

# 基于 MATLAB 的相关滤波跟踪算法仿真分析

王鑫, 刘中旺

(江南大学 物联网工程学院, 江苏 无锡 214122)

**摘要:** 基于相关滤波的目标跟踪算法是一种常见的视觉跟踪方法, 它利用目标的特征信息进行跟踪; 在跟踪过程中通过计算目标模板与当前帧图像中候选区域之间的相关性来判断目标的位置; 通过介绍首个将相关滤波理念与目标跟踪技术相结合的 MOSSE 算法, 引入了 3 种基于此算法的改进相关滤波跟踪算法: KCF 算法、DSST 算法以及 BACF 算法; 并基于视频跟踪基准 OTB100 数据集在 MATLAB 平台进行仿真实验, 一次性评估标准下, BACF 算法的平均成功率与平均精确度分别为最高的 64.5% 与 80.4%, 空间鲁棒性评估标准下, BACF 算法的平均成功率与平均精确度分别为最高的 58.2% 与 78.6%, 时间鲁棒性评估标准下, BACF 算法的平均成功率与平均精确度分别为最高的 65.8% 与 85.1%, 因此 BACF 算法的跟踪性能最佳, 而 KCF 算法实现了最高的 154.36 帧率的跟踪速度。

**关键词:** 目标跟踪; 特征信息; 相关滤波; OTB; MATLAB

## Simulation Analysis of Correlation Filtering Tracking Algorithm Based on MATLAB

WANG Xing, LIU Zhongwang

(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

**Abstract:** The target tracking algorithm based on correlation filtering is a common visual tracking method, which uses the feature information of the target to track; In the tracking process, the target position is determined by calculating the correlation between the target template and the candidate area in the current frame image; Through the introduction of the first MOSSE algorithm combining the concept of correlation filtering with target tracking technology, three improved correlation filtering tracking algorithms based on this algorithm are introduced; KCF algorithm, DSST algorithm and BACF algorithm; And conduct simulation experiments on MATLAB platform based on the OTB100 data set; Under the one-time evaluation standard, the average success rate and average accuracy of BACF algorithm are the highest 64.5% and 80.4% respectively; Under the spatial robustness evaluation standard, the average success rate and average accuracy of BACF algorithm are the highest 58.2% and 78.6% respectively; Under the time robustness evaluation standard, the average success rate and average accuracy of BACF algorithm are the highest 65.8% and 85.1% respectively; Therefore, BACF algorithm has the best tracking performance, while KCF algorithm achieves the highest tracking speed of 154.36 frame rate.

**Keywords:** target tracking; characteristic information; correlation filtering; OTB; MATLAB

## 0 引言

计算机视觉作为人工智能技术的重要领域, 堪称人工智能时代的“眼睛”。它使用摄像机采集信息并通过计算机处理来模拟人类视觉, 以实现视觉信息的提取、分析和处理, 旨在让计算机能“看见”。它在多个领域都有涉及, 如机器学习、图像信号处理、数理统计、认知科学以及控制工程等<sup>[1]</sup>。

目标跟踪技术是计算机视觉的主要组成部分与研究热点, 已得到迅速的发展并在无人驾驶、军事侦察、视频监控和人机交互等领域广泛应用。在无人驾驶领域, 利用目标跟踪技术实现动态目标的跟踪; 在军事侦察领域, 利用无人机实施对特定目标的定位、跟踪和打击等任务; 在视

频监控领域, 基于目标跟踪技术的智能监控系统能有效获取信息, 已被广泛应用于国防、学校等公共场所; 在人机交互领域中, 目标跟踪可实现对手势动作的检测与跟踪<sup>[2]</sup>。

本文重点讨论单目标跟踪, 它是指在一段视频序列中, 通过初始化目标在第一帧图像中的位置信息, 并在后续图像帧中检测、标定目标位置以实现目标跟踪<sup>[3]</sup>。

尽管目标跟踪技术在现实场景中具有很重要的实用价值和广阔的发展前景, 跟踪过程中的目标消失、目标外观变化、背景光线干扰、目标快速移动等诸多问题对目标跟踪技术造成了严峻的挑战。为解决这些问题并建立高效快速的跟踪器, 大量的目标跟踪算法应运而生<sup>[4]</sup>。其中包括模板匹配法、光流法、均值漂移法<sup>[5]</sup>、粒子滤波算

收稿日期: 2023-03-14; 修回日期: 2023-03-30。

基金项目: 国家自然科学基金(61703185); 高等学校学科创新引智计划项目(B12018)。

作者简介: 王鑫(1981-), 男, 山西忻州人, 工学博士, 讲师, 主要从事信号处理算法的设计与加速方向的研究。

引用格式: 王鑫, 刘中旺. 基于 MATLAB 的相关滤波跟踪算法仿真分析[J]. 计算机测量与控制, 2023, 31(8): 224-230, 237.

