F_{\text{mean}}$ ，其中  $F_{\text{mean}}$  表示  $E_{APCE}$  值的历史均值。 $\Delta F_{APCE}$  代表当前帧的历史均值与最大  $E_{APCE}$  值的差值，表示当前帧目标候选区域与历史目标区域峰值的波动情况。当  $\Delta F_{APCE}$  的值大于重检测阈值时，表明融合特征定位的目标区域的峰值波动发生突变，可能产生了漂移的情况，启动网格运动统计 (GMS) 检测器对目标进行判断重检测。当  $\Delta F_{APCE}$  的值小于重检测阈值时，更新滤波器模型以及颜色概率模型。针对目标的尺度变化问题，采用文献 [13] 中的方法，在候选区域构建目标尺度金字塔，使用尺度滤波器得到最佳的目标尺度。

### 3.1 位置滤波器与颜色概率模型

位置滤波器与颜色概率模型主要是用来提取特征，以获取相应的响应图。其中位置滤波器主要用于提取 HOG 特征，颜色概率模型主要用于统计搜索区域内的颜色特征。HOG 特征用于快速运动和快速变形的情况时效果不好，但用于运动模糊光照变化等情况时比较好；而颜色统计特征不属于相关滤波框架，不受边界效应的影响，用于快速运动和快速变形的情况时效果较好，但光照变化对其影响较大。因此，选择这两种互补的特征作为融合特征的基础特征。

#### 3.1.1 位置滤波器

位置滤波器主要通过使用 fDSST 算法提取搜索区域中的 HOG 特征，并使用相关滤波器将其转化为响应图。在响应图中，峰值代表该位置的响应分数最高，表示该位置是目标位置的可能性最大。fDSST 算法是对 DSST 算法的改进，本文借鉴该算法对目标进行 HOG 特征的提取。针对目标区域提取  $d$  维特征，记作  $f_l, l \in \{1, 2, \dots, d\}$ ， $l$  表示特征通道，通过建立最小化代价函数，构建最优的相关滤波器  $h_l$ 。目标函数的公式为

$$\varepsilon = \|g - \sum_{l=1}^d h_l \star f_l\|^2 + \lambda \sum_{l=1}^d \|h_l\|^2, \quad (3)$$

式中:  $\star$  表示循环相关;  $g$  表示期望的响应分数, 由高斯函数产生; (3) 式中的第二项表示权重参数为  $\lambda$  的正则化项,  $f_l, h_l$  与  $g$  的维数相同, 大小也相同。

为减少卷积计算耗费的时间, 基于 Parseval 公式将 (3) 式转换到傅里叶域进行计算, 得到最优滤波器为

$$H_l = \frac{\bar{G}F_l}{\sum_{k=1}^d \bar{F}_k F_k + \lambda}, \quad l=1, 2, \dots, d, \quad (4)$$

式中:  $\bar{G}$  表示  $g$  经傅里叶变换后的复共轭;  $\bar{F}_k$  表示  $f_l$  经傅里叶变换后的复共轭。(4) 式中的大写字母对应 (3) 式中的相应变量的离散傅里叶变换 (DFT), 在跟踪过程中, 根据每帧的跟踪结果决定是否进行模型更新。为确保模型能得到稳健的近似结果, 采用如下策略更新相关滤波器模型:

$$A_t^{(l)} = \bar{G}U_t^{(l)}, \quad l=1, 2, \dots, \tilde{d}, \quad (5)$$

$$B_t = (1 - \eta)B_{t-1} + \eta \sum_{k=1}^{\tilde{d}} \bar{F}_t^{(k)} F_t^{(k)}, \quad (6)$$

式中:  $t$  表示目标序列的帧数;  $\bar{F}_t^{(k)}$  表示傅里叶域第  $t$  帧、第  $k$  维特征的复共轭;  $U_t = \mathcal{F}\{\mathbf{P}_t u_t\}$ , 其中  $u_t$  是一个目标模板,  $u_t = (1 - \eta)u_{t-1} + \eta f_t$ , 用于减少快速傅里叶变换 (FFT) 需要的计算量,  $\mathbf{P}_t$  是通过最小化重构目标模板  $u_t$  构造出的投影矩阵, 其大小为  $\tilde{d} \times d$ , 作用是将提取到的特征投影到低维子空间,  $\tilde{d}$  表示特征经过压缩后的特征维度;  $F_t = \mathcal{F}\{\mathbf{P}_t f_t\}$ ;  $\eta$  表示学习率, 作为模型的更新参数。(5) 式和 (6) 式分别表示 (4) 式中分子和分母部分的更新策略。

在第  $t$  帧图像中, 参照目标在  $t-1$  帧的位置, 在指定的背景区域提取不同维度的特征样本, 对样本进行压缩后使用相关滤波器得到  $z_t^{(l)}$ , 对  $z_t^{(l)}$  进行 FFT 得到  $Z_t = \mathcal{F}\{\mathbf{P}_{t-1} z_t\}$ , 最终得到的响应分数为

$$R_t = \frac{\sum_{l=1}^{\tilde{d}} \bar{A}_{t-1}^{(l)} Z_t^{(l)}}{B_{t-1} + \lambda}, \quad (7)$$

式中:  $\bar{A}_{t-1}^{(l)}$  表示第  $t-1$  帧、第  $l$  维特征的复共轭。对得到的响应分数  $R_t$  进行傅里叶逆变换 (IFFT) 得到相关滤波器的响应值  $r_t$ , 响应值越大, 则越有可能是目标的位置。  $r_t$  的计算式为

$$r_t = \mathcal{F}^{-1}\{R_t\}, \quad (8)$$

式中:  $\mathcal{F}^{-1}$  表示傅里叶逆变换。

### 3.1.2 颜色概率模型

颜色概率模型主要用于将目标的前景与背景分开, 通过统计前景目标和背景区域的颜色直方图并进行归一化, 判别每个像素属于前景和背景的概率, 得到像素级的前景概率响应图, 前景响应分数越高的位置, 越有可能是目标的位置。响应图的得分计算式为

$$s(\Phi; \boldsymbol{\alpha}) = \frac{1}{|M|} \sum_{x \in M} \boldsymbol{\alpha}^T \Phi(x), \quad (9)$$

