

目标跟踪算法综述

孟 磊¹ 杨 旭¹

摘 要 目标跟踪一直以来都是计算机视觉领域的关键问题, 最近随着人工智能领域的飞速发展, 运动目标跟踪问题得到了越来越多的关注. 本文对主流目标跟踪算法进行了综述, 首先介绍了目标跟踪中常见的问题, 并由时间顺序对目标跟踪算法进行了分类: 早期的经典跟踪算法, 基于相关滤波的跟踪算法, 以及基于深度学习的跟踪算法. 接下来, 对每一类中经典的跟踪算法的原始版本和各种改进版本做了介绍、分析以及比较. 最后, 使用 OTB-2013 数据集对目标跟踪算法进行测试, 并对结果进行分析, 得出了以下结论: 1) 相比于光流法, Kalman, Meanshift 等传统算法, 相关滤波类算法跟踪速度更快, 深度学习类方法精度高. 2) 具有多特征融合以及深度特征的追踪器在跟踪精度方面的效果更好. 3) 使用强大的分类器是实现良好跟踪的基础. 4) 尺度的自适应以及模型的更新机制也影响着跟踪的精度.

关键词 目标跟踪, 特征选择, Meanshift, 尺度变化, 相关滤波

引用格式 孟磊, 杨旭. 目标跟踪算法综述. 自动化学报, 2019, XX(X): X-X

DOI 10.16383/j.aas.c180277

A Survey of Object Tracking Algorithms

MENG Lu¹ YANG Xu¹

Abstract Object tracking has always been a key issue in the computer vision field. Recently, with the rapid development of artificial intelligence, the issue of moving object tracking has attracted more and more attention. This paper reviews the main object tracking algorithms, firstly, we introduce the common problems in object tracking and classify the object tracking algorithms into three groups: early object tracking algorithms, kernel correlation filtering (KCF) object tracking algorithms, deep learning object tracking algorithms. Then, according to the three groups, we introduce and analyze many famous object tracking algorithms and their following improved versions. Finally, we analyze and compare the performance from many object tracking algorithms using dataset OTB-2013, and conclude that: 1) Compared with optical flow method, Kalman, Meanshift and other early algorithms, the tracking speed of KCF-based algorithms are faster and deep learning based algorithms have higher accuracy. 2) The tracking algorithms with multiple feature fusion or deep features have higher tracking accuracy. 3) Powerful classifiers are the basis for good tracking results. 4) Scale adaptation and updating mechanism also affect tracking accuracy.

Key words Object tracking, image feature selection, Meanshift, scale transformation, kernelized correlation filters

Citation Meng Lu, Yang Xu. A survey of object tracking algorithms. *Acta Automatica Sinica*, 2019, XX(X): X-X

运动目标跟踪一直以来都是一项具有挑战性的工作, 也是研究的热点方向. 现阶段, 随着硬件设施的不断完善和人工智能领域的快速发展, 运动目标跟踪技术越来越重要. 目标跟踪在现实生活中有很多应用, 包括交通视频监控、运动员比赛分析、智能人机交互^[1]、跟踪系统的设计^[2]等. 由于在目标跟踪中存在形态变化、图像分辨率低、背景复杂等情况, 因此研究出一个性能优良的跟踪器势在必行.

早期的目标跟踪算法主要是根据目标建模或者对目标特征进行跟踪, 主要的方法有: 1) 基于目标

模型建模的方法: 通过对目标外观模型进行建模, 然后在之后的帧中找到目标. 例如, 区域匹配, 特征点跟踪, 基于主动轮廓的跟踪算法, 光流法等. 最常用的是特征匹配法, 首先提取目标特征, 然后在后续的帧中找到最相似的特征进行目标定位, 常用的特征有: SIFT^[3] 特征, SURF^[4] 特征, Harris 角点^[5] 等. 2) 基于搜索的方法: 随着研究的深入, 人们发现基于目标模型建模的方法^[6] 对整张图片进行处理, 实时性差. 人们将预测算法加入跟踪中, 在预测值附近进行目标搜索, 减少了搜索的范围. 常见一类的预测算法有 Kalman^[7] 滤波, 粒子滤波^[8] 方法. 另一种减小搜索范围的方法是内核方法: 运用最速下降法的原理, 向梯度下降方向对目标模板逐步迭代, 直到迭代到最优位置. 诸如, Meanshift^[9-10]、Camshift^[11] 算法. 综上所述, 传统的目标跟踪算法存在两个致命的缺陷: 1) 没有将背景信息考虑在内, 导致在目标遮挡, 光照变化以及运动模糊等干扰下容易出现跟踪

收稿日期 2018-05-03 录用日期 2018-07-16

Manuscript received May 3, 2018; accepted July 16, 2018

国家自然科学基金 (61101057) 资助

Supported by National Natural Science Foundation of China (61101057)

本文责任编辑 桑农

Recommended by Associate Editor SANG Nong

1. 东北大学信息科学与工程学院 沈阳 110000

1. College of Information Science and Engineering, Northeastern University, Shenyang 110000

失败: 2) 跟踪算法执行速度慢 (每秒 10 帧左右), 无法满足实时性的要求。

在过去的 4 年中, 目标跟踪取得了突破性的进展, 主要由于将通信领域的相关滤波 (衡量两个信号的相似程度) 引入到了目标跟踪中。一些基于相关滤波的跟踪算法 (MOSSE^[12]、CSK^[13]、KCF^[14]、BACF^[15]、SAMF^[16]) 等, 也随之产生, 速度可以达到数百帧每秒, 可以广泛的应用于实时跟踪系统中。其中不乏一些跟踪性能优良的跟踪器, 诸如 SAMF、BACF 在 OTB^[17] 数据集和 VOT2015^[18] 竞赛中取得优异成绩。近期, 人们开始将深度学习方法应用到目标跟踪中, 取得了很好的效果, ECO^[19] 在 VOT2017^[20] 竞赛中获得了冠军。

据我们所知^[21], 目前还没有关于目标跟踪进展完整的文献综述和详细调查。在这项工作中, 我们由时间顺序回顾现有的跟踪算法, 并根据最新算法在 OTB 数据集上的结果进行分析。此外, 我们从以下几个方面对目标跟踪算法进行了分析、总结和综述:

1) 分析了生成式追踪和判别式追踪两类方法的基本原理, 并介绍了每种类型方法的跟踪算法。

2) 总结了早期跟踪算法的发展过程以及 Mean-shift 算法的改进策略。

3) 在文中, 详细解释了时域到频域的转换原理, 以及相关滤波算法如何减少计算的复杂度, 从而提升计算速度。

4) 本文中通过特征的选择, 尺度的变化, 框架的改变三个方面对相关滤波改进策略进行分析, 并列举了一系列改进的方法。

5) 文中分析深度学习类方法的结构特点, 通过对比了深度学习方法和相关滤波方法, 分析了今后的发展方向。

本文结构如下: 在第一节中, 我们简要介绍了跟踪中常见的问题以及生成式追踪和判别式追踪的特点。在第二节中, 我们简要介绍了几种经典的跟踪方法, 并从三个方面对 Meanshift 的改进算法进行了分析。在第三节中, 首先对相关滤波的理论基础及其数学推导进行介绍, 然后从特征的选择, 尺度的变化, 模型的改变三个方面对现有的相关滤波器进行了回顾。第四节中, 简要介绍深度学习类跟踪算法并对其进行分类讨论。第五节中, 针对基于相关滤波的目标跟踪算法和基于深度学习的目标跟踪算法, 进行比较和分析。总结了全文并展望目标跟踪算法下一步的发展方向。

1 背景和相关工作

1.1 跟踪中常见的问题

研究人员对运动目标跟踪技术的探索已有多, 目标跟踪算法的性能逐渐改善, 但是在复杂条件下对运动目标实现实时, 稳定的跟踪仍存在很大的挑战, 面临的挑战主要如下^[22]:

1) 形态变化. 姿态变化是目标跟踪中常见的干扰问题. 运动目标发生姿态变化时, 会导致它的特征以及外观模型发生改变, 容易导致跟踪失败. 例如: 体育比赛中的运动员, 马路上的行人。

2) 尺度变化. 尺度的自适应也是目标跟踪中的关键问题. 当目标尺度缩小时, 由于跟踪框不能自适应跟踪, 会将很多背景信息包含在内, 导致目标模型的更新错误: 当目标尺度增大时, 由于跟踪框不能将目标完全包括在内, 跟踪框内目标信息不全, 也会导致目标模型的更新错误. 因此, 实现尺度自适应跟踪是十分必要的。

3) 遮挡与消失. 目标在运动过程中可能出现被遮挡或者短暂的消失情况. 当这种情况发生时, 跟踪框容易将遮挡物以及背景信息包含在跟踪框内, 会导致后续帧中的跟踪目标漂移到遮挡物上面. 若目标被完全遮挡时, 由于找不到目标的对应模型, 会导致跟踪失败。

4) 图像模糊. 光照强度变化, 目标快速运动, 低分辨率等情况会导致图像模型, 尤其是在运动目标与背景相似的情况下更为明显. 因此, 选择有效的特征对目标和背景进行区分非常必要。

