

# 基于点轨迹的核相关滤波器跟踪算法

吕韵秋, 刘凯, 程飞

(西安电子科技大学计算机学院, 陕西 西安 710071)

**摘要:** 视频跟踪是计算机视觉领域的重要研究方向之一。然而, 目标跟踪过程中的遮挡问题会导致视频中目标信息的缺失, 因此许多算法无法稳定地跟踪目标。针对遮挡问题, 提出一种基于点轨迹的核相关滤波器跟踪算法。通过对目标及其周围物体长期运动轨迹的估计, 使用谱聚类对点轨迹进行标记, 以区分目标区域与背景区域, 从而判断目标是否发生遮挡或漂移, 决定是否启用重检测机制对目标进行检测。实验证明, 该方法对遮挡和漂移问题具有较强的顽健性。

**关键词:** 核相关滤波器; 点轨迹; 谱聚类

**中图分类号:** TP391

**文献标识码:** A

**doi:** 10.11959/j.issn.1000-436x.2018097

## Kernelized correlation tracking based on point trajectories

LYU Yunqiu, LIU Kai, CHENG Fei

School of Computer Science and Technology, Xidian University, Xi'an 710071, China

**Abstract:** Visual tracking is one of the most important directions in computer vision. However, many state-of-the-art algorithms cannot track the interested object reliably due to occlusion during tracking process, which leads to deficiency of object information. In order to solve occlusion problem, a kernelized correlation tracking method based on point trajectories was proposed. Through analyzing long-term motion cues of the local information, point trajectories were labeled by spectral clustering. These labeled points were used to differentiate the foreground and background objects and thus detect whether the target was occluded or drifts. If drifting and occlusion occur, re-detection was used to detect the re-entering of the target. Experimental results show that the proposed algorithm can handle occlusion and drifting problems effectively.

**Key words:** kernelized correlation filter, point trajectories, spectral clustering

### 1 引言

目标跟踪技术是计算机视觉领域的重要研究方向之一, 其目的是在给定第一帧目标位置的基础上自动预测后续帧中的目标位置。该技术在行为检测、视频监控、智能交通等领域具有广泛的应用。在空间信息网络技术中, 跟踪算法具有重要的作用。近年来, 随着单目标跟踪技术的不断发展, 许多算法已经能够在轻度遮挡和慢速运动的简单场景中对目标进行稳定的跟踪。然而, 由于背景杂乱、

严重遮挡和目标形变的影响, 复杂场景中的目标跟踪仍然是一个复杂且极具挑战性的任务。

跟踪算法通常基于其外观模型分为生成算法和判别算法。生成算法通过对当前帧中目标区域进行建模, 在下一帧中搜索与该外观模型最相似的区域作为预测位置。增量视觉<sup>[1]</sup> (IVT, incremental visual tracking)跟踪方法是经典的生成式算法之一, 该算法利用增广子空间模型减小外观变化的影响。在 $l_1$ 跟踪器中<sup>[2]</sup>, 该算法利用目标模板和碎片模板的稀疏线性组合对目标进行建模。为减少计算量,

收稿日期: 2017-09-12; 修回日期: 2018-04-08

基金项目: 国家自然科学基金资助项目 (No.91538101, No.61571345)

**Foundation Item:** The National Natural Science Foundation of China (No.91538101, No.61571345)

Li 等<sup>[3]</sup>使用正交追踪算法对  $I_t$  跟踪器进行加速。判别算法将跟踪问题转化为一个二分类问题, 以目标区域为正样本, 背景区域为负样本, 使用机器学习方法训练分类器, 在下一帧中使用训练好的分类器找到最优区域。在线多示例学习 (MIL, multiple instance learning) 跟踪方法中, Babenko 等<sup>[4]</sup>提出将多个样本放入正、负样本集中, 使用自提升集成学习方法 (boosting) 进行训练。跟踪—学习—检测<sup>[5]</sup> (TLD, tracking-learning-detection) 跟踪方法将跟踪算法和检测算法结合并通过在线学习机制对二者进行更新。Hare 等<sup>[6]</sup>使用在线结构化输出支持向量机自适应地跟踪目标。

最近, 基于相关滤波器的跟踪方法引起了人们的广泛关注。该方法将卷积运算转换为傅里叶域上的乘法运算, 从而减少了计算量。Bolme 等<sup>[7]</sup>设计了一个最小平方误差和输出自适应相关滤波器, 并在每一帧中用所训练的滤波器对目标外观进行编码。在基于循环矩阵结构 (CSK, circulant structure kernel) 的跟踪算法中, Henriques 等<sup>[8]</sup>对给定的图像块进行循环移位得到训练样本, 并利用这些样本在核空间中训练了一个相关滤波器。核相关滤波器<sup>[9]</sup> (KCF, kernelized correlation filter) 跟踪算法将 CSK 中使用的光线强度特征替换为 HOG 特征, 增强了模型的可靠性。

然而, 一旦目标发生遮挡, 外观模型受到噪声干扰, 大多数算法的跟踪可能会发生漂移甚至丢失目标。针对该问题, 国内外学者提出了很多解决方法。子块匹配法<sup>[10-12]</sup>将目标分成若干个子块分别进行跟踪, 该方法能够对未被遮挡的部分实现有效跟踪, 但是子块的大小和数量会对跟踪效果产生严重影响。轨迹预测法<sup>[13-14]</sup>利用目标位置、速度、加速度等信息预测目标在下一帧的位置。但当目标做无规则运动时, 跟踪效果会出现较大滑坡。贝叶斯预测法<sup>[15]</sup>使用粒子滤波并将跟踪问题转化为一个贝叶斯后验概率最大化问题。但在长时间跟踪中, 随着迭代次数增加, 粒子群很可能出现退化现象, 严重影响跟踪效果。