式中:  $M$  为待计算区域提取出的像素个数;  $\boldsymbol{\alpha}$  为模型参数;  $\Phi(x)$  为第  $x$  帧中像素网格内统计颜色特征的函数。通过提取并统计前景和背景网格中的颜色特征, 使用 (9) 式计算出像素的响应分数, 从而得到目标的颜色概率响应图。每帧确定跟踪目标位置后, 采用最小化损失函数的方式更新模型。该模型的损失函数为

$$L(x, \boldsymbol{\alpha}) = \frac{1}{|O|} \sum_{x \in O} [\boldsymbol{\alpha}^T \Phi(x) - 1]^2 + \frac{1}{|B|} \sum_{x \in B} [\boldsymbol{\alpha}^T \Phi(x)]^2, \quad (10)$$

式中:  $O$  表示模型所提取的目标区域的图像个数;  $B$  表示模型所提取的背景区域的图像个数。(10) 式由两项组成, 前一项表示目标区域的损失, 后一项表示背景区域的损失。在模型中使用的颜色特征为 RGB, 统计直方图得分可以使用平均投票的方式。在损失函数中, 使用每个图像的目标函数对每个特征像素进行线性回归。对于模型参数  $\boldsymbol{\alpha}$ , 采用以下策略解决相关岭回归问题:

$$\boldsymbol{\alpha}_t = \arg \min_{\boldsymbol{\alpha}} \left\{ L(x, \boldsymbol{\alpha}) + \frac{1}{2} \lambda \|\boldsymbol{\alpha}\|^2 \right\}, \quad (11)$$

$$\alpha_t^{(j)} = \frac{\rho^{(j)}(O)}{\rho^{(j)}(O) + \rho^{(j)}(B) + \lambda}, \quad j=1, 2, \dots, m, \quad (12)$$

式中:  $\rho^{(j)}(O)$  为目标区域中特征  $j$  的非零区域中像素比例;  $\rho^{(j)}(B)$  为背景区域中特征  $j$  的非零区域中像素比例;  $j$  为特征的维度;  $\lambda$  为模型的权重参数。

跟踪第  $t$  帧时, 当学习率为  $\eta$ , 采用以下策略在线更新像素比例:

$$\rho_t(O) = (1 - \eta)\rho_{t-1}(O) + \eta\rho'_t(O), \quad (13)$$

$$\rho_t(B) = (1 - \eta)\rho_{t-1}(B) + \eta\rho'_t(B), \quad (14)$$

式中:  $\rho_t(O)$  为颜色概率模型中第  $t$  帧目标区域的更新策略;  $\rho'_t(O)$  为第  $t$  帧计算目标区域的直方图得分;  $\rho_t(B)$  为颜色概率模型中第  $t$  帧背景区域的更新策略;  $\rho'_t(B)$  为第  $t$  帧计算背景区域的直方图得分。

### 3.2 基于 GMS 的目标检测器

在可信度策略的基础上,针对最高可置信度仍然较低的情况,提出了一种基于 GMS 特征点匹配<sup>[16]</sup>的目标检测器。检测器首先通过特征点匹配判断是否进行重检测,在重检测时将利用当前帧计算得出的目标位置与前一帧目标的预测位置进行比较,选择可信度最高的位置作为当前帧目标

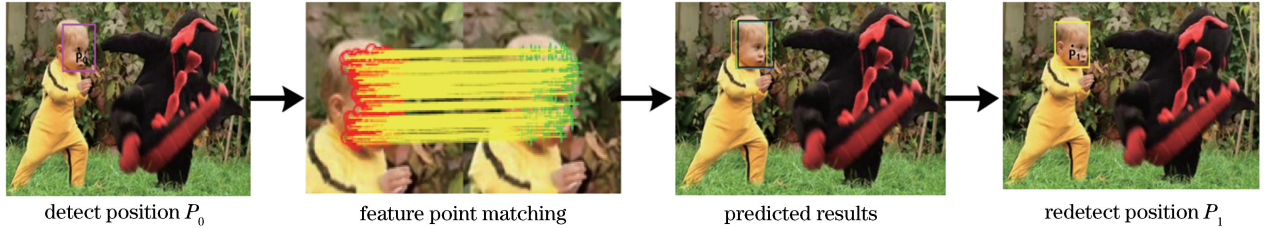


图 3 重检测过程示意图

Fig. 3 Schematic of redetection process

GMS 特征点匹配是文献<sup>[16]</sup>提出的一种基于网格运动统计的快速、超稳健的特征匹配算法,其主要思想是:首先进行 ORB (Oriented FAST and Rotated BRIEF) 特征匹配,得到特征匹配点,然后使用 GMS 方法对匹配结果进行过滤,消除错误的特征匹配点对,保留匹配成功的特征点。为了减少邻域之间像素的干扰,增加特征匹配结果的差异,构建了一种基本平滑运动核,其大小为  $3 \times 3$ ,每个网格的匹配统计值  $S_i$  满足二项分布,每个核的匹配统计值计算式为

$$S_{a,b} = \sum_{u=1}^K |X_{a^{(u)}b^{(u)}}|, \quad (15)$$

式中: $K=9$  为平滑运动核的网格个数; $|X_{a^{(k)}b^{(k)}}|$  为区域  $a$  与区域  $b$  中匹配成功的特征点对数。该算法设定判定阈值  $\tau = m_f + \delta s_f \approx \delta s_f \approx \delta \sqrt{n}$  ( $\delta=6$ ), $\delta$  表示设定阈值的平衡常数。其中  $m_f$  为错误匹配的对数均值, $s_f$  为方差,每个核的匹配结果为

$$P_{\text{cell-pair}}\{a,b\} = \begin{cases} \text{T} & S_{a,b} > \tau_a = \delta \sqrt{n_a} \\ \text{F} & \text{otherwise} \end{cases}, \quad (16)$$