1.2 目标跟踪算法的基本分类

近 20 年, 涌现出大量的目标跟踪算法, 根据其工作原理, 我们将其分为生成式模型和鉴别式模型两种。

1.2.1 生成式模型

早期的工作主要集中于生成式模型跟踪算法的研究, 如光流法^[23-24], 粒子滤波^[8], Meanshift 算法^[9-10]、Camshift^[11] 算法等. 此类方法首先建立目标模型或者提取目标特征, 在后续帧中进行相似特征搜索. 逐步迭代实现目标定位. 但是这类方法也存在明显的缺点, 就是图像的背景信息没有得到全面的利用. 且目标本身的外观变化有随机性和多样性特点, 因此, 通过单一的数学模型描述待跟踪目标具有很大的局限性. 具体表现为在光照变化, 运动模糊, 分辨率低, 目标旋转形变等情况下, 模型的建立会受到巨大的影响, 从而影响跟踪的准确性; 模型的建立没有有效地预测机制, 当出现目标遮挡情况时, 不能够很好地解决。

1.2.2 鉴别式模型

鉴别式模型是指, 将目标模型和背景信息同时考虑在内, 通过对比目标模型和背景信息的差异, 将目标模型提取出来, 从而得到当前帧中的目标位置. 文献在对跟踪算法的评估中发现^[25], 通过将背景信

息引入跟踪模型, 可以很好的实现目标跟踪. 因此鉴别式模型具有很大的优势. 2000 年以来, 人们逐渐尝试使用经典的机器学习方法训练分类器, 例如 MIL^[26], TLD^[27], 支持向量机^[28], 结构化学习^[29], 随机森林^[30], 多实例学习^[31], 度量学习^[32]. 2010 年, 文献 [12] 首次将通信领域的相关滤波方法引入到目标跟踪中. 作为鉴别式方法的一种, 相关滤波无论在速度上还是准确率上, 都显示出更优越的性能. 然而, 相关滤波器用于目标跟踪是在 2014 年之后. 自 2015 年以后, 随着深度学习领域的广泛应用, 人们开始将深度学习领域用于目标跟踪.

2 早期的跟踪算法

运动目标跟踪, 首先对目标进行有效地表达. 然后在接下来的视频序列的每一帧中找到相似度与目标最大的区域, 从而确定目标在当前帧中的位置. 早期的生成式方法主要有两种思路: 1) 依赖于目标外观模型. 通过对目标外观模型进行建模, 然后在之后的帧中找到目标. 例如: 光流法. 2) 不依赖于目标外观模型. 选定目标建立目标模型, 然后在视频中搜索找到目标模型. 例如: Meanshift.

2.1 光流法

2.1.1 光流法思想

光流法 (Lucas-kanade) 的概念首先在 1950 年提出, 它是针对外观模型对视频序列中的像素进行操作. 通过利用视频序列在相邻帧之间的像素关系, 寻找像素的位移变化来判断目标的运动状态, 实现对运动目标的跟踪. 但是, 光流法适用的范围较小, 需要满足三种假设: 图像的光照强度保持不变; 空间一致性, 即每个像素在不同帧中相邻点的位置不变, 这样便于求得最终的运动矢量; 时间连续. 光流法适用于目标运动相对于帧率是缓慢的, 也就是两帧之间的目标位移不能太大.

2.1.2 基于光流法的改进

随后, 人们开始在光流法的基础上进行了改进. 文献 [33] 中将 Harris 角点检测与光流法相结合, 将光流法中的像素换成 Harris 特征点进行目标跟踪, 这样减少了跟踪像素的数量, 使得算法的复杂度降低, 处理速度加快. 同时, 相比于普通的像素, Harris 角点对目标的描述更好, 具有很强的鲁棒性. 相似的, 文献 [34] 中将光流法与 SIFT 特征进行融合, 提取 SIFT 特征进行跟踪, 取得了很好的效果. 文献 [35] 中首次提出人造光流的概念, 将真实光流与人造光流进行比较, 通过矢量差判断目标的运动. 文献 [36] 中, 为了提高算法匹配的准确性, 在光流法的框架上引入了前景约束.

然而, 基于光流的目标跟踪有很多的缺点. 它对

视频中所有的像素进行计算, 实时性差. 因此, 采用一定的缩小计算范围的算法, 具有重大意义. 其中常见的算法有 Kalman 滤波, 粒子滤波等.

2.2 Kalman 滤波

Kalman 滤波是一种能够对目标的位置进行有效预测的算法. 它建立状态方程, 将观测数据进行状态输入, 对方程参数进行优化. 通过对前 n 帧数据的输入, 可以有效的预测第 n 帧中目标的位置, Kalman 估计也叫最优估计. 因此, 在目标跟踪过程中, 当目标出现遮挡或者消失时, 加入 Kalman 滤波可以有效地解决这种问题. 缺点是 Kalman 滤波只适合于线性系统, 适用范围小.

针对 Kalman 滤波适用范围小这一问题, 人们提出了粒子滤波的方法. 粒子滤波的思想源于蒙特卡洛思想, 它利用特征点表示概率模型. 这种表示方法可以在非线性空间上进行计算, 其思想是从后验概率中选取特征表达其分布. 最近, 人们也提出了改进平方根容积卡尔曼滤波的方法来减小误差^[37], 从而实现精准跟踪.

2.3 核方法

2.3.1 基本思想

另一类基于搜索的方法是核方法. 核跟踪方法是目标跟踪的主要方法, 应用非常广泛. 例如 Measht, Camshift 算法, 它直接运用最速下降法的原理, 向梯度下降方向对目标模板逐步迭代, 直到迭代到最优位置. 它的核心就是一步一步迭代寻找最优点, 在跟踪中, 就是为了寻找相似度值最大的候选区间. 我们这样定义方向矢量: 满足计算范围内的 t 个样本点与区域几何中心的矢量和. 其中 x_i 为样本点, x 为区域的几何中心. S_h 为符合计算区域的半径为 r 的球区域.

$$M_h(x) = \frac{1}{t} \sum_{x_i \in S_h} (x_i - x) \quad (1)$$

$$S_h(x) = \left\{ y : (y - x)^T (y - x) \leq r^2 \right\} \quad (2)$$

Meanshift 就是对样本点与中心点的矢量差求平均值. 矢量和的方向就是概率密度增加的方向, 沿着概率密度增加的方向移动向量, 逐步迭代直到找到最优解. 但是这种搜索方法存在缺陷, 只对样本点进行计算, 无论距离中心点的远近, 其贡献是一样的. 当目标出现遮挡或运动模糊时, 外层的特征容易受到背景的影响, 其准确性降低. 针对这一情况, 我们应该赋予不同的采样点不同的权值, 离中心点越近, 权值越高; 反之亦然. 因此, 我们应该选择合适的系数来提高跟踪算法的鲁棒性. 为了解决这一问题, 作者^[9]将 Epannechnikov 核函数引入到

Meanshift 中.

$$K(x) = \begin{cases} c(1 - \|x\|^2) & \|x\| \leq 1 \\ 0 & \text{其他} \end{cases} \quad (3)$$

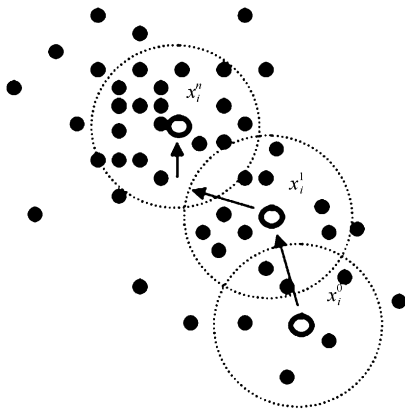


图 1 Meanshift 跟踪原理图

Fig. 1 The tracking schematic of Meanshift

图 1 表达了 Meanshift 的跟踪原理示意图: 可以看出, 算法通过逐步迭代寻找概率密度最大的方向. 当偏移量小于某一设定阈值时, 则可以认为样本迭代到最佳位置, 达到目标跟踪的目的. 跟踪的过程分为三个步骤:

1) 目标模型的建立: 在初始帧中选中目标模型, 对初始帧中的目标区域建立 RGB 颜色空间直方图. 目标的模型表示如下:

$$q_u = c \sum_{i=1}^n K(\|x_i^* - u\|^2) \delta[b(x_i) - u] \quad (4)$$

其中 c 是归一化系数, K 表示内核函数, 在本算法中使用 Epanechnikov 核函数, δ 函数是判断点 x_i 是否属于目标区域, 在区域内为 1, 不在区域内为 0.

2) 模型的搜索: 在后续帧中, 在前一帧的基础上逐步搜索, 得到目标区域的模型如下. 其中, y 为窗口的中心坐标, r 表示核窗口的宽度.

$$p_u(y) = c \sum_{i=1}^n K\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \quad (5)$$

3) 相似度判别: 相似度判别式是判断是否继续迭代的条件. 当目标模型与跟踪模型的相似度小于特定的阈值时, 迭代停止, 得到待跟踪的目标区域. 引入 BH 系数进行判断, 系数值越大, 相似度越高. S_u 为目标模板, Q_u 为候选区域模板.

$$\rho[p, q] = \sum_{u=1}^m \sqrt{p_u * q_u} \quad (6)$$

Meanshift 算法虽然实现了较好的跟踪效果, 但是对目标尺度的变化不能进行自适应的跟踪. 针对

这一缺陷, 人们提出了 Camshift 算法. Camshift 算法是在 Measnhift 基础上的改进, 在其中加入了尺度自适应机制, 可以根据目标的大小自动调节跟踪框的尺度, 算法的流程图如图 2: 首先, Camshift 算法中使用的是 HSV 颜色直方图. 这是由于 HSV 颜色直方图具有较强的鲁棒性. 其次, 根据所求得的颜色直方图将初始输入图像转化为概率分布图像. 最后是 Meanshift 迭代过程. 进行 Meanshift 迭代, 将前一帧的输出作为后续帧的输入. 其思想与 Meanshift 相同, 移动窗口中心到质心, 逐步迭代, 直到满足迭代终止条件. 在迭代的过程中, 通过零阶距调整窗口的大小.