本文针对目标跟踪过程中的遮挡问题, 提出一种基于点轨迹的核相关滤波跟踪算法。点轨迹是指利用密集光流在视频序列中追踪稀疏采样点形成的运动轨迹, 能够有效地反映物体的运动特征。从长期看, 当遮挡发生时, 目标与遮挡物的运动方向和速度不同。因此, 该算法通过分析

视频中采样点的长期运动轨迹, 将每一帧中的点轨迹标记为目标或背景, 从而预测目标在当前帧的状态并判断目标是否发生了遮挡或漂移。由于 KCF 跟踪算法具有较好的实时性并且能够解决一部分遮挡问题, 因此, 所提算法基于 KCF 算法对遮挡问题进行研究。所提算法主要分为 3 个步骤: 1) 使用 KCF 作为预跟踪器预测目标大致位置; 2) 对预测位置周围区域的点轨迹进行分类, 判断目标框中前景点是否占多数, 如果在目标框内的点大部分为背景点, 则认为目标发生遮挡; 3) 如果目标漂移或被严重遮挡, 使用颜色直方图和巴氏距离重新检测目标位置。实验结果表明, 所提算法能够取得良好的效果, 并且对于跟踪过程中的漂移、遮挡、目标形变等问题具有较强的顽健性。

## 2 KCF 算法

由于 KCF 算法具有较高的效率和准确率, 因此所提算法将其作为预跟踪器, 得到目标的大致位置并确定目标的搜索区域。本节简单介绍了 KCF 跟踪算法, 读者可参考文献<sup>[9]</sup>进一步了解该方法的实现细节。

KCF 算法的主要思路是根据当前帧和之前帧的信息训练出一个滤波器  $w$ , 然后与下一帧进行相关性计算, 将响应值最大的样本作为跟踪结果。在当前帧中, 以目标位置为中心, 取一个训练图像块  $x$ , 其尺寸为  $M \times N$ 。对该图像块进行循环移位得到训练样本  $x_{m,n}, m, n \in \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\}$ 。将训练样本  $x_{m,n}$  与滤波器  $w$  进行相关性计算, 使用岭回归方法最小化相关性计算结果和输出  $y$  的平方误差, 对  $w$  进行训练, 有

$$w = \min_w \sum_{m,n} \left| \langle \phi(x_{m,n}), w \rangle - y(m,n) \right| + \lambda \|w\|_2 \quad (1)$$

其中,  $\lambda$  为归一化参数,  $y(m,n)$  为  $x_{m,n}$  的标签值,  $\phi(\cdot)$  为映射函数, 该函数将样本映射到一个希尔伯特空间中, 使其在该空间内线性可分。 $\phi(\cdot)$  的具体形式未知, 但能够用核函数  $\mathcal{K}$  计算其点积  $\phi'(x)\phi(x') = \mathcal{K}(x, x')$ 。输出  $y$  是服从高斯分布的, 其在训练图像块上取值为 1, 而在其他移位图像块中的取值逐渐减小为 0。

通过定义映射函数  $\phi(\cdot)$ , 滤波器  $w$  可表示为  $w = \sum_{m,n} a(m,n)\phi(x_{m,n})$ , 其中,  $a(m,n)$  为  $x_{m,n}$  的系数,

该系数的解为

$$a = \mathcal{F}^{-1} \left( \frac{\mathcal{F}(y)}{\mathcal{F}(\phi(x) \cdot \phi(x)) + \lambda} \right) \quad (2)$$

其中,  $\mathcal{F}(\cdot)$  和  $\mathcal{F}^{-1}(\cdot)$  分别表示离散傅里叶变换 (DFT) 及其逆变换,  $y$  为标签集合。文献[9]中定理描述如下: 给定循环矩阵  $C(x)$ , 如果一个核函数对于任意排列矩阵  $B$  满足  $\mathcal{K}(x, x') = \mathcal{K}(Bx, Bx')$ , 则其对应的核矩阵是循环矩阵。因此, 如果核函数满足该定理, 式(2)可写为

$$a = \mathcal{F}^{-1} \left( \frac{\mathcal{F}(y)}{\mathcal{F}(k^{xx}) + \lambda} \right) \quad (3)$$

其中,  $k^{xx}$  为核函数  $K = C(k^{xx})$  的第一行。

对于一个新的尺寸为  $M \times N$  图像块  $z$ , 可由式(4)计算其置信图为

$$\hat{f}(z) = \mathcal{F}^{-1} \left( \mathcal{F}(k^{xx}) \odot \mathcal{F}(a) \right) \quad (4)$$

其中,  $\odot$  表示元素与元素之间的乘积,  $k^{xx}$  为  $K = C(k^{xx})$  的第一行。将式(4)中的响应值  $\hat{f}(z)$  变换到空间域中, 则响应值  $f(z)$  最大的样本为 KCF 算法得到的跟踪结果。