法<sup>[6]</sup>、跟踪学习检测算法<sup>[7]</sup>、相关滤波算法与深度学习算法。后两种是当下研究最多的两大类算法, 其中深度学习算法主要得益于卷积神经网络不同层的卷积输出所提取的特征能很好地表示目标, 算法准确率较高。但其往往需要大量的计算, 因此算法速度较慢<sup>[8]</sup>。而相关滤波算法则是本文讨论的重点。

## 1 相关滤波类跟踪算法

### 1.1 相关滤波跟踪算法发展历程

相关滤波器常作为检测器, 它们会对视频场景中目标对象产生较大的相关响应峰值, 而背景的相关响应值相对较低, 这为提取目标信息创造了有利条件。最小输出平方误差滤波器 (MOSSE, minimum output sum of squared error)<sup>[9]</sup>是第一个应用相关滤波理念的跟踪算法, 其利用傅里叶变换将空间域的卷积运算转化为频域的点乘运算, 提高了运行速度。但由于训练样本数量不足, 跟踪模型易受到背景和背景外观变化的影响。为此文献 [10] 提出基于核函数的循环结构跟踪 (CSK, circulant structure of tracking with Kernels) 模型。文献 [11] 则在 CSK 算法基础上引入了多通道方向梯度直方图 (HOG, histograms of oriented gradients)<sup>[12]</sup>特征以及高斯核函数提出了核化相关滤波 (KCF, Kernelized correlation filters) 跟踪算法进一步提升了滤波器的训练速度与性能。但上述 3 种算法在跟踪过程中使用的目标跟踪框是固定尺寸的, 当目标尺度变化时算法的跟踪性能会有明显的下降。于是文献 [13] 提出了判别式尺度空间跟踪 (DSST, discriminative scale space tracker) 算法, 将目标定位和尺度预测分离, 通过添加一个独立的尺度相关滤波器来实现目标跟踪框的尺度自适应变化。跟踪算法在处理靠近图像边缘的目标时性能会下降, 即边界效应问题。为此文献 [14] 提出基于空间正则化判别相关滤波器 (SRDCF, spatially regularized discriminative correlation filter) 算法, 而时空正则相关滤波器 (STRCF, learning spatial-temporal regularized correlation filters)<sup>[15]</sup>则通过引入时间正则项实现了时空信息的联合处理, 进一步提升了算法的跟踪性能。与 SRDCF 算法不同, 模板和像素学习总和 (Staple, sum of template and pixel-wise learners) 跟踪算法<sup>[16]</sup>将 HOG 与颜色直方图特征融合来缓解边界效应。文献 [17] 提出了基于上下文感知的跟踪算法, 将上下文信息引入滤波器的训练过程中, 提升了算法性能。视觉跟踪学习背景感知相关滤波器 (BACF, learning background-aware correlation filters for visual tracking) 算法<sup>[18]</sup>采用二值掩膜方案裁剪出目标样本来缓解边界效应。文献 [19] 提出的超越相关滤波器 (CCOT, beyond correlation filters: learning continuous convolution operators for visual tracking) 算法对特征图插值, 针对插值后的每个特征训练对应的相关滤波器, 通过滤波器响应峰值区域预测目标位置, 但由于通道数过多影响了跟踪速度。于是文献 [20] 分别从滤波器选择、样本集、模型更新 3 个角度对其优化, 提出了用于跟踪的高效卷积算子 (ECO, efficient convolution op-

erators for tracking)。文献 [21] 提出用于密集预测的级联融合网络 (CFNet, cascade fusion network for dense prediction) 算法, 在孪生网络中加入相关滤波层来实现在线更新, 并实现了在深度学习模型中不进行分模块或分阶段训练, 直接优化任务总目标的端到端学习。

大部分相关滤波跟踪算法采用逐帧更新的模型并不能适应一些复杂情况, 如光照变化、目标快速运动、运动模糊以及尺度变化等, 这将导致滤波器模型退化<sup>[22]</sup>。对此有相关研究通过对跟踪结果进行置信度评估<sup>[23]</sup>来缓解该问题。近年来深度跟踪算法<sup>[24]</sup>将深度学习和相关滤波算法相结合来进一步提升算法的精确度。虽然深度特征表示具有良好的抗干扰能力和目标表征能力, 但深度跟踪器难以满足目标跟踪实时性要求。

### 1.2 相关滤波跟踪算法原理概述

相关滤波跟踪算法是一种回归判别式方法, 其将目标跟踪视作视频序列中所有帧的目标检测任务, 使用目标图像特征训练滤波器, 并将图像中的目标区域和背景区域分别作为正、负样本。算法在后续帧中使用训练好的滤波器寻找最优解, 并基于每一帧图像中的跟踪结果对滤波器进行更新。

相关滤波跟踪算法主要是利用信号的相关性来实现目标跟踪。跟踪过程中将目标区域和待检测区域视作信号, 计算两者之间的相关性, 相关性越大, 得到的滤波器响应越强<sup>[25]</sup>, 并将相关性最大的区域预测为跟踪区域。因此为实现目标跟踪, 需要设计训练出一个最优的滤波器模板, 将输入图像映射到该模板上, 通过得到的响应峰值区域以确定目标中心区域。