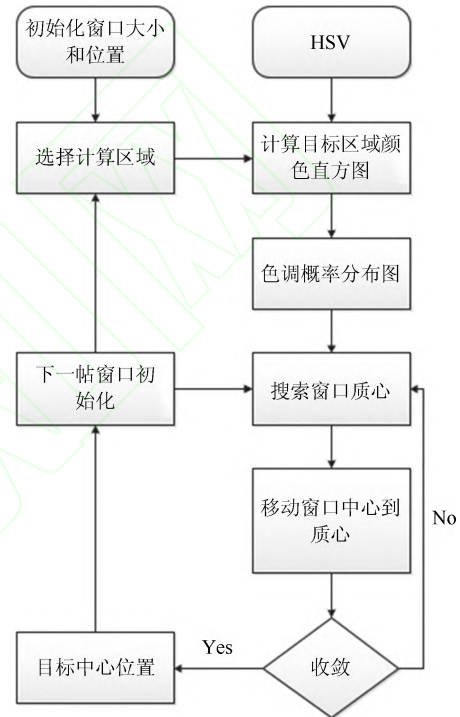


图 2 Camshift 算法流程图

Fig. 2 Camshift algorithm flow chart

2.3.2 改进方向

随着核方法的应用越来越广泛, 近年来基于其改进的算法层出不穷, 主要针对背景复杂, 遮挡与消失和算法的融合三个方面, 改进的结构如图 3 所示.

背景复杂: 追踪过程中视频序列会存在光照干扰, 背景混乱等复杂情况. 通过对特征提取的改进, 可以使得跟踪效果得到提高. 文献 [38] 中在 Camshift 算法的基础上扩展了颜色空间, 使得对光照强度变化不敏感. 文献 [39] 中提出了将纹理特征加入到 Meanshift 算法中进行跟踪的方法, 因为纹理特征能够更好的描述目标的表观特征, 能够克服背景颜色相似干扰. 文献 [40] 中使用全局匹配和局部跟踪方法, 提出一个启发式局部异常因子来提高跟踪精度. Harris 角点具有抗干扰, 旋转不变

性等优点, 因此文献 [41] 中将 Harris 角点加入到 Meanshift 算法中. 由于优质角点的引入, 不但减少了计算量, 提高了算法的效率; 还能够提高算法在复杂情况下的跟踪效果.

遮挡和消失: Meanshift 算法中没有模型的预测机制, 当目标被遮挡或者短暂消失时, 跟踪框会将包含的背景等干扰信息当成目标模型进行更新, 从而导致跟踪失败. 文献 [42] 将 Camshift 算法与 Kalman 滤波相结合, 运用 Kalman 滤波的预测机制来优化 Camshift 算法, 在目标短暂消失的情况下仍能很好的跟踪. 在文献 [43] 中, 作者提出了一种基于加权片段的方法来处理部分遮挡, 权重来源于片段和背景颜色之间的差异. 这种方法不仅对光照不敏感, 而且能够很好地解决目标遮挡问题.

多算法的融合: 算法的融合是为了让每个算法优势互补, 提高融合后算法的性能. 文献 [44–45] 中将 Camshift 算法与粒子滤波相结合, 通过粒子滤波选择特征点, 然后将选择的特征点进行 Camshift 跟踪, 这样可以对原始算法进行优化, 但是计算速度变慢. 之后的算法也有将 Meanshift 算法与光流法进行融合^[46], 对每个像素进行操作, 解决运动速度过快问题, 取得了较好的效果. 文献 [47] 提出了一种结合 Meanshift 算法和运动矢量分析的新型算法, 在算法中选择视频序列进行分析, 可以得到目标运动的方向和速度, 为 Meanshift 的迭代指出反向, 减少了迭代次数, 提高算法的效率.

核方法由于其快速收敛的性质在实际中得到越来越广泛的应用, 研究人员逐步将优质的特征, 尺度自适应以及模型的预测机制加入到核方法中并取得了很好的效果. 但是跟踪的速度难以得到本质上的提高, 人们开始考虑使用新的跟踪算法来提高跟踪

精度.

3 相关滤波跟踪算法

在相关滤波方法用于目标跟踪之前, 所有的跟踪算法都是在时域上进行处理. 在运算过程中, 涉及到复杂的矩阵求逆计算, 运算量大, 实时性差. 基于相关滤波的目标跟踪方法将计算转换到频域, 利用循环矩阵可以在频域对角化的性质, 大大减少了运算量, 提高运算速度. KCF 在相关滤波基础上进行了优化, 引入了循环矩阵. 目标跟踪过程中缺少大量的负样本, 通过循环矩阵可以增加负样本的数量, 提高分类器训练的质量. 将高斯核加入到岭回归中, 能够将非线性的问题转换到高维的线性空间中, 简化计算.

3.1 相关滤波跟踪理论

相关是描述两个信号之间的关系, 而卷积是计算激励信号作用于系统所产生的系统零状态响应. 在函数对称情况下, 卷积和相关具有相同的效果^[12].

相关:

$$f \star g = \int_{-\infty}^{\infty} f(\tau) g(\tau + t) d\tau \quad (7)$$

卷积:

$$f * g = \int_{-\infty}^{\infty} f(\tau) g(\tau - t) d\tau \quad (8)$$

基本思想为越是相似的两个目标相关值越大, 也就是视频帧中与初始化目标越相似, 得到的相应也就越大. 我们知道, 时域的卷积可以转化为频域的相乘来实现: 训练的目标就是得到这样的一个滤波器模板, 使得响应输出最大化.

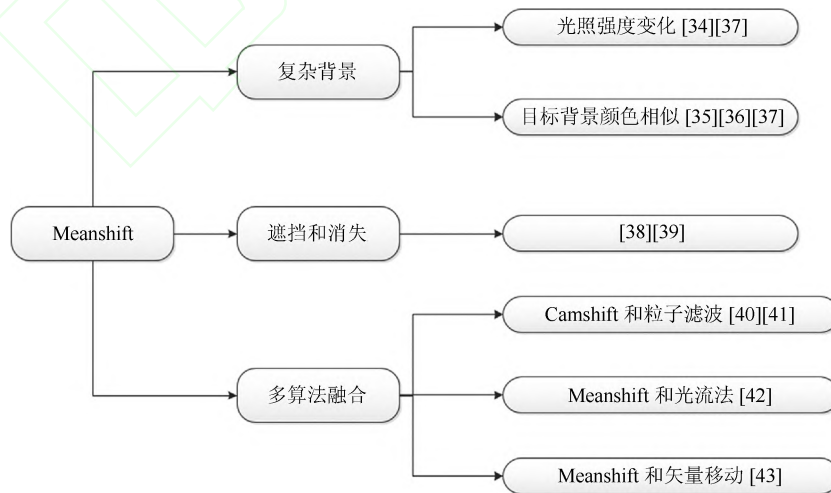


图3 核跟踪算法改进结构图

Fig.3 The improved structure diagram of kernel tracking algorithm

时间域上公式表示 (f : 输入; h : 滤波器; g : 输出):

$$g = f * h \quad (9)$$

频域公式表示 (G, F, H 分别为 g, f, h 对应的傅里叶变换结果):

$$G = F \odot H^* \quad (10)$$

所计算的目标为:

$$H_i^* = \frac{G_i}{F_i} \quad (11)$$

MOSSE 滤波器方法目标如下:

$$\min \sum_{i=1}^n \left| F_i \odot H^* - G_i \right|^2 \quad (12)$$

对目标函数求导, 得到结果如下:

$$H_* = \frac{\sum G_i \odot F_i^*}{\sum F_i \odot F_i^*} \quad (13)$$

3.2 核相关滤波器 (KCF)

在 KCF 中, 作者将目标跟踪问题的求解转化为一个分类问题 (图像中的目标和背景). 这个分类问题的求解应用了岭回归方法^[13-14], 所得到的分类器中包含了矩阵的逆运算, 其运算量复杂, 严重影响跟踪系统的实时性. KCF 在分类器的计算中引入了循环矩阵, 巧妙的规避了矩阵的逆运算, 大大减少了分类器的运算量. 高斯核函数的引入可以将非线性问题转化为高维空间中的线性问题, 使得算法更具有一般性. KCF 可以分为: 模型建立, 在线匹配, 模板更新三个部分.