### 3 基于点轨迹的核相关滤波器算法

本节主要介绍了基于点轨迹的遮挡处理算法。该算法首先将核相关滤波器跟踪作为预跟踪器得到目标的预测位置, 在该位置周围生成点轨迹; 然后使用谱聚类方法对点轨迹聚类, 对过分割所产生的多余的类进行合并; 最终将点轨迹标记为目标类和背景类。根据预测目标框中 2 类点轨迹的分布情况, 判断目标被遮挡的情况。记录未发生遮挡时的目标样本作为模板, 如果发生遮挡, 取候选样本, 计算候选样本与模板之间的相似度, 将响应值最大的样本作为最终跟踪结果。

#### 3.1 点轨迹生成

通过对视频序列进行分析, 可将遮挡的情况分为以下 2 种: 目标与遮挡物的方向相反; 目标与遮挡物的方向相同但速度不同, 从而相互靠近导致遮挡发生。因此, 从长期角度看, 目标与遮挡物的运动方向或运动速度不同, 通过分析目标及其周围物体像素点的长期运动轨迹, 能够判断目标是否被遮挡。

为了得到物体的长期运动轨迹, 需要对物体中每个像素点的历史运动规律进行分析。点轨迹利用光流方法对第一帧中生成的像素点进行跟踪, 从而

在视频序列中得到一条运动轨迹。点轨迹的质量依赖于光流方法的准确性, 因此, 算法选取效果较好的大位移光流法 (LDOF, large displacement optical flow) [16] 和 Lucas-Kanade 方法 [17] 用于测试视频序列。由于算法的计算复杂度与点轨迹的数量成正比, 则对搜索区域中所有的像素点进行跟踪会降低算法的速度, 为了减少计算量, 点轨迹在以预跟踪器结果为中心的取样区域中进行下采样。取下采样因子为  $c$ , 在原始图像中使用尺寸为  $c \times c$  的窗口进行滑动, 在每个窗口中随机选取一个像素点, 则对于一个尺寸为  $M \times N$  的图像, 样本的数量由  $M \times N$  减少到了  $\frac{M}{c} \times \frac{N}{c}$ 。由于物体外观变化或光流误差等原因, 点轨迹在向后续帧传递的过程中有可能由一个物体漂移到其他物体上, 因此, 需要清除这些点轨迹, 避免一个点轨迹同时具有目标和遮挡物的特征, 从而导致标记错误。在非遮挡情形下, 像素点的前向和反向光流的方向相反, 并且当后一帧的点轨迹反向映射到前一帧时, 应落在前一帧对应像素点的周围区域内。因此, 可以通过定义一致性条件检测点轨迹是否发生漂移, 一致性条件由式(5)和式(6)给出。

$$\omega_t \hat{\omega}_t < 0 \quad (5)$$

$$\|\omega_t + \hat{\omega}_t\|^2 < 0.01(\|\omega_t\|^2 + \|\hat{\omega}_t\|^2) + 0.5 \quad (6)$$

其中,  $\omega_t$  表示点的前向流向量,  $\hat{\omega}_t$  表示后向流向量。式(6)右侧为算法所设置的容忍区间, 允许点轨迹在传递过程中出现一定的误差, 该值的大小与光流大小呈线性关系。如果不满足式(5)和式(6)定义的一致性条件, 则认为点轨迹在该帧中发生了漂移。

前向—反向光流匹配准则示意如图 1 所示, 点  $a$  和点  $b$  表示点轨迹在第  $t$  帧和第  $t+1$  帧的位置, 点  $a'$  表示点  $a$  在第  $t+1$  帧中的映射。箭头①和箭头②分别表示前向光流向量和反向光流向量, 箭头③和箭头④分别表示点轨迹的前向和反向运动过程, 箭头⑤表示漂移情况下点轨迹的反向运动过程。当点轨迹未发生漂移时, 反向光流将第  $t+1$  帧中的点  $b$  映射到灰色区域所表示的容忍区间中, 如箭头④所示; 若发生漂移, 该点被反向光流映射到了图 1 左侧容忍区间外的点  $c$  上, 如箭头⑤所示。由箭头①和箭头②可得, 当点轨迹未发生漂移时, 前向流和反向流向量方向相反。因此, 当反向流终点不在容忍区间内或方向与前向流向量相同时, 点轨迹被终止。

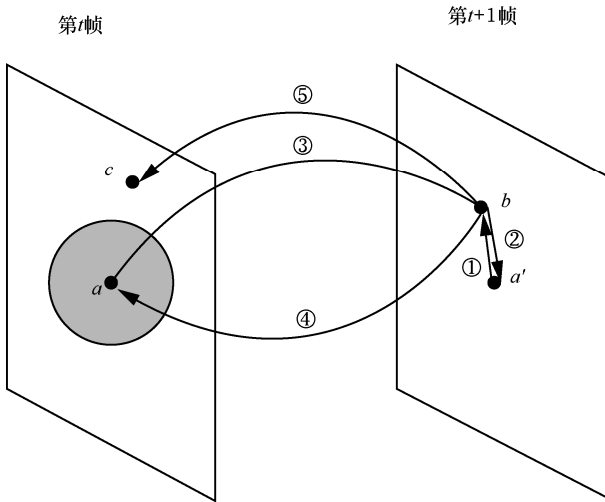


图 1 前向一反向光流匹配准则示意

生成点轨迹的步骤如下：若当前帧为第一帧，以核相关滤波器算法生成的初始目标框为中心，取该目标框 2 倍大小的取样区域，在该取样区域进行下采样得到点轨迹样本；若当前帧不是第一帧，判断点轨迹是否满足一致性条件，剔除不满足式(5)和式(6)的点轨迹，使用与第一帧中相同的方法选择取样区域，保留未发生漂移的点轨迹并在取样区域的空白部分采集新的点轨迹样本。