以第一个将相关滤波与目标跟踪相结合的算法 MOSSE 为例, 从数学角度对相关滤波算法的原理进行具体阐述。

### 1.3 MOSSE 算法

$$g = h \otimes f \quad (1)$$

其中:  $g$  表示响应输出,  $h$  表示滤波模板,  $f$  表示输入的图像信息,  $\otimes$  表示卷积运算关系。将  $g$  设计为高斯形状的响应输出, 已知时域内的卷积等于频域上的乘积, 自然想到利用快速傅里叶变换 (FFT, fast Fourier transform) 快速傅里叶变换来简化计算。对于  $n \times n$  大小的图片, 循环卷积运算的时间复杂度为  $O(n^4)$ , 而 FFT 只需要  $O(n^2 \cdot \lg(n))$ , 极大地减少了运算量。 $f$  与  $h$  做相关运算实际是将  $f$  与旋转了 180 度的  $h$  做卷积, 故下面转换到频域的表达式与传统转换公式有差别。

$$G = H^* \cdot F \quad (2)$$

其中:  $G, F$  分别是  $g, f$  经离散傅里叶变换的结果, 而  $H^*$  是频域上的滤波模板取共轭,  $\cdot$  表示点积运算。对上式变形, 得到:

$$H^* = \frac{G}{F} \quad (3)$$

在实际的视觉目标跟踪中, 鉴于目标的外观变换等因素可能会使得到的滤波器模板适用场景有限, 有必要以包

含目标在内的多个图像作为训练样本来提升滤波器的鲁棒性：

$$\min_{H^*} \sum_{i=1}^m |H^* \odot F_i - G_i|^2 \quad (4)$$

其中： $m$  代表输入图像的帧数， $F_i$  代表频域中输入的第  $i$  帧图像信息， $G_i$  代表频域中对应第  $i$  帧输入图像的输出图像信息， $\odot$  表示对应位置点乘。基于最小二乘法利用  $m$  个样本信息求解使输出误差平方和最小的滤波器  $H^*$ 。由于频域中的相关计算为逐元素相乘，因此在求解上式时可对滤波器中的每个元素进行独立优化。通过确保每帧信息对应的误差最小平方和均为最小值以得到使输出误差平方和最小的滤波器  $H^*$ 。对式 (4) 求偏导得到滤波器的闭式最终解：

$$H^* = \frac{\sum_{i=1}^m F_i^* \odot G_i}{\sum_{i=1}^m F_i^* \odot F_i} \quad (5)$$

得到滤波器模板后将其与新来帧图像进行相关操作，并使用快速傅里叶反变换 (IFFT, inverse fast fourier transform) 得到响应图谱，选取响应中最大的位置作为跟踪目标的新位置。之后根据下一帧图像的目标区域来更新滤波器模板，更新公式如下：

$$H_i^* = \frac{A_i}{B_i} \quad (6)$$

$$A_i = \eta G_i \odot F_i^* + (1 - \eta) A_{i-1} \quad (7)$$

$$B_i = \eta G_i \odot F_i^* + (1 - \eta) B_{i-1} \quad (8)$$

其中：系数为学习率，它可以有效限制不同帧的权重，离当前帧越近的帧权重越大，能防止跟踪算法中的过拟合问题，使得滤波器模板更好地适应目标旋转、遮挡等问题。反复执行上述操作来达到持续跟踪目标的目的。MOSSE 算法的整体流程如图 1 所示。

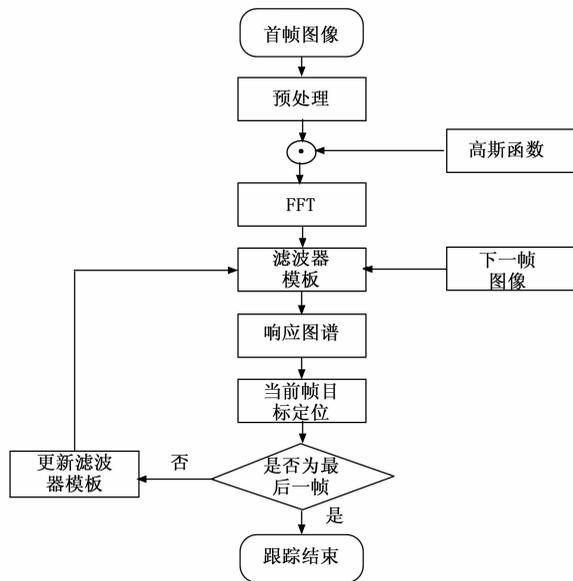


图 1 MOSSE 算法流程图

MOSSE 算法作为相关滤波类跟踪算法的开山之作，与

当时的热门跟踪算法<sup>[5-7]</sup>相比，速度与精度都有了很大的提升。但算法基于最小二乘法求解的线性滤波器模板分类性能一般；算法的样本采样是一种稀疏采样，训练效果一般；算法中提取的图像信息特征是最原始的单通道灰度特征，表达目标的能力有限。针对 MOSSE 算法的主要缺陷，KCF、DSST 以及 BACF 算法均对其进行优化处理，本文后续即对这 3 种算法进行具体阐述并实验分析。

## 2 基于 MOSSE 算法的改良算法

### 2.1 KCF 算法

与 MOSSE 算法不同，KCF 算法采用脊回归来训练分类器，它是一种改良的最小二乘估计法，在原来的损失函数基础上加入了正则化惩罚项以防止滤波器过拟合。设训练样本集为  $(x_i, y_i)$ ，则其线性回归函数为  $f(x_i) = \omega^T x_i$ ， $\omega$  列向量表示模型参数：

$$\min_{\omega} \sum_i (f(x_i) - y_i)^2 + \lambda \|\omega\|^2 \quad (9)$$

$i$  为帧数， $\lambda$  为正则化参数，将上式转换成矩阵形式：

$$\min_{\omega} \|\mathbf{X}\omega - \mathbf{y}\|^2 + \lambda \|\omega\|^2 \quad (10)$$