式中:T 表示成功匹配;F 表示错误匹配。

通过对当前帧可信度较高的目标区域与前一帧的目标区域进行特征点匹配,得到匹配成功的特征点对数,匹配成功点越多,表示两帧之间的目标区域越相似,匹配程度越高;反之,则匹配程度越低。实验中设定特征点总数为  $S_{\text{point}}$ ,匹配阈值为  $Y_{\text{threshold}}$ ,匹配阈值为特征点总数的  $1/2$ 。当匹配成功的特征点数大于阈值时,表明两帧之间的相似程度高于  $50\%$ ,虽然可信度较低,但是模型定位的区域仍是目

的最终位置,实现对目标的跟踪。以序列 DragonBaby 中的第 5 帧为例,目标重检测过程示意图如图 3 所示,图中的检测候选位置为  $P_0(107.8,219.1)$ ,特征点对数为 178,5 个预测区域的可信度集合为  $\{3.27,2.89,3.14,4.09,3.18\}$ ,可信度最高的位置为  $P_1(108.3,205.2)$ , $P_1$  为重检测确定的目标的最终跟踪位置。

标区域,对目标进行尺度估计以及整体模型的更新。当匹配成功的特征点数小于阈值时,说明两帧之间的相似程度低于  $50\%$ ,此时激活检测器,重新检测目标区域。

图 4 为以序列 Bird2 为例模拟得到的目标运动轨迹。结合图 4 中目标的运动轨迹可以看出,目标运动是平滑渐进的过程,所以两帧之间目标位置移动的距离不会呈阶跃性的变化。因此,根据前一帧的目标中心点位置,在其不同方向上设置预测区域作为重检测时的目标预测区域。同时考虑可信度较低以及匹配程度较低的原因可能在于:1)漂移的发生,导致目标丢失;2)对目标的定位不够准确。因此将模型在当前帧定位的可信度最大但低于可信度阈值的目標位置也作为预测位置,以实现目标的准确定位。本文共设置 5 个预测区域,预测区域集合为  $N = \{N_1, N_2, N_3, N_4, N_5\}$ ,其中  $N_5$  表示当前帧候选目标位置确定的预测区域, $N_1, N_2, N_3, N_4$  表示根据前一帧的目标位置设置 4 个预测区域,分别对应 4 个方向,预测区域的构建过程如图 5 所示。

图 5 中, $O_{\text{center}}$  为前一帧的目标中心点,预测区域的中心点分别为  $A_0, B_0, C_0$  和  $D_0$ ,前一帧的目标尺寸记作  $w_{\text{prior}} \times h_{\text{prior}}$ 。以  $A_0$  为例, $A_0$  是通过  $O_{\text{center}}$  先向右平移  $\varphi \times w_{\text{prior}}$  的距离,然后再向上平移  $\varphi \times h_{\text{prior}}$  的距离得到。 $B_0, C_0$  与  $D_0$  的获得过程与  $A_0$  类似,只是平移方向不同。根据实验效果, $\varphi = 1/4$  时效果最好。图 5 中所示的红色框是以  $O_{\text{center}}$  为中心的搜索区域,黄色框是以  $A_0$  为中心的搜索区域,虚线框表示  $A_0, B_0, C_0, D_0$  4 个点共同确定的搜索区域。

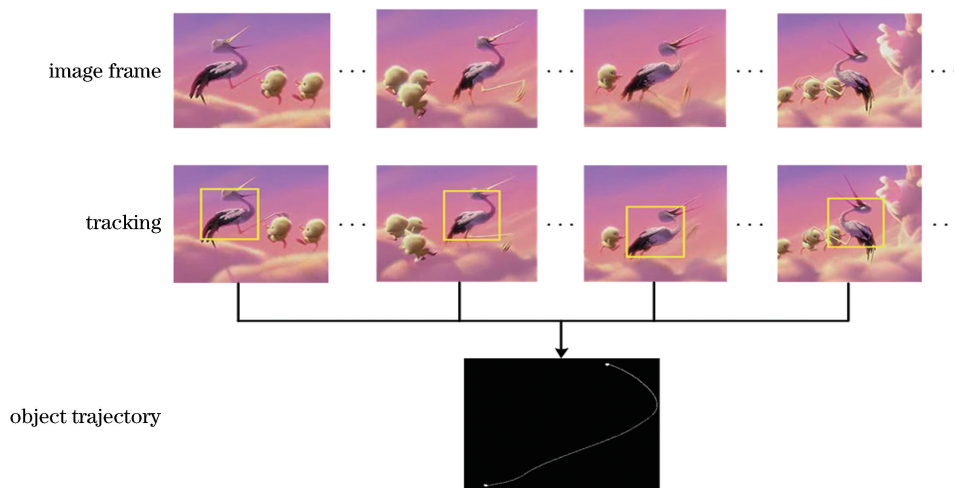


图 4 目标运动轨迹图

Fig. 4 Trajectory of object motion

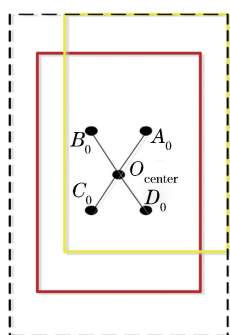


图 5 预测区域构造示意图

Fig. 5 Schematic of predicted area

使用当前帧的融合特征组合方式导致可信度较低,因此在对预测区域进行检测时,采用的融合特征融合方式中不包括当前帧已经使用过的融合方式,增强了检测器对偶然因素形成的可信度较高的特征的抗干扰能力。对于每个预测区域,使用位置滤波