### 3.2 点轨迹标记

本节采用谱聚类 (spectral clustering)<sup>[18]</sup>的方法对点轨迹进行聚类。谱聚类方法是近年来流行的聚类方法之一，与  $K$  均值 ( $K$ -means) 和混合高斯模型 (GMM, Gaussian mixture model) 等传统方法相比具有更好的聚类效果和较低的计算复杂度。

谱聚类方法是一种基于图论的聚类方法，该方法将每个数据点作为节点，连接节点的线作为边，从而构造完全图。其中，每条边的权重表示数据点之间的相似度，则该完全图的邻接矩阵即数据的相似度矩阵。为了降低对不规则误差数据的敏感度，利用该相似度矩阵的谱（即矩阵的特征值）对数据进行降维，再使用  $K$  均值算法进行聚类。

首先，需要定义一个相似度矩阵。当 2 个点轨迹的运动规律越相同时，表示它们之间的相似度越高。为了更加有效地区分物体，对于轨迹  $i$  和轨迹  $j$ ，应考虑二者之间运动规律相差最大的时刻。因此，轨迹  $i$  和轨迹  $j$  之间的距离定义为

$$d(i, j) = \max_t d_t(i, j) \quad (7)$$

由于相似度需要包含点轨迹的长期运动信息，定义轨迹  $i$  和轨迹  $j$  在  $T$  帧间的平均距离  $v_t^i$  和  $v_t^j$  为

$$v_t^i = \frac{1}{T} (x_{t+T}^i - x_t^i, y_{t+T}^i - y_t^i)'$$

$$v_t^j = \frac{1}{T} (x_{t+T}^j - x_t^j, y_{t+T}^j - y_t^j)' \quad (8)$$

其中， $(x_t^i, y_t^i)$  为第  $t$  帧中的点轨迹位置， $(x_{t+T}^i, y_{t+T}^i)$  为其在第  $t+T$  帧中的位置。通常设定  $T=5$ ，但当 2 个点轨迹重合的帧数小于 5 帧时，则将  $T$  设置为二者的公共帧数。2 个点轨迹之间的相似度通常与它们之间的距离和运动规律有关，因此，定义轨迹  $i$  和轨迹  $j$  之间在第  $t$  帧中的距离为

$$d_t(i, j) = \left\| p_t^i - p_t^j \right\|_2^2 \frac{\left\| v_t^i - v_t^j \right\|_2^2}{\sigma_i^2} \quad (9)$$

其中， $p_t^i$  和  $p_t^j$  分别表示轨迹  $i$  和轨迹  $j$  在第  $t$  帧中的像素点位置， $\left\| p_t^i - p_t^j \right\|_2^2$  为轨迹  $i$  和轨迹  $j$  之间的欧氏距离， $\sigma_i$  为归一化因子，该值由每一帧中的局部光流变化定义<sup>[19]</sup>。

对于轨迹  $i$  和轨迹  $j$ ，二者之间的相似度与它们之间的距离成反比，定义为

$$s(i, j) = \exp(-\alpha d^2(i, j)) \quad (10)$$

其中， $\alpha$  是尺度参数，为了防止式(10)中的相似度变得过小，通常取  $\alpha=0.1$ 。由式(10)得到一个  $n \times n$  的相似性矩阵  $S$ ， $n$  为点轨迹的个数。根据该相似性矩阵  $S$ ，使用谱聚类方法标记生成的点轨迹。

光流向量的灰度表示如图 2 所示，图 2(a)为测试视频序列中的一帧，图 2(b)为该帧中光流向量的灰度表示，其中左边为光流在  $x$  方向上分量的可视化灰度图，右边为在  $y$  方向上分量的可视化灰度图。由图 2(b)可以看出，目标位置与周围物体的灰度值具有明显的对比，说明该位置与周围位置的光流向量具有较为明显的差异。图 2(b)中的横、纵坐标分别表示图中像素点对应位置的横、纵坐标，右侧图例中的坐标表示图中颜色对应的灰度值级别。

### 3.3 合并过分割的类

如果目标与遮挡物上的点轨迹运动差异较大，由 3.2 节中定义的相似度矩阵，根据物体的长期运动特征，谱聚类方法能够成功地将点轨迹分为目标类和背景类。然而，当二者运动差异较小或目标及周围物体的运动情况较为复杂时，可能会造成对点轨迹的过分割。此时，需要对具有相似特征的类进行进一步合并，最终将属于目标的点轨迹合并为一类，属于背景的点轨迹合并为一类。合并过分割类的过程示意如图 3 所示。