其中： $\mathbf{X} = [x_1, x_2, \dots, x_n]^T$ ，每一行表示一个向量， $\mathbf{y}$  是列向量，每个元素对应一个样本标签。基于  $\omega$  对上式求导并使其导数为 0，得到频域中  $\omega$ ：

$$\omega = (\mathbf{X}^H \mathbf{X} + \lambda)^{-1} \mathbf{X}^H \mathbf{y} \quad (11)$$

$\mathbf{X}^H$  为频域中  $\mathbf{X}$  的共轭转置矩阵。

在进行分类器的训练时，将跟踪目标所在区域样本称作正样本，将目标区域之外的背景区域样本称为负样本。处理好正负样本之间的平衡关系才能得到性能良好的分类器。MOSSE 算法中利用稀疏采样的方法来获取目标样本，产生了大量冗余样本，影响了跟踪性能。而 KCF 算法采用循环移位方式扩充了负样本的数量，有效缓解了这一问题。假设目标样本  $x = [x_1, x_2, \dots, x_n]^T$ ，置换矩阵为：

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \quad (12)$$

通过将目标样本左乘矩阵即  $\mathbf{P}x = [x_n, x_1, \dots, x_{n-1}]^T$ ，这样样本中元素便向右平移一位，通过左乘  $n$  次置换矩阵，样本向量完成了一次完整的循环位移，组成的循环矩阵：

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \quad (13)$$

循环矩阵可在频域中相似对角化：

$$\mathbf{X} = \mathbf{F} \text{diag}(\hat{x}) \mathbf{F}^H \quad (14)$$

其中： $\mathbf{F}$  为傅里叶变换矩阵， $\hat{x}$  是  $x$  在频域中的表示形式， $\text{diag}(\hat{x})$  为以  $\hat{x}$  为对角线元素的对角矩阵， $\mathbf{F}^H$  为  $\mathbf{F}$  的共轭转置矩阵。

将上式代入式 (11) 中:

$$\omega = \mathbf{F} \text{diag} \left( \frac{\hat{x}^*}{\hat{x}^* \odot \hat{x} + \lambda} \right) \mathbf{F}^H \mathbf{y} \quad (15)$$

\* 表示取对应向量的共轭向量。对上式两边取傅氏变换, 得到:

$$\hat{\omega} = \frac{\hat{x}^* \odot \hat{\mathbf{y}}}{\hat{x}^* \odot \hat{x} + \lambda} \quad (16)$$

通过将矩阵运算转换为向量之间的点积运算, 同时消除了求逆运算, 大大提升了计算速度。现实生活中大部分模型均为非线性的, KCF 算法通过寻找一个非线性映射函数  $\varphi(x)$ , 使得映射后的样本在新空间中线性可分<sup>[11]</sup>。令  $\omega = \sum_i \alpha_i \varphi(x_i)$ , 引入核函数  $k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$ , 则此时:

$$\hat{\alpha} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{xx} + \lambda} \quad (17)$$

其中:  $\hat{\mathbf{k}}^{xx}$  是向量  $\mathbf{x}$  自身的核相关向量。算法采用高斯核函数形式为:

$$k^{xx'} = \exp \left( -\frac{1}{\sigma^2} (\|x\|^2 + \|x'\|^2 - 2\mathbf{F}^{-1}(\hat{x}^* \odot \hat{x}')) \right) \quad (18)$$

$\sigma$  为核函数参数,  $\mathbf{F}^{-1}$  表示傅里叶逆变换。此外 KCF 针对 MOSSE 算法中特征信息只能为单通道的缺陷, 设计了多通道特征的计算方式如下, 也基于此式引入 HOG 特征进行跟踪。

$$k^{xx'} = \exp \left( -\frac{1}{\sigma^2} (\|x\|^2 + \|x'\|^2 - 2\mathbf{F}^{-1}(\sum_c \hat{x}_c^* \odot \hat{x}'_c)) \right) \quad (19)$$

其中:  $c$  表示通道数。算法检测过程中输出响应为:

$$\hat{f}(z) = \hat{\mathbf{k}}^{xx} \odot \hat{\alpha} \quad (20)$$

$\hat{f}(z)$  最大值即为当前帧跟踪目标的位置。同 MOSSE 算法一般对跟踪器模板更新:

$$\hat{\alpha}_i = (1 - \eta) \hat{\alpha}_{i-1} + \eta \hat{\alpha}_i \quad (21)$$

$$\hat{x}_i = (1 - \eta) \hat{x}_{i-1} + \eta \hat{x}_i \quad (22)$$

KCF 算法的大致流程如图 2 所示。

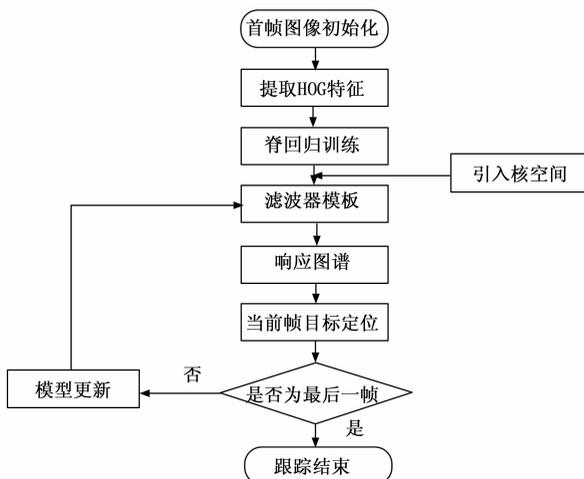


图 2 KCF 算法流程图

## 2.2 DSST 算法

DSST 算法基于 MOSSE 以及 KCF 算法在样本提取时选择融合多维特征: 样本的灰度、颜色以及 HOG 等特征, 在特征提取上对前两种算法做了改进。另外 MOSSE 与 KCF 算法中的跟踪框是固定大小的, 当目标尺度相对于检测器逐渐变大时, 跟踪框只能捕捉到目标的一部分信息; 当目标逐渐变小时框内又会掺杂过多的干扰背景信息。DSST 算法便在目标跟踪过程中对目标进行实时地尺度预测, 一定程度上提升了跟踪算法的精度和鲁棒性。