器与颜色概率模型进行特征提取,通过分配不同的权重得到融合特征,选取最大的可信度作为该预测区域的可信度。预测区域可信度集合表示为  $Q = \{Q_1, Q_2, Q_3, Q_4, Q_5\}$ ,取可信度集合中数值最大的预测区域为目标所在的区域,同时使用该预测区域中选择的特征对目标的位置进行定位,找到目标的中心位置,实现跟踪。 $Q$  集合中最大可信度低于阈值  $\theta_{APCE}$  时,表明该区域的特征不足以表征目标,因此对当前帧不进行模型更新,以确保整体模型对目标描述的准确性。

对序列 DragonBaby 的前 10 帧的位置数据进行采集,结果如表 2 所示。通过对比表 2 中检测位置、重检测位置以及目标实际位置可以看出,进行重新检测后,得到的目标位置与实际位置接近,说明了本文的重检测机制的必要性以及目标检测器的有效性。

表 2 目标中心位置比较

Table 2 Comparison of central positions of targets

Frame	Detection position	Feature point	Redetection position	Actual position
1	(115.5,188.0)	—	—	(115.5,188.0)
2	(115.0,195.2)	—	—	(116.5,193.5)
3	(112.0,202.5)	—	—	(110.5,201.0)
4	(107.9,203.7)	—	—	(107.5,202.5)
5	(107.8,219.1)	178	(108.3,205.2)	(109.5,207.0)
6	(102.5,212.4)	198	(101.3,215.7)	(99.5,217.5)
7	(93.5,212.7)	226	(91.1,213.8)	(89.5,215.5)
8	(79.1,202.8)	120	(68.2,194.0)	(66.0,194.5)
9	(66.0,188.8)	—	—	(67.5,188.5)
10	(79.1,182.3)	—	—	(76.5,184.5)

### 3.3 多尺度滤波

针对目标跟踪过程中的尺度变化问题,在确定目标位置后,使用尺度相关滤波器对目标的尺度变换进行估计,构建目标尺度金字塔。尺度滤波器模型的期望响应函数是高斯函数,使用相关滤波器对尺度池中不同的尺度进行评估,响应分数最大的尺度即为目标当前帧的最佳尺度,构建尺度池的方法为

$$S_{\text{scale}} = \sigma^n \times W \times \sigma^n \times H, \\ n \in \left\{ \left[ -\frac{S-1}{2} \right], \dots, \left[ \frac{S-1}{2} \right] \right\}, \quad (17)$$

式中: $W \times H$  为前一帧目标的大小; $\sigma$  为尺度因子;

$n$  为比例滤波器的大小; $S$  为压缩后的特征维数; $S_{\text{scale}}$  为尺度响应值。

尺度滤波器与位置滤波器的滤波过程相似,采用一维高斯函数作为期望函数,将尺度滤波器  $h_{\text{scale}}$  与尺度训练样本  $f_{\text{scale}}$  代入(3)式,然后在傅里叶域进行计算,得到 17 维的尺度响应向量,通过对 17 维的尺度向量进行内插操作,得到 33 维的响应向量,响应值最大的尺度即为目标的最佳尺度。为提高计算和存储效率,并未像位置滤波器中通过构建自相关矩阵的方式获得尺度投影矩阵  $\mathbf{P}_{\text{scale}}$ ,而是通过对  $f_{\text{scale}}$  进行正交三角(QR Decomposition)分解得到,这样处理不会影响跟踪结果的输出。表 3 中的数据

表 3 估计尺度与实际尺度信息表

Table 3 Estimated scale and actual scale information

Estimated scale	Frame						
	22	32	42	52	62	72	82
1	(44,27)	(45,28)	(46,29)	(47,29)	(48,30)	(51,32)	(53,33)
2	(45,28)	(46,29)	(47,29)	(48,30)	(49,30)	(52,32)	(54,34)
3	(45,28)	(47,29)	(48,30)	(49,30)	(50,31)	(53,33)	(55,34)
4	(46,29)	(48,30)	(49,30)	(50,31)	(51,32)	(54,34)	(57,35)
5	(47,29)	(49,30)	(50,31)	(51,32)	(52,32)	(55,34)	(58,36)
6	(48,30)	(50,31)	(51,32)	(52,32)	(53,33)	(57,35)	(59,36)
7	(49,30)	(51,32)	(52,32)	(53,33)	(54,34)	(58,36)	(60,37)
8	(50,31)	(52,32)	(53,33)	(54,34)	(55,34)	(59,36)	(61,38)
9	(51,32)	(53,33)	(54,34)	(55,34)	(57,35)	(60,37)	(62,39)
10	(52,32)	(54,34)	(55,34)	(57,35)	(58,36)	(61,38)	(64,39)
11	(53,33)	(55,34)	(57,35)	(58,36)	(59,36)	(62,39)	(65,40)
12	(54,34)	(57,35)	(58,36)	(59,36)	(60,37)	(64,39)	(66,41)
13	(55,34)	(58,36)	(59,36)	(60,37)	(61,38)	(65,40)	(68,42)
14	(57,35)	(59,36)	(60,37)	(61,38)	(62,39)	(66,41)	(69,43)
15	(58,36)	(60,37)	(61,38)	(62,39)	(64,39)	(68,42)	(70,44)
16	(59,36)	(61,38)	(62,39)	(64,39)	(65,40)	(69,43)	(72,44)
17	(60,37)	(62,39)	(64,39)	(65,40)	(66,41)	(70,44)	(73,45)
18	(32,20)	(33,21)	(34,21)	(34,21)	(35,22)	(37,23)	(39,24)
19	(32,20)	(34,21)	(34,21)	(35,22)	(36,22)	(38,24)	(40,25)
20	(33,21)	(34,21)	(35,22)	(36,22)	(37,23)	(39,24)	(40,25)
21	(34,21)	(35,22)	(36,22)	(37,23)	(37,23)	(40,25)	(41,25)
22	(34,21)	(36,22)	(37,23)	(37,23)	(38,24)	(40,25)	(42,26)
23	(35,22)	(37,23)	(37,23)	(38,24)	(39,24)	(41,25)	(43,27)
24	(36,22)	(37,23)	(38,24)	(39,24)	(40,25)	(42,26)	(44,27)
25	(37,23)	(38,24)	(39,24)	(40,25)	(40,25)	(43,27)	(45,28)
26	(37,23)	(39,24)	(40,25)	(40,25)	(41,25)	(44,27)	(45,28)
27	(38,24)	(40,25)	(40,25)	(41,25)	(42,26)	(45,28)	(46,29)
28	(39,24)	(40,25)	(41,25)	(42,26)	(43,27)	(45,28)	(47,29)
29	(40,25)	(41,25)	(42,26)	(43,27)	(44,27)	(46,29)	(48,30)
30	(40,25)	(42,26)	(43,27)	(44,27)	(45,28)	(47,29)	(49,30)
31	(41,25)	(43,27)	(44,27)	(45,28)	(45,28)	(48,30)	(50,31)
32	(42,26)	(44,27)	(45,28)	(45,28)	(46,29)	(49,30)	(51,32)
33	(43,27)	(45,28)	(45,28)	(46,29)	(47,29)	(50,31)	(52,32)
Optimum scale	(41,25)	(45,28)	(46,29)	(47,29)	(49,30)	(52,32)	(53,33)
Actual scale	(43,24)	(48,25)	(46,27)	(47,27)	(53,28)	(55,26)	(54,27)