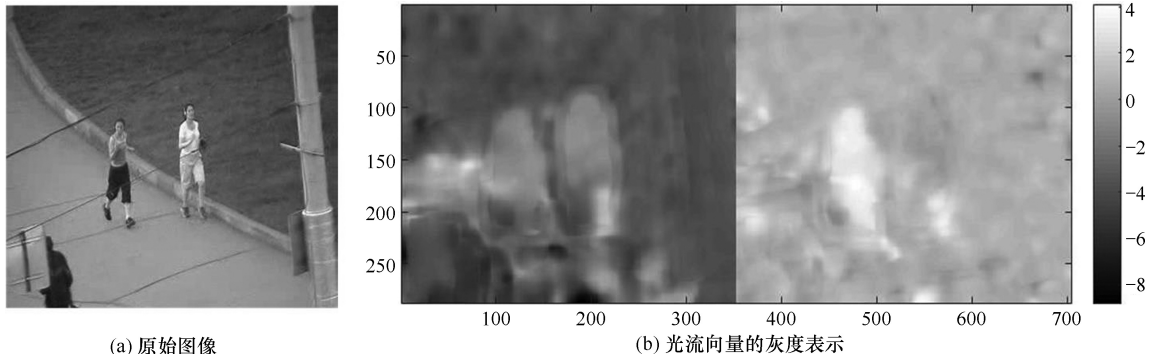


图 2 光流向量的灰度表示

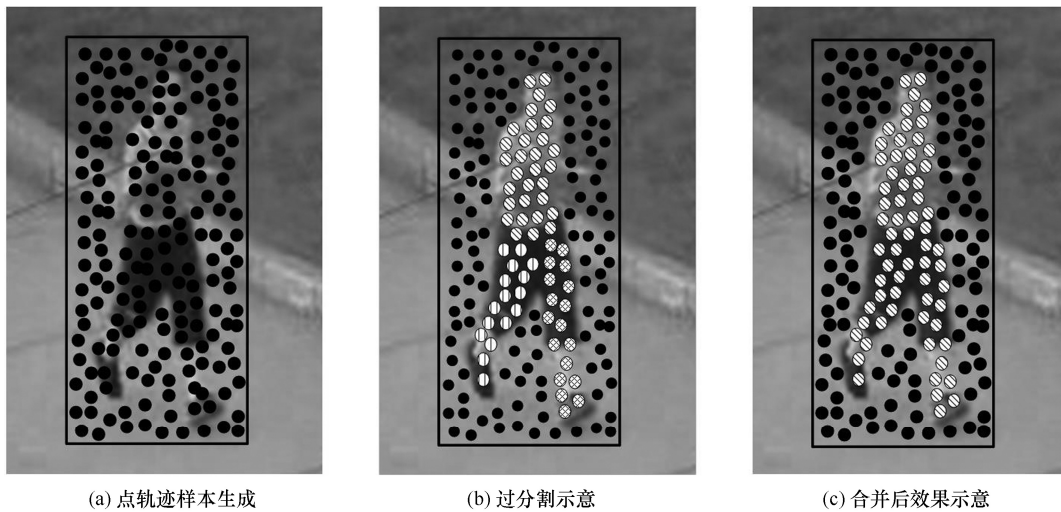


图 3 合并过分割类的过程示意

假设最终的类别数为  $k$ , Ochs 等<sup>[20]</sup>定义了一个包含空间约束项的能量方程, 并且通过最小化该能量方程确定  $k$  的大小. 设  $\lambda_i$  为聚类过程中相似度矩阵的第  $i$  个特征值,  $u_i^A$  是第  $i$  个特征向量  $u_i$  的第  $A$  个元素,  $u^A$  是所有特征向量的第  $A$  个元素组成的向量. 上标  $A$  对应一条单独的点轨迹. 设  $\mathcal{N}(A)$  为点轨迹  $A$  在空间邻域中的点轨迹集合. 初始化聚类个数  $K$ , 最小化式(11)得到聚类结果  $\pi^A \in \{1, \dots, K\}$  为

$$E(\pi, K) = \sum_A \sum_{k=1}^K \delta_{\pi^A, k} \left| u^A - \mu_k \right|_{\lambda}^2 + \nu \sum_A \sum_{B \in \mathcal{N}(A)} \frac{1 - \delta_{\pi^A, \pi^B}}{\left| u^A - u^B \right|} \quad (11)$$

其中, 范数定义为

$$\left| u^A - \mu \right|_{\lambda} = \frac{\sum_i (u_i^A - \mu_i)^2}{\lambda_i} \quad (12)$$

式(11)中, 第一项为一元项,  $\mu_k$  为类别  $k$  的质

心. 第二项为对多余类别的惩罚项.  $\delta_{\pi^A, \pi^B}$  使用克罗内克积定义, 当轨迹  $A$  和  $B$  被归为不同的类时, 该值为 0, 否则, 该值为 1. 因此, 当 2 个轨迹被归为不同类时, 使用类间特征向量差的倒数进行惩罚. 当相似性较大的区域被分割时, 该项的值较大, 而将不连续的区域分割开所产生的惩罚值较小.  $\nu$  为平衡参数, 通常取  $\nu = \frac{1}{2}$ .

最小化式(11)是一个较为复杂的问题, 因为该式有很多局部最小值. Brox 等<sup>[19]</sup>将该问题转化为一个质心未知的多标签马尔可夫随机场 (MRF, Markov random field) 问题, 该问题解决的具体描述可参考文献[20].