DSST 算法设计了两个相关滤波器, 分别为实现目标位置跟踪的位置滤波器与实现跟踪框自适应目标尺度变化的尺度滤波器。

论文<sup>[13]</sup>指出该算法使用的尺度估计方法可以移植到任意算法中。设输入图像为  $d$  维特征  $f$ ,  $f^l$  为第  $l$  维特征。结合式 (4) 和 (9) 式容易得到最优相关滤波器  $h$  为:

$$\epsilon = \left\| \sum_{l=1}^d h^l * f^l - g \right\|^2 + \lambda \sum_{l=1}^d \|h^l\|^2 \quad (23)$$

转换到频域上, 并将分子分母单独表示:

$$H^l = \frac{\bar{\mathbf{G}} \mathbf{F}^l}{\sum_{k=1}^d \bar{\mathbf{F}}^k \mathbf{F}^k + \lambda} = \frac{A_t^l}{B_t + \lambda} \quad (24)$$

其中:  $\bar{\mathbf{G}} \mathbf{F}^k$  分别表示对应的共轭矩阵,  $t$  表示当前帧数。对分子分母分别更新:

$$A_t^l = (1 - \eta) A_{t-1}^l + \eta \bar{\mathbf{G}} \mathbf{F}_t^l \quad (25)$$

$$B_t = (1 - \eta) B_{t-1} + \eta \sum_{k=1}^d \bar{\mathbf{F}}_t^k \mathbf{F}_t^k \quad (26)$$

通过求解最大响应值确定新来帧图像信息  $Z$  中的目标位置信息:

$$y = \mathbf{F}^{-1} \left\{ \frac{\sum_{l=1}^d \bar{A}_t^l Z^l}{B_t + \lambda} \right\} \quad (27)$$

DSST 算法将位置估计与尺度估计分开处理, 其认为在连续的两帧图像中, 位置的变化程度往往比尺度变化大。因此先使用二维的位置相关滤波器确定当前帧响应峰值区域, 将其视作目标中心位置, 然后通过一维的尺度相关滤波器以此位置为中心点, 获取多组不同尺度的候选区域, 最终找到最适配的尺度信息。尺度的选择原则为:

$$\alpha^n P \times \alpha^n R, n \in \left\{ \left[ -\frac{S-1}{2} \right], \dots, \left[ \frac{S-1}{2} \right] \right\} \quad (28)$$

其中:  $P, R$  分别为目标在前一帧的宽高,  $\alpha=1.02$  为尺度因子,  $S=33$  为尺度的数量, 使用这 33 个尺度样本特征对尺度滤波器进行训练。

DSST 算法的大致流程如图 3 所示。

## 2.3 BACF 算法

虽然训练相关滤波器的样本数量越多效果越好, 但正样本的周围不能包含太多背景信息, 否则分类器会误认为这些背景信息也是正样本, 从而导致分类失败。而 BACF 算法在传统相关滤波算法的框架基础上扩大了循环矩阵采样的区域, 并创造性地在每个样本上裁剪出有效的样本区域, 样本的数量和质量均得到提升。

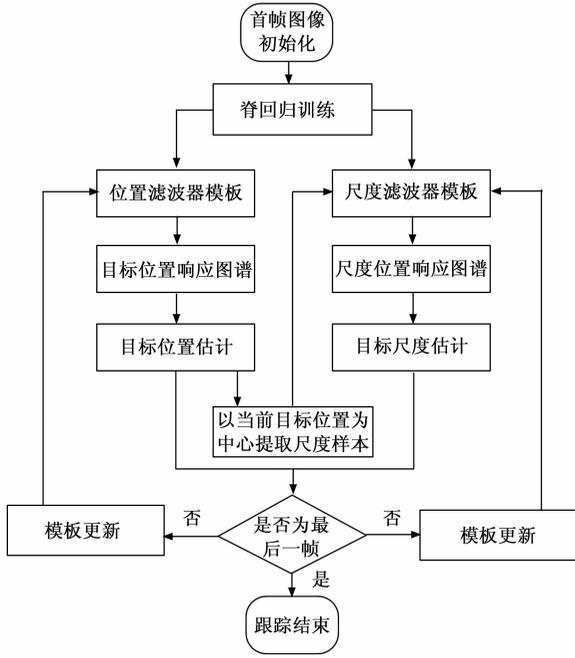


图 3 DSST 算法流程图

假设目标区域的初始大小为  $M \times N$ ，算法扩大采样区域 5 倍并将其延伸为正方形区域，此时样本  $x$  与样本标签  $y$  大小均为  $T \times T$ ， $T = \sqrt{5MN}$ 。设滤波器为  $h$ ，大小为  $D \times D$ ， $D = \frac{T}{l}$ ， $l$  为特征采样率，一般情况下  $T \gg D$ 。基于式 (4) 和式 (9) 对传统的相关滤波器设计改为：

$$E(h) = \frac{1}{2} \sum_{j=1}^D \left\| y(i) - \sum_{k=1}^K h_k^T P x_k \right\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^K \| h_k \|_2^2 \quad (29)$$