表示序列 CarScale 的部分帧的尺度估计信息,并将尺度滤波器选取的最佳尺度与实际尺度进行了比较,从表中数据可以看出,最佳估计尺度与目标的实际尺度差距很小,表明本文的尺度估计方法能够适应目标的尺度变化。

## 4 算法步骤

自适应特征选择的相关滤波跟踪算法步骤如下:

1) 建立模型。使用已知的首帧目标位置信息,选取首帧中的目标区域和搜索区域,分别训练 2 维位置滤波器、颜色概率模型以及 1 维尺度滤波器。

2) 融合特征。取  $t-1$  帧的目标位置作为中心位置,利用该位置构建相应的背景区域和前景区域。使用位置滤波器在背景区域中采集候选样本  $z_t$ ,将  $z_t$  进行 FFT 变换后代入(7)式和(8)式,得到滤波器的响应图  $f_{cf}$ ;同时使用颜色概率模型统计前景和背景的颜色直方图并进行归一化,将结果代入(9)式得到像素级的前景概率响应图  $f_{color}$ 。对得到的  $f_{cf}$  与  $f_{color}$  分配不同的权重,代入(1)式得到多个最终的特征响应图  $f_{fuse}$ 。

3) 可信度判断。通过(2)式对步骤 2)中分配不同的权重  $\omega$  的多个  $f_{fuse}$  进行可信度判断,取最大可信度的融合特征作为当前帧的跟踪特征,并估计出目标的候选位置。若最大可信度高于阈值  $\theta_{APCE}$ ,则候选位置即为当前帧的目标位置,转至步骤 5);若最大可信度不超过阈值  $\theta_{APCE}$ ,则可能需要对目标的位置进行重新检测,转至步骤 4)。

4) 特征点匹配和重检测。对于步骤 3)中最大可信度仍较低的情况,根据(15)式和(16)式,使用特征点匹配的方法比较该可信度下  $t$  帧中确定的目标区域与  $t-1$  帧中确定的目标区域的相似程度,若特征点匹配成功的点对数大于匹配阈值  $Y_{threshold}$ ,则转至步骤 5);否则对预测区域进行重新检测,选取可信度最大的预测区域作为目标区域。

5) 尺度估计。以当前帧确定的目标位置为中心位置,使用(17)式提取不同的候选样本以构建尺度金字塔,根据(7)式和(8)式计算尺度滤波器的响应值,最大响应尺度即为第  $t$  帧目标的最佳尺度。

6) 模型更新。对于步骤 4)中重新检测过程得到的预测区域中最大可信度仍不超过阈值  $\theta_{APCE}$  的情况,不进行模型更新,使用(5)式和(6)式更新滤波器,同时根据(13)式与(14)式更新颜色概率模型。

7) 输出跟踪结果。将跟踪窗口进行可视化输

出,继续执行步骤 2)。

## 5 实验结果及分析

### 5.1 实验参数设置及数据

所有实验均在 CPU 为 Intel Core i7-6700, 3.4 GHz 主频和 16 GB 内存的计算机上完成,测试开发平台为:MATLAB R2016a。在位置滤波器模型中,初始提取的特征维度  $d=32$ ,压缩后的特征维度  $\tilde{d}=17$ ,正则化权重参数  $\lambda=0.001$ ,学习率  $\eta=0.03$ 。在颜色概率模型中,正则化权重参数  $\lambda=0.001$ ,学习率  $\eta=0.04$ 。在尺度滤波模型中,尺度因子  $\sigma=1.02$ ,压缩后的特征维数  $S=17$ 。学习率  $\eta=0.025$ ,在特征点匹配过程中,特征点总数  $S_{point}=1000$ ,匹配阈值  $Y_{threshold}=500$ 。

为了便于与其他主流算法进行对比,实验选取的标准测试数据集为 OTB50<sup>[17]</sup> 和 OTB100<sup>[18]</sup>。OTB50 数据集包含 50 组视频序列,涵盖 11 种不同视觉跟踪挑战的属性,如光照变化(IV)、尺度变换(SV)、运动模糊(MB)等,每段序列可能有多个属性。OTB100 是在 OTB50 的基础上新增了 50 组测试视频序列,背景属性更加复杂,跟踪挑战性更大。