### 3.4 算法步骤

本节主要对所提算法进行详细的阐述. 首先, 如果当前帧为视频序列的第一帧, 在初始目标框  $box_1$  周围取采样区域, 并在该采样区域中的像素点进行下采样, 得到初始点轨迹样本; 否则, 将核相

关滤波器跟踪算法作为预跟踪器得到目标的预测边界框  $box_t$ , 利用光流方法得到由前一帧传递的点轨迹, 清除漂移的点轨迹样本并在采样区域内产生新的样本。然后, 根据式(10)定义的相似度矩阵, 利用谱聚类方法对点轨迹进行标记, 将其划分到不同的类别中, 并通过最小化式(11)所定义的能量函数合并过分割的类, 最终实现对目标和背景的标记。根据预测边界框  $box_t$  中目标点轨迹和背景点轨迹的分布情况, 可以判断预跟踪器的结果是否偏移目标, 从而判断遮挡是否发生。当不发生遮挡或目标被轻度遮挡时, 继续使用核相关滤波器进行跟踪; 当目标被严重遮挡时, 利用前一帧结果训练滤波器, 并且不更新预跟踪器参数; 当目标长期发生漂移或被遮挡时, 则启用重检测机制对目标状态进行检测。重检测机制的工作原理描述如下: 记录未遮挡时的目标跟踪框作为模板, 以当前帧的预测边界框  $box_t$  为中心, 取 2 倍大小的区域作为候选样本采样区域, 并在该区域中利用滑动窗口采集候选样本, 利用巴氏距离和颜色直方图计算候选样本与目标模板之间的相似度, 最终将响应值最大的样本作为新的目标跟踪框。本文所提算法流程如下。

**Step1** 在第一帧中, 训练回归方程  $f(z)$ , 记录  $box_1$  为目标模板, 取点轨迹样本。

**Step2** 在第  $t$  帧中, 取点轨迹样本并标记点轨迹, 分别计算目标框中正、负样本数所占比例  $P_{pos}$  和  $P_{neg}$ , 计算目标框  $box_t$  中正样本外接矩形  $P_{rect}$ 。

**Step3** 如果  $P_{neg} < 60\%$ , 则认为不发生遮挡或轻度遮挡, 更新 KCF 参数, 否则, 转到 Step4。如果  $P_{pos} > 70\%$ , 将该帧的结果作为目标模板, 用于重检测。如果  $P_{rect} < 60\%$ , 则认为目标发生漂移, 重新检测目标并更新  $box_t$ 。

**Step4** 如果  $P_{neg} > 60\%$ , 则将目标状态判定为严重遮挡, 不更新 KCF 参数; 若上一帧状态也是严重遮挡, 则对目标进行重检测并更新  $box_t$ 。

**Step5** 若视频未结束,  $t = t + 1$ , 转到 Step2。

## 4 实验分析

本节使用 Wu 等<sup>[21]</sup>提出的流行目标跟踪测试平台 OTB 测试集对算法进行测试, 并且重点分析了算法在遮挡、快速运动和目標形变等复杂场景中的跟踪效果。算法的测试平台为 Windows 7 的 64 位操作系统, 硬件配置为 3.0 GHz 主频的 Intel CPU 以

及 8 GB 的内存。此外, 本节还将本文所提算法与一些跟踪方法进行了比较。

在本文所提算法执行之前, 需要对参数进行设置。为使回归方程更加稳定, 式(1)中的岭回归归一化因子设置为  $\lambda = 10^{-4}$ 。KCF 算法中采样区域的尺寸取目标尺寸的 2.5 倍, 使算法能够取得更多的负样本。为减少计算量, 3.1 节中的下采样因子取为 6, 式(10)中的归一化因子设置为  $\alpha = 0.1$ 。

本文主要采用 3 个标准对跟踪效果进行评价: 1) 中心位置误差, 定义为真实目标框和估计目标框中心的欧氏距离, 图 4 展示了各个算法中心误差的结果; 2) 距离准确度, 定义为估计跟踪结果在给定真实目标框中帧数所占的百分比; 3) 成功率, 定义为  $s = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)}$ , 其中,  $ROI_T$  是估计目标框,

$ROI_G$  是真实目标框。如果该得分大于 0.5, 则认为跟踪成功。准确率与成功率的实验结果如表 1 所示。

本文所提算法在大多数视频序列中取得了较好的效果, 根据上述 3 个标准可以看出, 该算法能够成功地解决遮挡问题。图 5 直观地展示了各个算法在不同视频序列中的跟踪目标框。在 Coke 序列中, 物体被周围叶片遮挡, 而 KCF、DFT 和 TLD 算法的跟踪结果发生了漂移。例如, 在第 255 帧到第 263 帧中, 目标几乎被全部遮挡, 直到 269 帧才完全出现。虽然遮挡时间较长且遮挡程度严重, 本文算法仍可以成功地对目标进行定位。由于在 Coke 序列中所跟踪的目标相对遮挡物运动速度较快, 因此, 通过点轨迹聚类能够成功地将目标从背景中分离, 从而可以准确地判断遮挡是否发生。在 Girl 序列中, 目标为一个女孩的面部, 而在第 412 帧中出现一个另一人物并逐渐靠近目标, 并从 453 帧开始对目标进行了较长时间的遮挡, 直至 467 帧目标才完全出现。于是 KCF 将跟踪目标转移为该人物, 跟踪失败。由中心误差图可以看出, 本文所提算法在 Girl 序列中的中心误差与 KCF、DFT 和 TLD 算法相比较小。由于遮挡物与目标人物的运动规律不同, 所提算法能够准确地将二者区分开。

另外, 该算法也能够很好地解决漂移问题。在 Singer1 序列中, 由于目标外观及尺寸变化, KCF 跟踪的结果发生了严重的漂移。但是, 由图 5 可以看出, 该算法始终能够跟踪目标。由表 1 可知, 该算法的成功率和准确率与其他算法相比是最优的。表 1 中在统计准确率时允许目标位置与真实位置有 20px 的误差。