1) 模型建立. 目标函数为:

$$f(z) = w^T z \quad (14)$$

我们的目标是最小化采样数据与下一帧真实目标位置的距离.

$$\min \sum_i^n (f(X_i) - y_i)^2 + \lambda \|w\|^2 \quad (15)$$

在解方程组中求解极值时, 只要对 w 进行微分, 使导数为 0, 即可得到最小值:

$$w = (X^H X + \lambda)^{-1} X^H y \quad (16)$$

转化在傅里叶域为:

$$\hat{w} = \frac{\hat{X}^* \odot \hat{Y}}{\hat{X}^* \odot \hat{X} + \lambda} \quad (17)$$

但是, 大多数情况下是非线性问题. 我们引入了高维核函数的概念. 在高维空间中非线性问题 w 可以变成一个线性问题.

$$w = \sum_i^n \alpha_i \varphi(X_i) \quad (18)$$

那我们的目标函数就可以表示成:

$$f(z) = w^T z = \sum_{i=1}^n \alpha_i K(z, x_i) \quad (19)$$

其中 k 表示核函数它的定义运算如下:

$$\varphi^T(x) \varphi(x') = K(x, x') \quad (20)$$

将之前求 w 的问题转换成了求 α 的问题:

$$\alpha = (K + \lambda I)^{-1} y \quad (21)$$

对于核方法, 一般不知道非线性函数 $\varphi(x)$ 的详细表达, 而只是刻画在核空间的核矩阵 $\varphi(x) \varphi(x)^T$. 那么令 K 表示核空间的核矩阵. 由核函数得到, 那么:

$$K = \varphi(x) \varphi(x)^T \quad (22)$$

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \quad (23)$$

这里 \hat{k}^{xx} 表示 K 矩阵的第一行的傅里叶变换.

2) 在线匹配. 定义 K^z 是测试样本和训练样本间在核空间的核矩阵. 通过推导^[14] 最终得到各个测试样本的响应:

$$\hat{f} = (\hat{K}^{xz}) * \hat{\alpha} \quad (24)$$

其中, K^{xz} 是指核矩阵 K^z 的第一行, 找到最大值 \hat{f} 对应的位置即为所求.

3) 模板更新. 针对外观模型的变化, 将双线性插值方法加入到目标模型的更新中. 滤波器系数 α 和目标观测模型 x 为:

$$\hat{\alpha}^t = (1 - \gamma) \hat{\alpha}^{t-1} + \gamma \hat{\alpha} \quad (25)$$

$$\hat{x}^t = (1 - \gamma) \hat{x}^{t-1} + \gamma \hat{x} \quad (26)$$

3.3 相关滤波算法发展

根据相关滤波算法的结构, 在本节中, 我们将从特征的选择, 尺度的估计, 以及分类器的模型三个方面进行讨论. 本节的框架图如图 4 所示: 特征分为单一特征和混合特征两类; 尺度的估计包括尺度池估计, 分块估计, 特征点估计三类; 跟踪模型的改进分为模型结构上的改进和模型融合.



图4 相关滤波类算法发展方向

Fig. 4 Development direction of correlation filters algorithm

3.3.1 特征的选取

1) 单特征. 最早的相关滤波器 MOSSE^[12] 使用的是灰度特征. 由于灰度特征的单一性, 使得其优点是跟踪速度快, 可以达到 669 帧每秒, 但缺点是跟踪精度低. 随后的算法 CSK^[13] 是在 MOSSE 的基础上增加了循环矩阵和核函数进行估计, 但在特征上没有做进一步的优化, 仍使用灰度特征, 其目标跟踪速度可达 320 帧每秒. 随后 CN^[48] 是在 2014 年在 CSK 的基础上提出, 它采用的是颜色特征 (color name). 为了提高运算速度, 论文中还用了 PCA 将 11 维的特征降到 2 维, 并改进了模型跟踪方案. KCF^[14] 与 CSK^[13] 是同一作者, 可以说是 KCF 是对 CSK 的进一步完善. KCF 中使用的是 HOG^[49] 特征, 目标的表面纹理特征和轮廓形状能够很好地被 HOG 特征来描述.

2) 多特征. 随着研究的深入, 人们逐渐发现使用单独的特征在跟踪准确率和跟踪速度上存在着性能瓶颈. 因此, 多特征融合成为了目标跟踪的研究热点. SAMF^[16] 是将 CN、灰度特征和 HOG 特征进行矢量链接, 将链接后的特征作为跟踪的特征. STAPLE^[50] 是在 DSST^[51] 算法基础上将特征进行融合, DSST 算法中使用的 HOG 特征, 其对运动模糊、光照变化具有很强的鲁棒性; 而对于目标发生形变时跟踪效果差. 由于目标在运动过程中颜色分布不会改变, 颜色直方图可以很好的处理目标形变情况. 因此, 作者将 HOG 特征与颜色统计特征进行互补, 共同作为目标的输入特征. MOCA^[52] 中使用的是 MC-HOG 特征, 首先, 将输入的 RGB 图像转换

为颜色空间; 然后, 分别从颜色空间中的每个通道提取 HOG; 最后, 将所有 HOG 特征结合在一起, 形成一个三维矩阵或一个长向量, 作为特征矩阵或特征向量.

近年来, 随着深度学习方法的广泛应用, 人们考虑将深度特征运用在目标跟踪算法中. HCF^[53], 将深度学习提取的卷积特征用于目标跟踪. DeepSRDCF^[64] 算法中, 采用单卷积层进行建模, 用输出的深度特征进行目标跟踪, 这样能够更好的区别目标与背景.

由上述的讨论我们可以得出, 目标特征的选择经历了由简单到复杂的过程. 最开始人们使用灰度、颜色、HOG、深度特征等单一的特征. 随着研究的进展, 为了发挥每种特征的优势, 人们将多种特征进行融合, 将融合后的特征作为目标特征. 后续的研究可以考虑更多的有效特征或特征融合方式.

3.3.2 尺度自适应

早期的相关滤波类算法, 如: KCF^[14], CSK^[13], CN^[48] 等, 其目标跟踪尺度是固定的. 当跟踪目标发生旋转, 形变或者尺度变化时, 跟踪器不能自适应的进行跟踪. 跟踪框内会包含很多干扰信息, 从而导致跟踪失败. 因此, 针对这一问题的改进, 主要有以下几种方法.

1) 基于尺度池的方法. 尺度池方法就是将原有的目标进行不同尺度的放大和缩小. 计算变化后的目标, 选择响应值最大的作为跟踪目标, 从而达到尺度自适应追踪.

稳健的尺度估计是视觉对象追踪中的一个挑战

性问题. 为了解决这一难点, 人们提出 SAMF^[16] 算法. 其工作原理如图 5 所示: 跟踪窗口 i 的大小为二维向量 S_i , S_0 为目标的输入, k 为尺度因子, $k = \{k_1, k_2, \dots, k_i\}$, $S_i = k_i * S_0$. 对初始目标进行七个尺度的缩放, 经过相关滤波器后分别得到对应的响应值. 然后比较不同尺度响应值, 并取响应值最大一个, 作为最佳的目标尺度值. 在 SAMF 中七个固定的尺度池: $k = \{0.985, 0.99, 0.995, 1.0, 1.005, 1.01, 1.015\}$ SAMF 中使用最佳的尺度值对目标进行跟踪, 能够解决目标跟踪过程中目标尺度变化的问题, 并在 2014 年的 VOT 视频跟踪竞赛中取得第一名^[54]. DSST^[51] 中提出了一种基于三维尺度空间相关滤波器的联合平移尺度跟踪方法. 在设计滤波器时, 在二维位置平移的基础上增加一维尺度. 两个滤波器工作相互独立, 分别进行目标定位与尺度缩放, 这样可以更加高效的工作, 方便将该算法移植到其他算法中. 追踪时在目标周围矩形区域计算一个特征金字塔, 由于特征金字塔的引入, 使得 DSST 算法的尺度变化有 33 种.

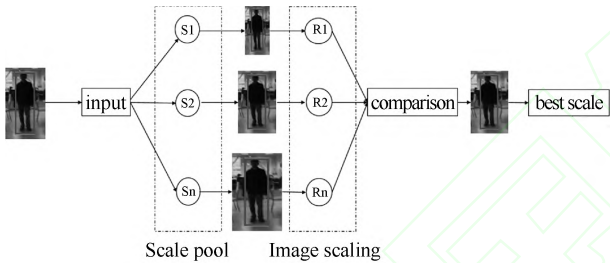


图 5 SAMF 算法原理示意图

Fig. 5 Schematic diagram of SAMF algorithm

2) 基于分块的方法. 分块方法是将目标分成若干个小块, 计算当前帧中块与块之间的距离. 其核心思想是通过判断分块间的距离是否达到所设定的阈值来估计目标的尺度变化.

基于分块的跟踪也能很好的解决自适应跟踪问题. 文献 [55] 中提出了一种基于多个相关滤波器的跟踪方法 (RPAC). 其原理如图 6 所示: 首先在图像中心选择一个块, 在其上下左右依次选择四个大小相同的块. 在跟踪过程中, 计算当前帧中间块和上下左右块中心点的距离. 通过判断各中心点间的距离是否达到尺度更新的阈值来进行尺度自适应跟踪.

PSC^[56] 中, 将目标分成四个小块, 计算初始帧中四个块最大响应位置之间的距离. 然后在后续帧中使用 CN 跟踪器检测四个块中最大响应值的位置. 通过比较位置变化的大小来判断目标尺度的变化. 文献 [57] 中, 将目标分成两个块, 通过比较两个块在当前帧位置的关系来判断目标尺度的变化. 当两个块距离变小时, 目标尺度减小; 当两个块距离变大

时, 目标尺度变大. 通过当前帧两个块之间的距离的变化, 来确定目标尺度的变化.