其中： $K$  为特征通道数，由于 BACF 算法采用 31 维的 HOG 特征，因此  $K=31$ 。 $P$  为 0, 1 的二值矩阵，大小为  $D \times T$ ，目的即是在训练样本  $x_k$  中裁剪出  $D$  个元素。将  $P, x_k$  合并为  $X_k$ ，表示裁剪后的样本。上式转换至频域：

$$\hat{E}(h, \hat{g}) = \frac{1}{2} \| \hat{y} - \hat{X} \hat{g} \|_2^2 + \frac{\lambda}{2} \| h \|_2^2 \quad (30)$$

其中： $\hat{g} = \sqrt{T} (FP^R \otimes \mathbf{I}_K) h$  是为使用增广拉格朗日算法 (ALM, augmented lagrange method) 增广拉格朗日算法<sup>[26]</sup>构造的一个辅助变量。 $\mathbf{I}_K$  是大小为  $K \times K$  的单位矩阵。

通过交替方向乘法 (ADMM, alternating direction method of multipliers) 交替方向乘法<sup>[27]</sup>将问题转化为求解滤波器和辅助变量两个子问题。将得到的滤波器模板与检测到的当前帧目标区域进行卷积：

$$\hat{G} = \hat{h} \hat{X} \quad (31)$$

对  $G$  作傅里叶逆变换得到最终响应图，最大响应值处即为跟踪目标估计位置。模型更新策略与传统相关滤波器一般：

$$\hat{h}_i = \eta \hat{h}_i + (1 - \eta) \hat{h}_{i-1} \quad (32)$$

BACF 算法的大致流程如图 4 所示。

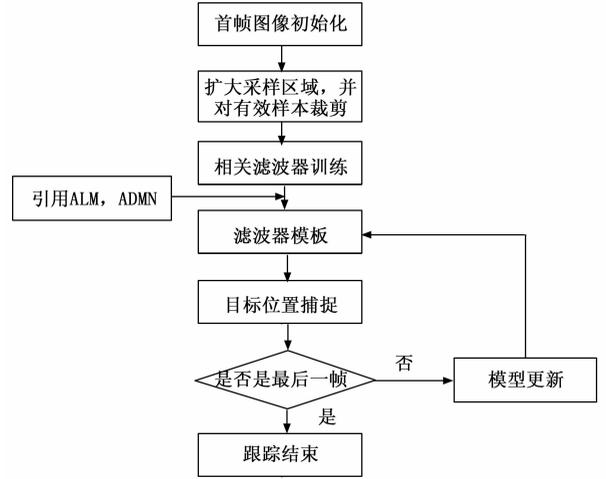


图 4 BACF 算法流程图

### 3 仿真实验与分析

#### 3.1 实验环境

本文针对第 2 节介绍的 3 种算法在 MATLAB 平台上进行了仿真实验。硬件实验环境为 Intel (R) Core (TM) i5-4590 CPU，主频为 3.3 GHz，内存为 16 GB 的计算机；软件环境为 MATLAB R2021a，64 位 Windows10 操作系统。

#### 3.2 数据集介绍

目标跟踪检测基准 (OTB, object tracking benchmark) 数据集是视觉目标跟踪领域公认的测试数据集之一，本文选用 OTB100。它包含 100 个视频序列及其对应地每帧图像中目标标签的 ground\_truth.txt 文本文件。

OTB 数据集的评价标准通常是两个基本参数：中心位置误差和区域重叠面积比率。中心位置误差是指跟踪目标的中心位置和人工标注的准确位置之间的欧式距离，用于生成精确度曲线图；区域重叠面积比率是指通过计算跟踪算法得到的边界框与人工标注的准确边界框之间的面积重叠比，用于生成成功率曲线图。

以 OTB 数据集中对应视频序列的 ground\_truth.txt 文本中目标的位置初始化第一帧图像，然后运行跟踪算法得到精度与成功图。这种方法称为一次性评估 (OPE, one-pass evaluation) 一次性评估。为进行鲁棒性评估，OTB 提出在对测试视频序列初始化时进行扰乱：从不同帧开始跟踪的时间鲁棒性评估 (TRE, temporal robustness evaluation)，其通过视频序列在时间轴上平均找出 20 个点作为起点，终点仍为最后一帧，对 20 段视频序列运行算法，绘制平均重叠率和像素误差图；以不同边界框开始跟踪的空间鲁棒性评估方法 (SRE, spatial robustness evaluation)，其通过对 ground\_truth.txt 中的边界框使用 8 个空间移位以及 4 个比例变化来产生 12 种跟踪框，即对视频序列进行了 12 次评估，绘制平均重叠率和像素误差图。