实验采用一次性通过评估(OPE)评价准则来计算跟踪准确率和成功率,将准确率、成功率以及跟踪速度作为跟踪性能的评价标准。跟踪准确率(DP)为跟踪中心点位置误差小于阈值的帧数占整个序列帧数的比例,跟踪成功率(OP)则表示跟踪重叠率(交并比)大于阈值的帧数占总体序列帧数的比例,跟踪速度(FPS)为跟踪器每秒读取视频序列中图片的个数,其中准确率阈值设定为 20 pixel,成功率阈值设定为 0.5。

### 5.2 对比实验分析

为了验证本文跟踪算法的有效性,选取 7 种近几年出现的跟踪效果较好的跟踪算法进行对比,分别为 fDSST<sup>[13]</sup>, SAMF<sup>[19]</sup>, SRDCF<sup>[20]</sup>, BACF<sup>[21]</sup>, STRCF<sup>[22]</sup>, STAPLE<sup>[10]</sup>, ECO<sup>[23]</sup>。这些算法涵盖基于相关滤波的算法、相关滤波改进的算法以及深度学习的算法,本节将从跟踪的准确性与时效性两个方面进行跟踪效果的对比。

#### 5.2.1 准确性分析

在 OTB50 与 OTB100 数据集的不同属性下测试了 8 种算法,得到 8 种算法的距离精确度和跟踪重叠率见表 4。8 种算法分别在 OTB50、OTB100 上的跟踪成功率曲线和跟踪精确率曲线如图 6 和图 7 所示,横坐标分别表示重叠率阈值和中心误差阈

表 4 8 种跟踪算法平均跟踪性能比较

Table 4 Average tracking performance comparison among eight tracking algorithms

Algorithm		fDSST	SAMF	SRDCF	BACF	STAPLE	STRCF	ECO	Ours
Mean DP	OTB50	0.696	0.653	0.736	0.735	0.693	0.831	0.868	0.789
	OTB100	0.742	0.741	0.778	0.801	0.795	0.855	0.886	0.819
Mean OP	OTB50	0.623	0.558	0.652	0.673	0.605	0.684	0.698	0.675
	OTB100	0.583	0.554	0.612	0.643	0.604	0.680	0.697	0.654

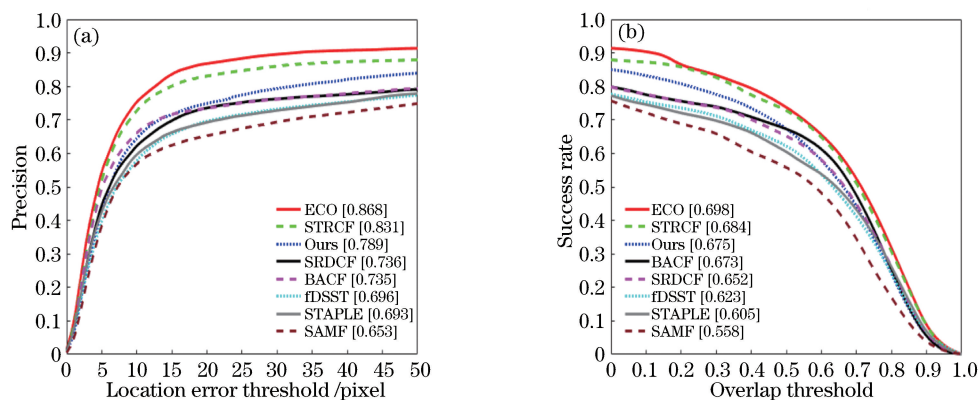


图 6 8 种跟踪算法在 OTB50 上的精确率和成功率。(a)精确率对比曲线;(b)成功率对比曲线

Fig. 6 Precisions and success rates for 8 tracking algorithms on OTB50. (a) Precision comparison; (b) success rate comparison

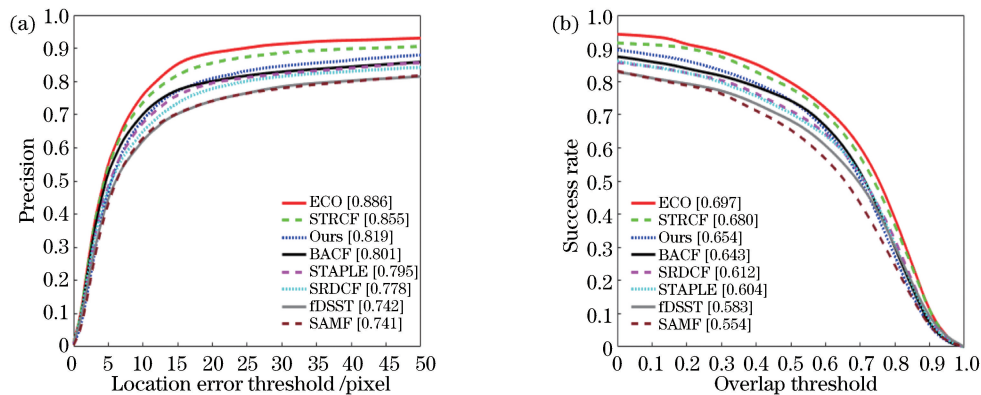


图 7 8 种跟踪算法在 OTB100 上的精确率和成功率。(a)精确率对比曲线;(b)成功率对比曲线

Fig. 7 Precisions and success rates of 8 tracking algorithms on OTB100. (a) Precision comparison; (b) success rate comparison

值,纵坐标表示满足阈值条件的帧数占总帧数的比例。

结合表 4 中的数据和图 6、图 7 中的曲线可以看出,在 OTB50 数据集上,本文提出的跟踪算法在跟踪成功率上接近 STRCF 算法,低于基于深度学习的 ECO 算法,高于其他的算法,其在跟踪精度上的表现同样如此。在 OTB100 数据集上,本文算法的跟踪成功率低于 ECO 算法与 STRCF 算法,高于其他的算法,跟踪精度也仅次于 ECO 算法和