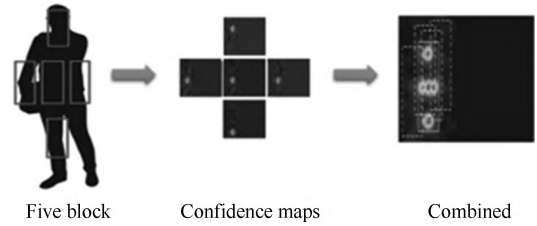


图 6 RPAC 算法原理示意图

Fig. 6 Schematic diagram of RPAC algorithm

3) 基于特征点的跟踪. 随着研究的深入, 人们想到使用特征点进行目标跟踪. 在目标跟踪与特征匹配中, 常用的特征点有 SIFT^[3], SURF^[4], FAST^[58], BRIEF^[59] 等特征点. 在 sKCF^[60] 中, 在视频序列相邻帧中提取特征点, 赋予特征点不同的权值, 离目标中心近, 权重大; 离目标中心远, 权重小. 通过比较特征点的位置变化实现尺度自适应变化. 早期的相关滤波方法 MOSSE, CSK, CN, KCF, 重点工作放在了对目标特征的改进上, 对跟踪尺度没有进行改进. 随着研究的深入, 后续的研究算法大多数解决了尺度变化的问题. SAMF, DSST 中采用了尺度池策略, 对目标尺度进行估计. 但是由于尺度池有限, 有时不能实现准确跟踪. 后续人们采用了分块跟踪策略和特征点跟踪策略, 对尺度池方法进行完善. 特征点位置的变化对目标尺度更加敏感. 相比之下, 针对待跟踪目标的特点, 自适应的定义尺度是最好的选择.

3.3.3 基于跟踪框架的改进

1) 基于模型的改进. 更多的算法集中于相关滤波框架的改进^[61], 如 SRDCF^[62], BACF^[15]. 由于相关滤波是模板类方法, 如果目标快速运动或者发生形变, HOG 特征不能实时的跟上目标的变化, 会产生错误样本造成分类器性能下降, 导致边界效应的产生. SRDCF 针对这一问题, 扩大了搜索区域, 加入空间正则化约束, 其原理图 7 所示, 离目标中心越近, 约束越小; 离目标中心越近越远, 约束越大; 这样减少了背景区域的权重, 提高了分类器的可靠性. SRDCFdecon^[73] 在 SRDCF 基础上将学习率从固定值改为自适应函数. BACF 算法直接从训练分类器入手, 扩大样本数量的同时又增加样本的质量. 先扩大循环采样区域, 这样样本数量会大量增加; 同时为了保证样本质量, 缩小搜索区域, 减少背景信息的干扰.

LMCF^[64] 使用 CF 来优化结构化 SVM, 提高计算速度. 相比于传统 SVM, 结构化 SVM 具有更强大的判别力. 但是在求解结构化 SVM 时计算量大, 实时性差, 因此只能进行稀疏的采样. 相关滤波

思想出现后, 通过使用循环矩阵增加采样范围, 在频域进行计算, 提高了计算速度. 因此可以使用 CF 来对结构化 SVM 进行优化. 同时, 文中对目标检测以及模型更新策略也进行了改进. 如图 8 所示: 在目标检测时, 当响应值为明显的峰值时, 正常进行跟踪; 当响应值为多个峰值时, 说明存在干扰信息, 不能选取盲目的取峰值最高的位置, 应该在产生峰值的位置重新进行比较. 在模型更新时, 当产生多个峰值时, 停止对目标模板的更新, 这样可以减少相似干扰、目标遮挡等情况下错误的目标更新.

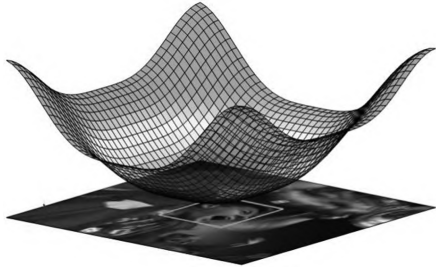


图 7 SRDCF 空间正则化示意图^[62]

Fig. 7 SRDCF space regularization diagram^[62]

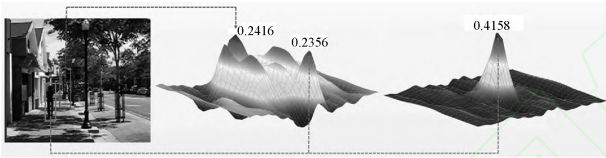


图 8 LMCF 模型在线检测示意图^[64]

Fig. 8 LMCF model online detection diagram^[64]

上述的目标跟踪算法都是在短时间内进行在线跟踪, 若需要进行长时间目标跟踪, 必须配合在线检测机制才能实现. LCT^[65] 中, 在 DSST 二维位置滤波器和一维尺度滤波器的基础上, 增加一个检测器并给检测器设置两个阈值, 用来判断跟踪是否正确. 当响应值小于最小阈值, 说明跟踪不可靠, 应重新跟踪; 当响应值大于最大阈值时, 目标跟踪准确, 应进行在线学习. 通过使用短时间跟踪算法和在线检测器可以实现目标的长时间跟踪.

相比于传统的算法, 随着深度学习领域的发展, 研究人员将深度学习用于目标模型的建立. CF2^[66] 中采用分层卷积进行建模. 其中第一个卷积层提供更精确的定位, 但无法处理待跟踪目标的外观剧烈变化的情况; 第二个卷积层的输出对目标的表现信息, 并且这种表示可以处理待跟踪目标的外观剧烈变化的情况, 但是, 它们无法精确定位目标. 将两个卷积层融合可以得到目标精确的位置.

2) 基于融合模型. 每一种跟踪算法都有自己的优势. 有的是通过提取优良的特征进行跟踪, 有的是引入可变化尺度进行跟踪. 通过将不同的算法进行合理地集成, 可以进行优势互补, 达到更好的效

果. 文献 [67] 中提出了一种基于集成 (EBT) 的方法, 将多种算法结合在一起, 每个跟踪器独立的负责目标跟踪过程中的一个部分, 例如: 利用 DLT^[68] 提取深度特征进行特征提取, 利用 LSST^[69] 估计尺度, 利用 Struct^[70] 和 CSK 作为跟踪模型, 并对跟踪结果进行集成处理, 集成多个跟踪器的优势提高跟踪的稳定性, 但其缺点是实时性差.

4 深度学习的方法

最近, 随着深度学习方法的广泛应用, 人们开始考虑将其应用到目标跟踪中^[71]. 人们开始使用深度特征并取得了很好的效果. 之后, 人们开始考虑用深度学习建立全新的跟踪框架, 进行目标跟踪.

深度学习的效果之所以火热, 是由于通过多层卷积神经网络训练大量的数据. 但是将深度学习用到目标跟踪存在严重的缺陷:

1) 因为在目标跟踪中的正样本只有初始帧中的目标, 没有大量的数据难以训练出性能优良的分类器.

2) 深度学习方法通过将多层网络连接训练分类器来提高跟踪精度, 但是随着卷积层数量的提高以及训练网络的复杂, 算法的实时性很低, 不能满足快速运动目标的实时跟踪.

虽然深度学习存在着缺陷, 但是由于其在结构上的巨大优势, 研究人员逐步将其缺点进行改进. 从改进的方向上看主要可以分成两个部分:

4.1 基于相关滤波方法改进

优良的特征是实现良好跟踪的基础, 深度学习得出的深度特征比传统的手工特征 (HOG、灰度、CN、Harris 等) 更加精确. 相关滤波的核心思想是将计算转化到频域, 相比于传统方法, 跟踪速度得到了大幅度的提升. 为了结合两种方法的优点, 人们开始将深度学习引入到相关滤波中.

HCF^[53] 中将深度特征与相关滤波算法相结合, 由于卷积神经网络不同层提取的特征具有不同的特点. 前层图像分辨率高, 特征包含更多的位置信息; 后层的特征包含更多的语义信息. 作者使用三层网络分别训练相关滤波器, 加权后得到目标的最终响应位置. DeepLMCF 是在 LMCF 的基础上引入了深度特征, 提高了跟踪的精度和算法在复杂情况下的鲁棒性. 由于前层的网络包含更多的位置信息, DeepSRDCF^[62] 在 SRDCF 的基础上建立单层网络, 这样不但减少了深度特征的计算量, 而且极大提升了算法的性能. C-COT^[72] 获得了 VOT2016 竞赛的第一名, 文中提出了一种连续卷积滤波器的方法, 在 SRDCFdecon^[73] 的基础上, 采用多层深度特征, 通过连续空间域差值转换操作, 实现了滤波器的

输入可以是不同分辨率的特征. ECO^[19] 是 C-COT 的改进版本, 从模型大小、样本集大小以及更新策略三个方面进行改进. 首先筛选去除贡献小的滤波器; 然后简化训练样本集, 减少相邻样本间的冗余; 最后每隔 6 帧更新一次模型, 既减少了模型更新的次数, 又能有效地减轻目标遮挡时的影响.

深度学习用于相关滤波方法中不局限于特征提取一个方向, 人们开始考虑用深度学习方法构建分类器. SiamFC^[74] 一种孪生网络的思想: 其主要任务就是学习一个相似匹配函数, 一个分支是保存目标信息, 另一个分支是搜索特征, 然后将两部分特征进行相关操作, 最后置信度高的区域即为目标的位置.