#### 3.3 实验结果

基于 OTB100 数据集对 KCF、DSST 以及 BACF 算法

从 OPE、SRE、TRE 三种评估角度分别测试。

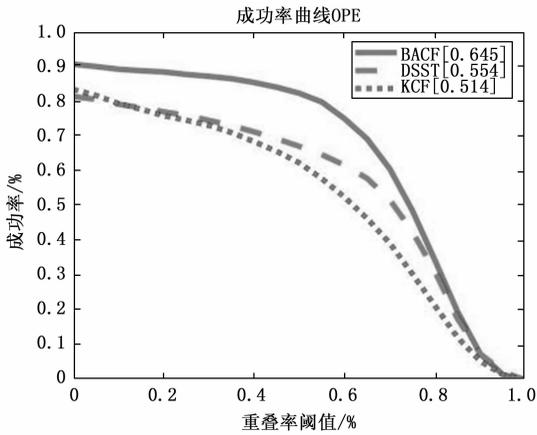


图 5 OPE 评估成功率

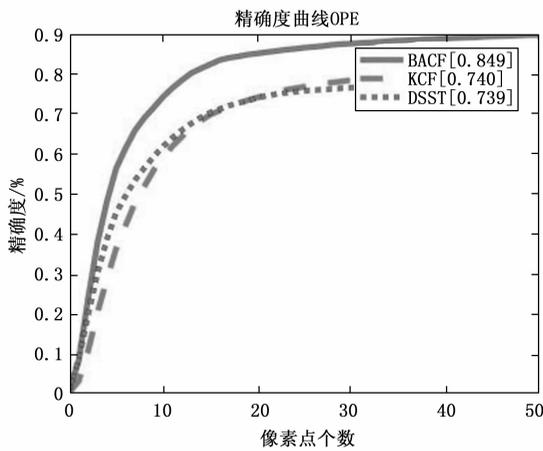


图 6 OPE 评估精度图

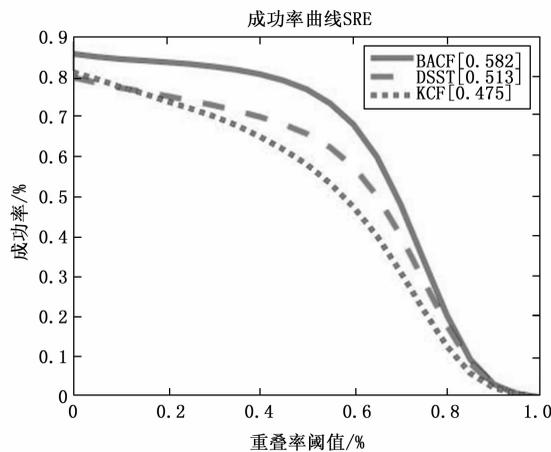


图 7 SRE 评估成功率

鉴于精度图仅使用边界框位置,不能反映目标大小与尺度变换带来的影响,因此成功率曲线图相比精度图更可靠。比较数据过程中,在精度相差不大的情况下,优先以成功率考量性能。

在 OPE 评价标准中:成功率方面,BACF 算法的平均成功率(64.5%)比 DSST 算法(55.4%)高 9.1%,比

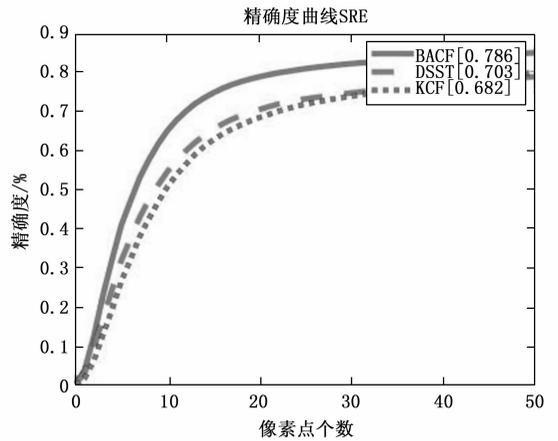


图 8 SRE 评估精度图

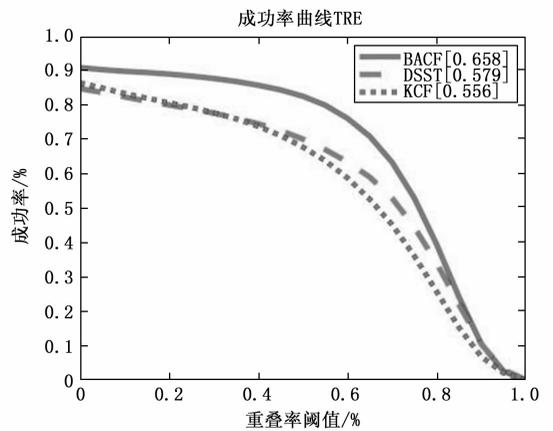


图 9 TRE 评估成功率

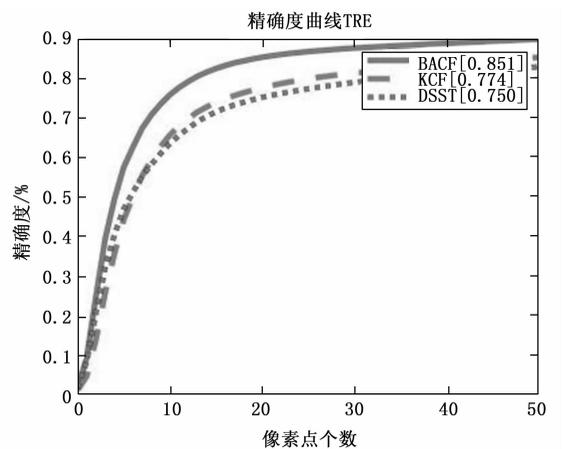


图 10 TRE 评估精度图

KCF 算法(51.4%)高 13.1%;精度度方面,BACF 算法的平均精度度(80.4%)比 KCF 算法(74.0%)高 6.4%,比 DSST 算法(73.9%)高 6.5%。因此 BACF 算法跟踪效果最好,DSST 算法次之,KCF 算法最差。