STRCF 算法而高于其他算法。综上所述,本文算法虽然比基于深度学习算法的跟踪准确度低,但是在大多数情况下表现较好,能准确、稳健地跟踪目标。不过与针对遮挡问题的 STRCF 算法相比,本文算法对于遮挡情况的处理能力欠佳。因此在以后的工作中,将重点对如何提高模型对遮挡的处理能力做进一步研究,以提高跟踪方法的整体准确性和稳健性。同时,虽然本文算法的滤波跟踪器是基于 fDSST 算法,但是在跟踪的平均准确

率和成功率上有明显提高。结合表 4 数据可知，在 OTB50 数据集上，相比 fDSST 算法，本文算法的平均准确率与平均成功率均提升了约 7%。为

了更加直观地表示不同算法的跟踪效果，将 8 种跟踪算法在部分序列上的跟踪效果进行对比，结果如图 8 所示。



图 8 8 种跟踪方法在部分序列上的跟踪结果

Fig. 8 Tracking results of 8 tracking algorithms in partial sequences

BlurOwl 序列中目标具有运动模糊以及快速运动的属性,在第 10 帧到第 388 帧的跟踪过程中,目标在平面内快速运动,部分跟踪算法逐渐漂移,尤其在第 156 帧时,只有 ECO 算法、STRCF 算法以及本文算法能精确地跟踪到目标位置,经历运动模糊过程后(第 470 帧、第 631 帧),仅有 STRCF 算法与本文算法在尺度和位置上可精确地定位目标,其中 STRCF 算法在第 470 帧时发生了轻度漂移,表明本文算法对于快速运动和运动模糊的情形具有稳健性。

在 Couple 序列对目标跟踪的过程中(第 10 帧、第 140 帧),目标所处的背景比较杂乱且目标的姿态不断变化,导致跟踪器跟丢目标或无法准确跟踪目标。在第 47 帧和第 108 帧中,STAPLE、SAMF 以及 fDSST 算法因背景干扰未能正确跟踪目标,ECO 算法因为目标尺度的不断变化,不能精确地定位目标的尺度。在第 140 帧中,STRCF 算法被相似目标吸引导致错误地跟踪了目标,只有 BACF 算法与本文算法成功地跟踪了目标,表明本文算法在背景杂乱与尺度变化时具有稳定性,有效地实现了对目标的跟踪。

Skiing 序列中显示的是目标在光照变化和变形的情况下,8 种跟踪算法的跟踪效果。从第 2 帧到第 81 帧的跟踪过程中,除了 ECO 算法与本文算法外,其余算法均在第 16 帧时发生了漂移且无法再次跟踪到目标。在第 33 帧时,由于目标发生了变形,ECO 算法未能精确定位目标的位置,仅本文算法准确地跟踪到目标,表明本文选择特征的方法对光照变化以及变形具有较强的适应能力,跟踪较为稳定。

Bird2 序列中的目标存在遮挡与平面外旋转的情况,在第 49 帧时,只有本文算法准确地跟踪到目标,其他跟踪算法由于部分遮挡发生了不同程度的漂移,除 ECO 算法外,其他方法由于在漂移过程中学习了大量的背景信息,导致跟丢目标(第 61 帧、第 83 帧)。在第 98 帧时,STRCF、SAMF 以及 fDSST

跟踪算法丢失目标后无法重新检测跟踪到目标,本文算法通过在跟踪过程中采用特征点匹配机制,一定程度上克服了部分遮挡的影响,准确地跟踪到了目标。

Bolt2 序列与 Football1 序列中目标背景比较复杂,同时目标在平面内快速运动并存在大量相似目标的干扰。其中 Bolt2 序列中,由于在目标的快速运动过程中,SAMF 算法与 fDSST 算法中的跟踪器对于错误的跟踪结果没有可靠性判定,导致目标逐渐跟踪错误(第 25 帧、第 88 帧)。STRCF 算法由于相似目标的干扰,在第 182 帧时错误地跟踪了目标。在对 Football1 序列的跟踪过程中,目标还存在平面内旋转的属性。跟踪过程中目标经过快速运动和旋转后,在第 74 帧时,只有本文跟踪算法能长期、稳定地跟踪到目标。本文通过在跟踪过程中引入可信度策略,对每帧的跟踪结果进行可靠性判定,增强了跟踪器对于错误跟踪结果的判别能力,从而提高了跟踪的准确性。

#### 5.2.2 时效性分析

8 种算法在 OTB50 和 OTB100 数据集上的平均运行速度见表 5。结合表 5 数据可以看出,本文提出的算法在运行速度上仅次于 fDSST 算法,优于其他算法。综合表 4 和表 5 的数据可知,虽然基于深度学习的 ECO 算法与针对遮挡问题的 STRCF 算法在跟踪效果上略优于本文算法,但是在跟踪速度上,上述两种算法的运行速度较慢,不能满足实时性的要求。相较于 ECO 算法与 STRCF 算法,本文算法的优势在于能保证跟踪效果相对准确、稳定的同时,还能基本满足实时性的要求;相较于固定等权重融合特征的 STAPLE 算法,本文算法除了采用多权重分配方式融合特征之外,在基础特征提取以及尺度估计过程中,采用比 STAPLE 算法计算量更小的加速算法,因此在跟踪准确率以及跟踪速度上均优于 STAPLE 算法;相较于运行速度比本文算法较高的 fDSST 算法,本文算法的跟踪准确率更高,跟踪性能更好。

表 5 8 种跟踪算法平均跟踪速度比较

Table 5 Average tracking speed comparison among 8 tracking algorithms

Algorithm	fDSST	SAMF	SRDCF	BACF	STRCF	STAPLE	ECO	Proposed	
Mean FPS / (frame · s <sup>-1</sup> )	OTB50	74.78	19.81	6.46	27.92	18.76	55.51	1.43	71.43
	OTB100	88.44	22.80	6.46	31.92	21.90	68.83	1.38	72.61