4.2 基于网络结构的改进

DLT^[68] 算法首先将深度学习方法应用到目标跟踪中. 它针对在线训练时正样本缺失的问题, 提出了一种离线训练与在线调整相结合的方法.

作者首先在辅助数据集上进行离线训练, 获取目标的特征表示. 然后在跟踪时, 初始帧给定的情况下对网络结构进行微调, 实现准确地跟踪. 在模型更新时设置阈值机制, 因为过多次数的模型更新会导致算法的复杂度增加, 实时性差, 因此作者通过比较当前帧目标与初始帧目标的相似度是否达到确定的阈值来判断模型是否更新.

虽然 DLT 算法成功的将深度学习应用到了目标跟踪中, 但是仍存在以下缺陷: 数据集上的图片分辨率低, 使得训练的分类器效果差, 很难得到有效的特征. 训练时会存在很多干扰, 比如数据集中图片的目标很有可能成为追踪时的背景. 当发生目标姿态变化, 背景杂乱等干扰时, 会导致目标模型一直更新, 会增加算法的复杂度, 实时性差.

由于训练时存在样本分辨率不高的情况, 对于样本与背景的定义非常模糊, 为了解决这类问题. 文献 [75] 将高斯函数引入到卷积神经网络中训练中, 来提高深度特征提取的精度.

针对目标与背景对比度差异较大的情况, 文献 [76] 中降低了网络的层次, 减少了减法的复杂度. 同时在模型更新时, 由于初始帧是目标的正确描述, 采用与训练相同的网络进行调整; 在后续的跟踪中会存在跟踪漂移等问题, 作者改变了更新策略, 简化了网络结构, 提高算法的实时性^[77].

当目标与背景相似时, 为了提高特征的精确度, 需要采用多层次的神经网络. 文献 [78] 提出一种网状结构的神经网络, 它的思想是利用多个卷积神经网络进行加权处理, 每隔 10 帧更新一个神经网络, 并根据更新时间顺序赋予每个神经网络不同的权重. 这种算法可以有效地提高跟踪的精度, 但是实时性差, 每秒更新 1.5 帧. 文献 [79] 中针对正负样本之间存在严重不平衡分布问题, 用生成式对抗网络来丰富

高质量正样本; 同时提出了一个高阶敏感损失来减小简单负样本对于分类器训练的影响, 跟踪效果超过了 ECO.

5 实验分析讨论

在本节中, 首先对数据集 OTB-2013^[17,80] 及目标跟踪算法的性能评估方法进行介绍; 然后, 在数据集 OTB-2013 上, 对 50 种目标跟踪算法进行比较、分析; 最后对目标跟踪发展趋势进行展望.

5.1 评估方法

为了比较各个算法的性能, 我们使用了三种指标进行度量^[80]. 第一种指标是精准度 (Precision). 在视频序列中跟踪的中心位置与标准中心位置差值小于某一个阈值所占视频总帧数的比例.

$$Precision = \sum_{i=1}^N f \quad (27)$$

$$f = \begin{cases} 1 & CLE \leq d \\ 0 & CLE > d \end{cases} \quad (28)$$

$$CLE = dis(p_1, p_2) \quad (29)$$

其中, N 表示一个视频序列中的帧数, CLE 表示一帧中的中心位置误差, d 表示一个特定阈值, $dis()$ 表示两点之间的欧氏距离, p_1 表示被跟踪目标的中心, p_2 表示实际目标中心.

第二种是成功率 (Success Rate). 将当前帧中跟踪框和标准框的重叠部分的面积与其覆盖总面积做比值, 得到数值 (VOR). 若所得到的数值超过特定的阈值, 我们认为追踪成功. Success Rate 是指计算跟踪成功视频所占总视频的比例. 公式为:

$$VOR = \begin{cases} 1 & k \geq d \\ 0 & k < d \end{cases} \quad (30)$$

$$SuccessRate = \frac{Area(B_T \cap B_G)}{Area(B_T \cup B_G)} \quad (31)$$

公式中, B_T 表示当前帧的跟踪区域; B_G 表示标准的目标区域; \cap 表示两者重叠区域; \cup 表示两者总的覆盖区域.

第三种是跟踪算法的速度, 采用帧/秒 (FPS) 表示.

5.2 实验与讨论

随着深度学习领域的广泛应用, 人们开始将深度特征应用到目标跟踪领域. 之后, 人们开始考虑用深度学习建立全新的跟踪框架进行目标跟踪. 近两年, 基于深度学习的目标跟踪算法层出不穷, 例如 ECO^[19], MDNet^[81], SANet^[82], BranchOut^[83],

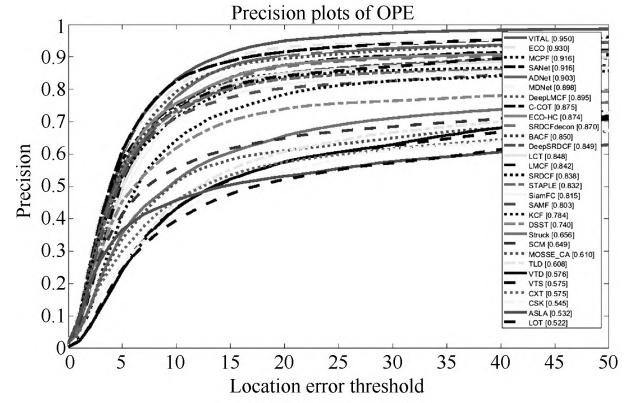
TCNN^[78] 等, 并取得了很好的效果. 本文将相关滤波方法、深度学习方法和传统的目标跟踪算法在 OTB-2013 上进行了对比. 图 9 中给出了 50 种目标跟踪算法在 OTB-2013 中的总体精确度和成功率曲线.

由图 9 的比较结果可以看出, 相比于相关滤波类方法, 深度学习方法的精度和成功率更高. 在数据集 OTB-2013 中, 排名前十的算法有 9 个是基于深度学习类的算法. ECO 算法在各种条件下都有很好的跟踪精度, ECO-HC 中将 ECO 的特征换成传统的 HOG 特征, 提高了处理速度达到 20FPS/s. 相关滤波类算法 SRDCF^[62], LCT^[65], SAMF^[16], KCF^[14] 在精度上排名都比较靠后. 这是由于深度学习方法提取出优秀的特征具有很高的鲁棒性, 深度学习方法训练出的模型更加完善. 传统算法 SCM、TLD、VTD 排名最后, 由此可见, 经典方法已经被远远落后. 但是由表 1 可以看出, 深度学习类算法跟踪速度过慢都多在 0.1~5 帧每秒之间, 不能达到实时跟踪; 而相关滤波类算法跟踪速度可以达到百帧每秒, 可以对目标实时跟踪. 在表 1 中: AUC 表示跟踪的精度, FPS 表示每秒跟踪的帧数.

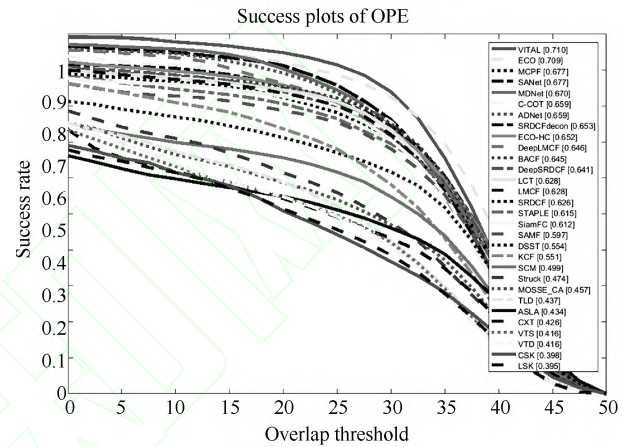
6 结论

通过本文的讨论, 未来的研究方向可以从以下几个方面来进行:

1) 选择有效的特征. 强大的特征是良好跟踪的基础, HOG, CN, CNN 特征已经取得了很好的效果. 如何将优秀的特征进行融合是接下来的发展趋势. 而且, 从实验中可以看出 CNN 的特征优势是明显的, 我们应该研究如何将深度特征运用到所有的算法中.