在 SRE 评价标准中:成功率方面,BACF 算法的平均成功率(58.2%)比 DSST 算法(51.3%)高 6.9%,比 KCF 算法(47.5%)高 10.7%;精度度方面,BACF 算法

的平均精确度 (78.6%) 比 DSST 算法 (70.3%) 高 8.3%, 比 KCF 算法 (68.2%) 高 10.4%。因此 BACF 算法跟踪效果最好, DSST 算法次之, KCF 算法最差。

在 TRE 评价标准中: 成功率方面, BACF 算法的平均成功率 (65.8%) 比 DSST 算法 (57.9%) 高 7.9%, 比 KCF 算法 (55.6%) 高 10.2%; 精确度方面, BACF 算法的平均精确度 (85.1%) 比 KCF 算法 (77.4%) 高 7.7%, 比 DSST 算法 (75.0%) 高 10.1%。因此 BACF 算法跟踪效果最好, DSST 算法次之, KCF 算法最差。

综合考虑准确率和成功率, KCF 算法总体性能较低, 在目标发生尺度变换该算法容易丢失跟踪目标, 跟踪性能大大降低。虽然 DSST 算法针对目标的尺度变换额外增加了一个尺度滤波器, 但当尺度变换过大时仍会导致尺度估计精度低。BACF 算法得益于对样本的高效处理, 时间、空间鲁棒性等效果均为最佳。实验同样对算法的速度进行了对比。

表 1 3 种算法的帧率

算法	KCF	DSST	BACF
FPS	154.36	25.34	28.37

可以看出 KCF 算法的速度最快, BACF 算法次之, 最后是 DSST 算法。BACF 算法由于增加了样本数量以及额外的裁剪工作, 速度下降许多; 而 DSST 算法增加的尺度滤波器在多通道滤波的情况下, 仍需在每个通道检测 33 个尺度的图像块, 速度显然降低严重。

#### 4 结束语

由于相关滤波跟踪算法的准确性与实时性较好, 大量算法分别从尺度自适应、特征提取、边界效应等角度进行改进。对目标跟踪算法的研究依然围绕准确性和实时性展开。随着深度学习跟踪算法地不断推出, 将深度特征与相关滤波结合的算法将是发展趋势之一。但深度特征给算法带来高精确度的同时, 也不可避免地给算法的跟踪速度增加负荷, 往往难以满足目标跟踪对实时性的要求。将传统手工特征与深度特征融合时, 考虑如何在损失精度的前提下提升算法速度, 也将是研究热点。

#### 参考文献:

- [1] 谭飞杨. 视觉目标跟踪算法及其 CUDA 优化加速研究 [D]. 西安: 西安电子科技大学, 2017.
- [2] 陈凯. 人机交互系统中的目标跟踪算法研究 [D]. 西安: 长安大学, 2015.
- [3] KRISTAN M, PFLUGFELDER R, LEONARDIS A, et al. The visual object tracking VOT2013 challenge results [C] // Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops, IEEE, 2013: 98-111.
- [4] 韩瑞泽, 冯伟, 郭青. 视频单目标跟踪研究进展综述 [J]. 计算机学报, 2022, 45 (9): 1877-907.
- [5] COLLINS R T. Mean-shift blob tracking through scale space [C] // Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings, IEEE, 2003: II-234.
- [6] NUMMIARO K, KOLLER-MEIER E, GOOL L V. An adaptive color-based particle filter [J]. Image Vis Comput., 2003, 21: 99-110.
- [7] KALAL Z, MIKOLAJCZYK K, MATAS J. Tracking-learning-detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34 (7): 1409-22.
- [8] 曹小波. 基于 GPU 加速的视觉目标跟踪算法研究 [D]. 西安: 西安电子科技大学, 2019.
- [9] BOLME D S, BEVERIDGE J R, DRAPER B A, et al. Visual object tracking using adaptive correlation filters [C] // Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010: 2544-2550.
- [10] HENRIQUES J F, CASEIRO R, MARTINS P, et al. Exploiting the circulant structure of tracking-by-detection with kernels [C] // Proceedings of the Computer Vision-ECCV 2012 / Springer, 2012: 702-715.
- [11] 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): 583-96.
- [12] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection [C] // Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005: 886-893.
- [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): 1561-75.
- [14] DANELLJAN M, HAGER G, KHAN F S, et al. Learning spatially regularized correlation filters for visual tracking [C] // Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015: 4310-4318.
- [15] LI F, TIAN C, ZUO W, et al. Learning spatial-temporal regularized correlation filters for visual tracking [C] // Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018: 4904-4913.
- [16] BERTINETTO L, VALMADRE J, GOLODETZ S, et al. Staple; complementary learners for real-time tracking [C] // Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 4904-4913.
- [17] MUELLER M, SMITH N, GHANEM B. Context-aware correlation filter tracking [C] // Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 1387-1395.
- [18] GALOOGAHI H K, FAGG A, LUCEY S. Learning background-aware correlation filters for visual tracking [C] // Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017: 1144-1152.

(下转第 237 页)