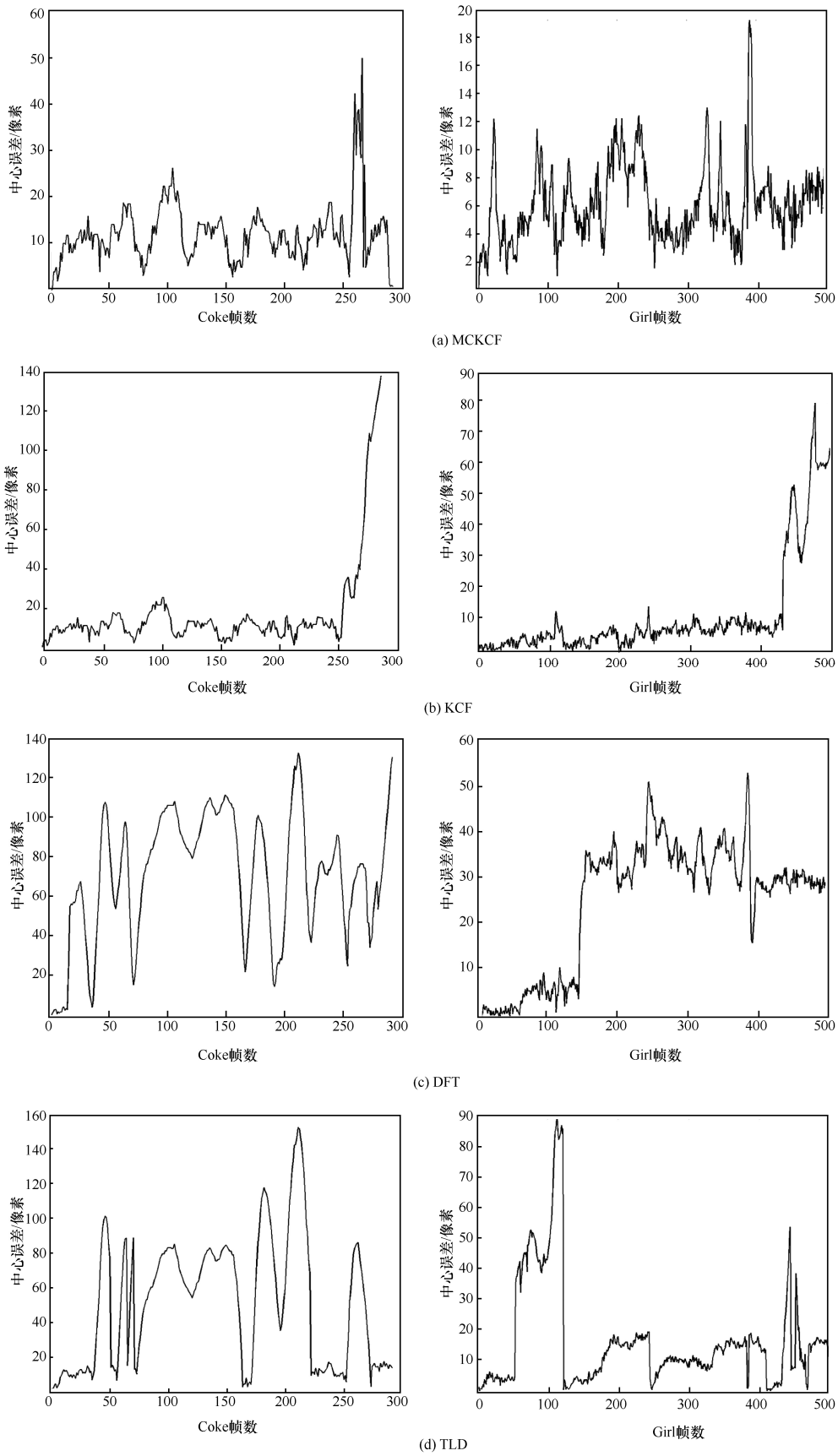


图 4 中心误差曲线

表 1 4 种算法在测试序列上成功率和准确率比较

算法	平均成功率	平均准确率 (20px)
MCKCF	72.96%	82.63%
KCF	65.75%	68.91%
DFT	53.69%	59.63%
TLD	51.47%	52.80%

该算法在空间信息网络中能够进行应用，为了验证其在遥感视频跟踪目标的有效性，本文使用无人机拍摄的场景对目标进行跟踪<sup>[22]</sup>，跟踪效果如图 6 所示。在数据集 Car9 中，第 257 帧到第 278 帧被广告牌遮挡。由图 6 可以看出，本文所提算法能够在遮挡结束后成功定位目标。

以上实验证明了本文所提算法能够有效地解决遮挡、漂移、光照变化、目标形变等问题。但是，根据表 2 可得，该算法与 KCF 算法相比速度

较慢，在后续工作中，可根据文献[23-26]，利用 GPU 对本文所提算法的实时性进行优化。另外，由于点轨迹是基于光流方法选取的，当摄像机抖动剧烈时，本文所提算法无法准确地判断遮挡；当目标窗口尺寸太小时，基于颜色直方图的重检测机制无法有效地检测目标位置。针对这些问题，需要对本文所提算法做进一步的改进。

表 2 2 种算法在测试视频中的帧率 (单位为帧·秒<sup>-1</sup>)