(a) 精确度对比曲线图
(a) Accuracy comparison curve



(b) 成功率对比曲线图
(b) Success rate comparison curve

图 9 50 种目标跟踪算法在数据集 OTB-2013 上的总体性能对比, 这里只显示了排名前 30 的算法

Fig. 9 The overall performance comparison of 50 object tracking algorithms in the data set OTB-2013, only the top 30 algorithms are shown here

表 1 各种目标跟踪算法的速度比较

Table 1 Speed comparison of various object tracking algorithms

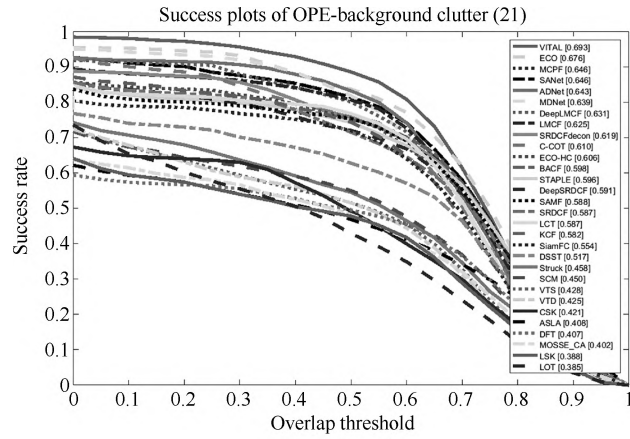
| 基于相关滤波 | AUC | FPS | 基于深度学习 | AUC | FPS |
|------------------------|-------|-----|----------------------------|-------|-----|
| MCPF ^[84] | 0.677 | 0.5 | VITAL ^[79] | 0.710 | 1.5 |
| BACF ^[15] | 0.645 | 35 | ECO ^[19] | 0.709 | 6 |
| LMCF ^[64] | 0.628 | 85 | SANet ^[82] | 0.677 | 1 |
| LCT ^[65] | 0.628 | 27 | MDNet ^[81] | 0.670 | 1 |
| SAMF ^[16] | 0.597 | 7 | C-COT ^[72] | 0.659 | 0.3 |
| DSST ^[51] | 0.554 | 24 | ADNet ^[85] | 0.659 | 3 |
| KCF ^[14] | 0.551 | 172 | HDT ^[86] | 0.654 | 10 |
| CSK ^[13] | 0.398 | 368 | SRDCFdecon ^[73] | 0.653 | 1 |
| MOSSE ^[12] | 0.357 | 669 | CF2 ^[66] | 0.562 | 11 |
| ECO-HC ^[19] | 0.652 | 20 | DeepLMCF ^[64] | 0.646 | 8 |
| | | | DeepSRDCF ^[62] | 0.641 | 0.3 |
| | | | SiamFC ^[74] | 0.612 | 58 |
| | | | DRT ^[63] | 0.581 | 0.4 |

2) 强大的分类器. 在好的跟踪算法中, 都有一个强大的分类器. LMCF 中的结构化 SVM, BACF 中的岭回归以及 ECO 中的 CNN 深度学习架构. 如何选择好的分类器进行结合是我们接下来需要考虑的问题.

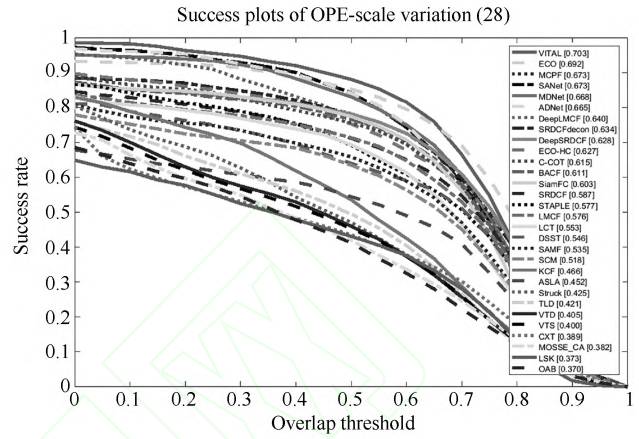
3) 模型的更新. 模型更新一直是目标跟踪中重

要的问题. 好的模型更新机制可以避免将遮挡物以及背景作为目标来更新, 从而有效的解决目标遮挡问题. 因此, 研究出好的模型更新机制势在必行.

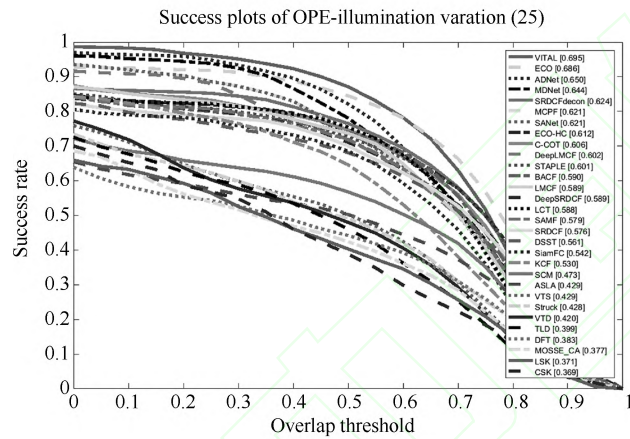
4) 有以上分析可以看出, 相关滤波方法和深度学习学习方法各具特点, 如何能够将二者进行融合, 发挥各自的优势值得我们深入研究.