## 6 结 论

本文算法使用不同的权重分配方式融合两种基础特征,从多样化的融合特征中选择可置信度最大的特征,将其作为目标的跟踪特征对目标进行跟踪,同时提出了一个基于GMS特征点匹配的检测器,用于重新检测。通过将多样的特征融合方式与高置信度选择策略相结合,增强了模型对于目标以及背景变化的适应能力;当选择的特征可置信度较低时,根据特征点匹配的结果对目标进行重新检测,实现对目标稳定、准确的跟踪。上述实验结果表明,本文跟踪算法的跟踪速度基本满足实时性的要求,有效地提高了目标跟踪的准确性和稳健性,尤其对于快速运动、运动模糊、光照变化等场景具有良好的性能。本文算法没有更多地考虑大面积遮挡对模型的影响,因此下一步的工作重点是将遮挡考虑在内,以进一步提高目标跟踪效果。

## 参 考 文 献

- [1] Zhang T Z, Xu C S, Yang M H. Multi-task correlation particle filter for robust object tracking [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 4819-4827.
- [2] Zhao G P, Shen Y P, Wang J Y. Adaptive feature fusion object tracking based on circulant structure with kernel [J]. Acta Optica Sinica, 2017, 37(8): 0815001.  
赵高鹏, 沈玉鹏, 王建宇. 基于核循环结构的自适应特征融合目标跟踪 [J]. 光学学报, 2017, 37(8): 0815001.
- [3] Zhang Z, Sun J, Yang L T. Tracking algorithm based on correlation filter fusing and keypoint matching [J]. Acta Optica Sinica, 2019, 39(2): 0215001.  
张哲, 孙瑾, 杨刘涛. 融合相关滤波与关键点匹配的跟踪算法 [J]. 光学学报, 2019, 39(2): 0215001.
- [4] Ge B Y, Zuo X Z, Hu Y J. Long-term object tracking based on feature fusion [J]. Acta Optica Sinica, 2018, 38(11): 1115002.  
葛宝义, 左宪章, 胡永江. 基于特征融合的长时目标跟踪算法 [J]. 光学学报, 2018, 38(11): 1115002.
- [5] Lu H C, Li P X, Wang D. Visual object tracking: a survey [J]. Pattern Recognition and Artificial Intelligence, 2018, 31(1): 61-76.  
卢湖川, 李佩霞, 王栋. 目标跟踪算法综述 [J]. 模式识别与人工智能, 2018, 31(1): 61-76.
- [6] Bolme D S, Beveridge J R, Draper B A, *et al.* Visual object tracking using adaptive correlation filters [C]//2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 13-18, 2010, San Francisco, CA, USA. New York: IEEE, 2010: 2544-2550.
- [7] Henriques J F, Caseiro R, Martins P, *et al.* Exploiting the circulant structure of tracking by detection with kernels [M]//Fitzgibbon A, Lazebnik S, Perona P, *et al.* Computer Vision-ECCV 2012. Berlin, Heidelberg: Springer, 2012, 7575: 702-715.
- [8] Henriques J F, Caseiro R, Martins P, *et al.* High-speed tracking with kernelized correlation filters [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.
- [9] Danelljan M, Häger G, Khan F, *et al.* Accurate scale estimation for robust visual tracking [C]//British Machine Vision Conference 2014, September 1-5, 2014, Nottingham. Durham, England, UK: BMVA Press, 2014: 65.
- [10] Bertinetto L, Valmadre J, Golodetz S, *et al.* Staple: complementary learners for real-time tracking [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 1401-1409.
- [11] Tang M, Feng J Y. Multi-kernel correlation filter for visual tracking [C]//2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE, 2015: 3038-3046.
- [12] Zhang K H, Zhang L, Yang M H, *et al.* Fast tracking via spatio-temporal context learning [EB/OL]. (2013-11-08) [2018-12-15]. <https://arxiv.org/pdf/1311.1939.pdf>.
- [13] 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.
- [14] Possegger H, Mauthner T, Bischof H. In defense of color-based model-free tracking [C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 7-12, 2015, Boston, MA, USA. New York: IEEE, 2015: 2113-2120.
- [15] Wang M M, Liu Y, Huang Z Y. Large margin object tracking with circulant feature maps [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 4800-4808.
- [16] Bian J W, Lin W Y, Matsushita Y, *et al.* GMS: grid-based motion statistics for fast, ultra-robust

- feature correspondence [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 2828-2837.
- [17] Wu Y, Lim J, Yang M H. Online object tracking: a benchmark [C]//2013 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2013, Portland, OR, USA. New York: IEEE, 2013: 2411-2418.
- [18] Wu Y, Lim J, Yang M H. Object tracking benchmark [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37 (9): 1834-1848.
- [19] Li Y, Zhu J K. A scale adaptive kernel correlation filter tracker with feature integration [M]//Agapito L, Bronstein M, Rother C. Computer Vision-ECCV 2014 Workshops. Cham: Springer, 2015, 8693: 254-265.
- [20] Danelljan M, Häger G, Khan F S, *et al.* Learning spatially regularized correlation filters for visual tracking [C]//2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE, 2015: 4310-4318.
- [21] Galoogahi H K, Fagg A, Lucey S. Learning background-aware correlation filters for visual tracking [C]//2017 IEEE International Conference on Computer Vision (ICCV), October 22-29, 2017, Venice, Italy. New York: IEEE, 2017: 1144-1152.
- [22] Li F, Tian C, Zuo W M, *et al.* Learning spatial-temporal regularized correlation filters for visual tracking [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 4904-4913.
- [23] Danelljan M, Bhat G, Khan F S, *et al.* ECO: efficient convolution operators for tracking [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 6931-6939.