算法	Girl	Coke	Jogging	Walking2	Singer1	平均帧率
KCF	298	192	241	122	116	193.8
MCKCF	33	13	10	25	9	17.8

### 5 结束语

本文针对视频跟踪中的遮挡问题提出了一种创新的跟踪算法。该算法的主要目的是判断跟踪过



图 5 目标跟踪框示意



图 6 算法在遥感视频数据集 Car9 序列中的跟踪结果

程中遮挡问题的发生并在遮挡结束时重新定位目标。首先使用核相关滤波器跟踪作为预跟踪器确定目标的大致位置和搜索区域。然后通过分析该位置周围物体的运动规律,利用点轨迹分割对目标和背景进行标记,并根据标记结果判断是否发生遮挡。当目标受到严重遮挡时,选取与未遮挡目标模板相似度最大的候选样本作为跟踪的最终结果。实验结果证明,所提算法对于遮挡和漂移具有较强的顽健性。

### 参考文献:

- [1] ROSS D A, LIM J, LIN R S, et al. Incremental learning for robust visual tracking[J]. International Journal of Computer Vision, 2008, 77(1): 125-141.
- [2] MEI X, LING H. Robust visual tracking and vehicle classification via sparse representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(11): 2259-2272.
- [3] LI H, SHEN C, SHI Q. Real-time visual tracking using compressive sensing[C]//2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2011: 1305-1312.
- [4] BABENKO B, YANG M H, BELONGIE S. Visual tracking with online multiple instance learning[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2009: 983-990.
- [5] KALAL Z, MIKOLAJCZYK K, MATAS J. Tracking-learning-detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409-1422.
- [6] HARE S, GOLODETZ S, SAFFARI A, et al. Struck: structured output tracking with kernels[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(10): 2096-2109.
- [7] BOLME D S, BEVERIDGE J R, DRAPER B A, et al. Visual object tracking using adaptive correlation filters[C]//2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2010: 2544-2550.
- [8] HENRIQUES J, CASEIRO R, MARTINS P, et al. Exploiting the circulant structure of tracking-by-detection with kernels[C]//European Conference on Computer Vision (ECCV). 2012: 702-715.
- [9] 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.
- [10] SHU G, DEHGHAN A, OREIFEJ O, et al. Part-based multiple-person tracking with partial occlusion handling[C]//2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2012: 1815-1821.
- [11] YANG B, NEVATIA R. Online learned discriminative part-based appearance models for multi-human tracking[C]//European Conference on Computer Vision (ECCV). 2012: 484-498.
- [12] YAO R, SHI Q, SHEN C, et al. Part-based visual tracking with online latent structural learning[C]//2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2013: 2363-2370.
- [13] NGUYEN H T, WORRING M, VAN D B R. Occlusion robust adaptive template tracking[C]//2001 IEEE International Conference on Computer Vision (ICCV). 2001: 678-683.
- [14] WENG S K, KUO C M, TU S K. Video object tracking using adaptive Kalman filter[J]. Journal of Visual Communication and Image Representation, 2006, 17(6): 1190-1208.
- [15] BREITENSTEIN M D, REICHLIN F, LEIBE B, et al. Robust tracking-by-detection using a detector confidence particle filter[C]//2009 IEEE 12th International Conference on Computer Vision (ICCV). 2009: 1515-1522.
- [16] BROX T, MALIK J. Large displacement optical flow: descriptor matching in variational motion estimation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(3): 500-513.
- [17] BAKER S, MATTHEWS I. Lucas-kanade 20 years on: a unifying framework[J]. International Journal of Computer Vision, 2004, 56(3): 221-255.
- [18] VON LUXBURG U. A tutorial on spectral clustering[J]. Statistics and Computing, 2007, 17(4): 395-416.
- [19] BROX T, MALIK J. Object segmentation by long term analysis of point trajectories[C]//European Conference on Computer Vision (ECCV). 2010: 282-295.
- [20] OCHS P, MALIK J, BROX T. Segmentation of moving objects by long term video analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(6): 1187-1200.
- [21] WU Y, LIM J, YANG M H. Online object tracking: a benchmark[C]//2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2013: 2411-2418.
- [22] MUELLER M, SMITH N, GHANEM B. A benchmark and simulator for UAV tracking[C]//European Conference on Computer Vision (ECCV). 2016: 445-461.
- [23] PAUWELS K, VAN HULLE M M. Realtime phase-based optical flow on the GPU[C]//2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2008: 1-8.
- [24] KROEGER T, TIMOFTE R, DAI D, et al. Fast optical flow using dense inverse search[C]//European Conference on Computer Vision (ECCV). 2016: 471-488.
- [25] YAN D, HUANG L, JORDAN M I. Fast approximate spectral clustering[C]//The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2009: 907-916.
- [26] NAUMOV M, MOON T. Parallel spectral graph partitioning[R]. NVIDIA Technical Report, NVR-2016-001, 2016.

### [作者简介]



吕韵秋 (1994-), 女, 江苏徐州人, 西安电子科技大学硕士生, 主要研究方向为目标跟踪、机器学习。



刘凯 (1977-), 男, 陕西西安人, 博士, 西安电子科技大学教授, 主要研究方向为图像编码、FPGA/嵌入式片上系统。

程飞 (1985-), 男, 陕西宝鸡人, 博士, 西安电子科技大学讲师, 主要研究方向为机器学习、目标跟踪。