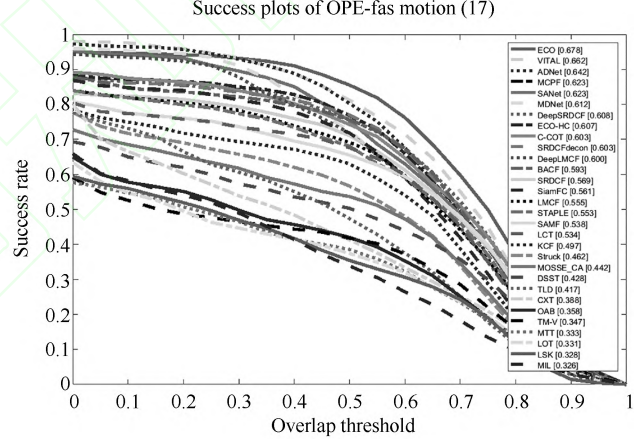
(a) 背景杂乱
(a) Background clutter



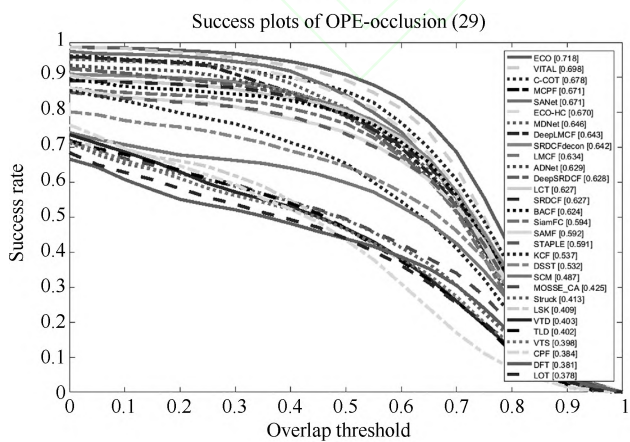
(c) 尺度变化
(c) Scale variation



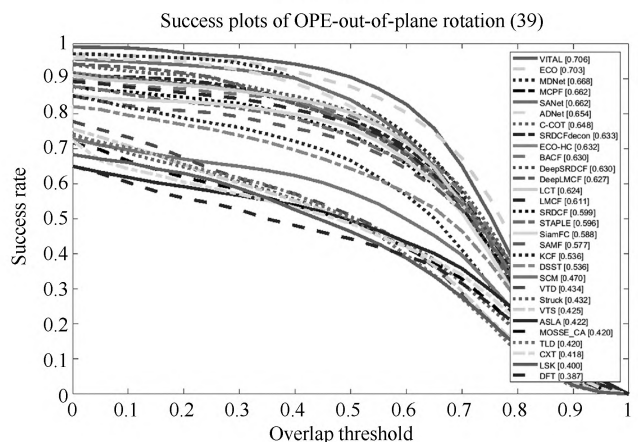
(b) 光照变化
(b) Illumination variation



(d) 快速运动
(d) Fast motion



(e) 遮挡
(e) Occlusion



(f) 平面外旋转
(f) Out-of-plane rotation

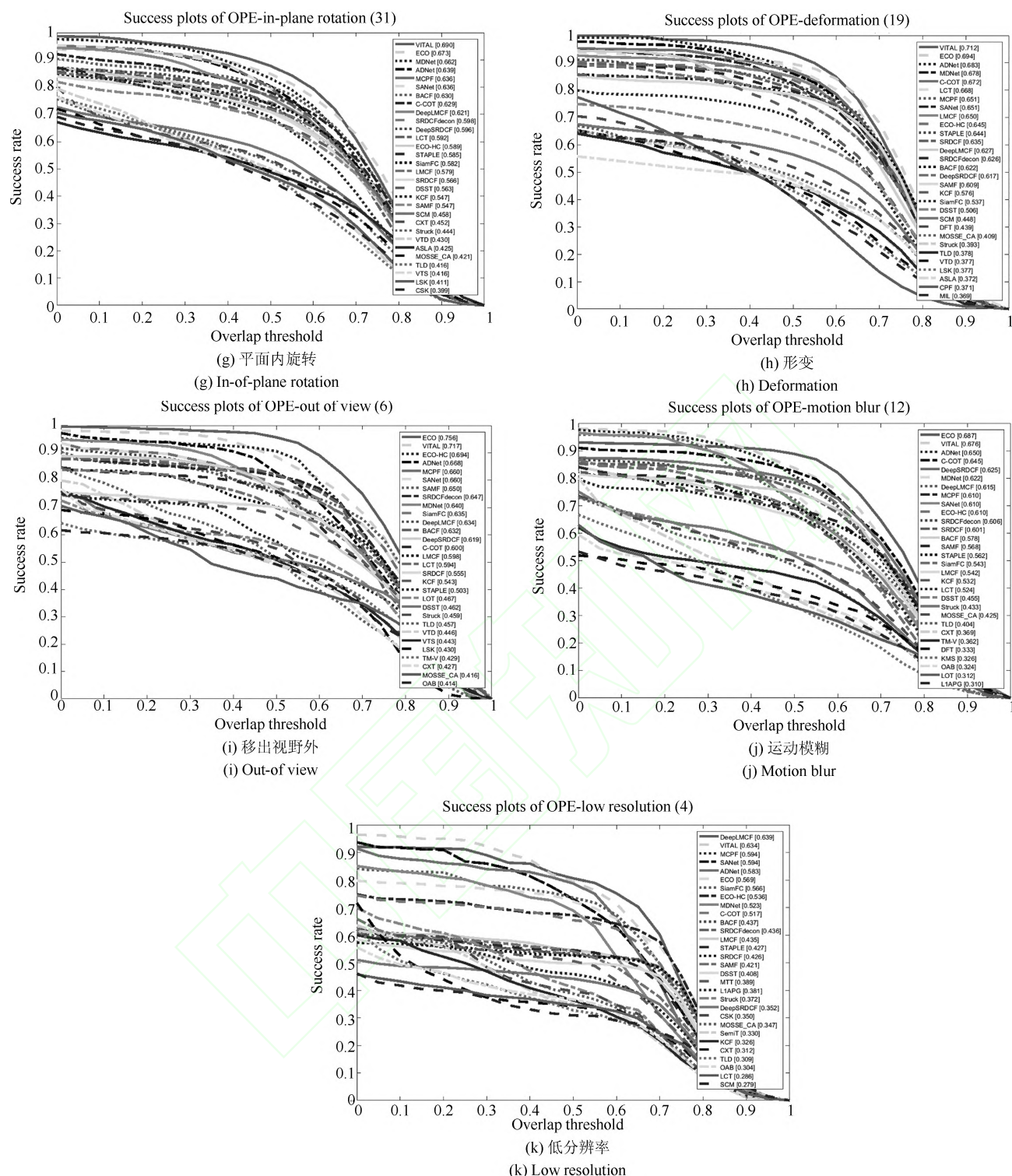


图 10 50 种目标跟踪算法, 在数据集 OTB-2013 中 11 种属性下的成功率曲线

Fig. 10 50 object tracking algorithms, success rate curves under 11 attributes in data set OTB-2013

References

- 1 Luo Jian-Hao, Wu Jian-Xin. A survey on fine-grained image categorization using deep convolutional features. *Acta Automatica Sinica*, 2017, **43**(8): 1306–1318
(罗建豪, 吴建鑫. 基于深度卷积特征的细粒度图像分类研究综述.
- 2 Wang Rui-Xia, Peng Guo-Hua. An image retrieval method with sparse coding based on riemannian manifold. *Acta Automatica Sinica*, 2017, **43**(5): 778–788

自动化学报, 2017, **43**(8): 1306–1318)

- (王瑞霞, 彭国华. 基于黎曼流形稀疏编码的图像检索算法. 自动化学报, 2017, **43**(5): 778–788)
- 3 Yang Sai, Zhao Chun-Xia, Xu Wei. A novel salient object detection method using bag-of-features. *Acta Automatica Sinica*, 2016, **42**(8): 1259–1273
(杨赛, 赵春霞, 徐威. 一种基于词袋模型的新的显著性目标检测方法. 自动化学报, 2016, **42**(8): 1259–1273)
 - 4 Xu M, Jiang L, Ye Z T, Wang Z L. Bottom-up saliency detection with sparse representation of learnt texture atoms. *Pattern Recognition*, 2016, **60**: 348–360
 - 5 Goferman S, Zelnikmanor L, Tal A. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(10): 1915–1926
 - 6 Achanta R, Estrada F, Wils P, Süsstrunk S. Salient region detection and segmentation. In: *Proceedings of Computer Vision Systems, Lecture Notes in Computer Science*, Vol. 5008. Berlin, Heidelberg: Springer, 2008. 66–75
 - 7 Yan Q, Xu L, Shi J P, Jia J Y. Hierarchical saliency detection. In: *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA: IEEE, 2013. 1155–1162
 - 8 Cheng M M, Mitra N J, Huang X L, Torr P H S, Hu S M. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, **37**(3): 569–582
 - 9 Borji A, Itti L. Exploiting local and global patch rarities for saliency detection. In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA: IEEE, 2012. 478–485
 - 10 Achanta R, Hemami S, Estrada F, Süsstrunk S. Frequency-tuned salient region detection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL: IEEE, 2009. 1597–1604
 - 11 Achanta R, Süsstrunk S. Saliency detection using maximum symmetric surround. In: *Proceedings of the 17th IEEE International Conference on Image Processing*. Hong Kong, China: IEEE, 2010. 2653–2656
 - 12 Hou X D, Zhang L Q. Saliency detection: a spectral residual approach. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, Minnesota, USA: IEEE, 2007. 1–8
 - 13 Wang J P, Lu H C, Li X H, Tong N, Liu W. Saliency detection via background and foreground seed selection. *Neurocomputing*, 2015, **152**: 359–368
 - 14 Kim J, Han D, Tai Y W, Kim J. Salient region detection via high-dimensional color transform. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: IEEE, 2014. 883–890
 - 15 Zhang J M, Sclaroff S. Exploiting surroundedness for saliency detection: a boolean map approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, **38**(5): 889–902
 - 16 Itti L, Koch C, Niebur E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, **20**(11): 1254–1259
 - 17 Harel J, Koch C, Perona P. Graph-based visual saliency. In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. Canada: ACM, 2006. 545–552
 - 18 Wei Y C, Wen F, Zhu W J, Sun J. Geodesic saliency using background priors. In: *Proceeding of European Conference on Computer Vision, Lecture Notes in Computer Science*, Vol. 7574. Berlin, Heidelberg: Springer-Verlag, 2012. 29–42
 - 19 Li X, Li Y, Shen C H, Dick A, Van Den Hengel A. Contextual hypergraph modeling for salient object detection. In: *Proceeding of IEEE International Conference on Computer Vision*. Sydney, Australia: IEEE, 2013. 3328–3335
 - 20 Li J, Fang S, Tian Y H, Huang T J, Chen X W. Image saliency estimation via random walk guided by informativeness and latent signal correlations. *Signal Processing: Image Communication*, 2015, **38**: 3–14
 - 21 Yuan Y C, Li C Y, Kim J, Cai W D, Feng D D. Reversion correction and regularized random walk ranking for saliency detection. *IEEE Transactions on Image Processing*, 2018, **27**(3): 1311–1322
 - 22 Yang C, Zhang L H, Lu H C, Ruan X, Yang M H. Saliency detection via graph-based manifold ranking. In: *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*. Portland, USA: IEEE, 2013. 3166–3173
 - 23 Tao D P, Cheng J, Song M L, Lin X. Manifold ranking-based matrix factorization for saliency detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2016, **27**(6): 1122–1134
 - 24 Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(11): 2274–2281
 - 25 Zhao M Y, Jiao L C, Ma W P, Liu H Y, Yang S Y. Classification and saliency detection by semi-supervised low-rank representation. *Pattern Recognition*, 2016, **51**: 281–294
 - 26 Zhu W J, Liang S, Wei Y C, Sun J. Saliency optimization from robust background detection. In: *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, USA: IEEE, 2014. 2814–2821
 - 27 Chen J Z, Ma B P, Cao H, Chen J, Fan Y B, Li R, et al. Updating initial labels from spectral graph by manifold regularization for saliency detection. *Neurocomputing*, 2017, **266**: 79–90
 - 28 Li S Q, Zeng C, Fu Y, Liu S P. Optimizing multi-graph learning based salient object detection. *Signal Processing: Image Communication*, 2017, **55**: 93–105
 - 29 Zhang J X, Ehinger K A, Wei H K, Zhang K J, Yang J Y. A novel graph-based optimization framework for salient object detection. *Pattern Recognition*, 2017, **64**: 39–50

- 30 Lan Y D, Deng H F, Chen T. A new method of distance measure for graph-based semi-supervised learning. In: Processing of International Conference on Machine Learning and Cybernetics. Guilin, China: IEEE, 2011. 1444–1448
- 31 Jiang H Z, Wang J D, Yuan Z J, Liu T, Zheng N N. Automatic salient object segmentation based on context and shape prior. In: Proceedings of the 22nd British Machine Vision Conference. Dundee, Scotland, British: BMVA Press, 2011. 110.1–110.12
- 32 Liu T, Yuan Z J, Sun J, Wang J D, Zheng N N, Tang X O, et al. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, **33**(2): 353–367
- 33 Alpert S, Galun M, Brandt A, Basri R. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(2): 315–327
- 34 Guo Ying-Chun, Yuan Hao-Jie, Wu Peng. Image saliency detection based on local and regional features. *Acta Automatica Sinica*, 2013, **39**(8): 1214–1224
(郭迎春, 袁浩杰, 吴鹏. 基于 Local 特征和 Regional 特征的图像显著性检测. 自动化学报, 2013, **39**(8): 1214–1224)



孟 琬 东北大学信息科学与工程学院副教授. 主要研究方向为人工智能及图像处理. 本文通信作者.

E-mail: menglu@ise.neu.edu.cn

(**MENG Lu** Associate professor at the College of Information Science and Engineering, Northeastern University.

His research interest covers artificial intelligence and image processing. Corresponding author of this paper.)



杨 旭 东北大学信息科学与工程学院硕士研究生. 主要研究方向为图像处理.

E-mail: 13998346746@163.com

(**YANG Xu** Master's degree student in the School of Information Science and Engineering, Northeastern University. The main research direction is image processing.)