

基于稀疏子空间选择的在线目标跟踪

黄丹丹¹ 孙怡¹

摘要 本文在粒子滤波框架下提出一种基于稀疏子空间选择的两步在线跟踪方法. 在跟踪的第一步, 利用稀疏子空间选择算法筛选出与目标状态相似性较高的候选区域, 并将目标与背景间的过渡区域定义为单独的类别以降低目标发生漂移的可能; 第二步则通过构建有效的观测模型计算候选区域与目标状态间的相似性, 其中相似性函数综合考虑二者在整体和局部特征上的相似性, 且将目标的原始状态和当前状态都作为参考, 因此增强了观测模型的可靠性; 最后利用最大后验概率估计目标状态. 此外, 该算法通过对目标数据的更新来适应目标的表现变化. 实验结果表明该算法能有效处理目标跟踪中的遮挡、运动模糊、光流与尺度变化等问题, 与当前流行的 9 种跟踪方法在多个测试视频上的对比结果验证了该算法的有效性.

关键词 目标跟踪, 表现变化, 稀疏子空间选择, 粒子滤波

引用格式 黄丹丹, 孙怡. 基于稀疏子空间选择的在线目标跟踪. 自动化学报, 2016, 42(7): 1077–1089

DOI 10.16383/j.aas.2016.c150493

Online Object Tracking via Sparse Subspace Selection

HUANG Dan-Dan¹ SUN Yi¹

Abstract This paper proposes a two-stage online tracking method based on sparse subspace selection in the particle filter framework. At the first stage of the tracking, those candidates that highly similar to the target state are selected using the sparse subspace selection algorithm. The transition region between target and background region is defined as a separate category to alleviate the drifting. And at the second stage, a simple but effective observation model is built to calculate the similarities between the candidates and the target. The similarity measuring function not only considers the integral and local similarities comprehensively but also uses both the original target state and the state after appearance changes as references to enhance the reliability of the measurement model. Finally, the state of target is estimated according to the maximum posterior probability. Furthermore, the target appearance changes are captured by update strategy. The experimental results show that the proposed method can effectively handle the occlusion, motion blur, as well as illumination and scale changes in tracking. Comparative results on challenging benchmark image sequences show that this method performs favorable against the 9 other state-of-the-art algorithms.

Key words Object tracking, appearance changes, sparse subspace selection, particle filter

Citation Huang Dan-Dan, Sun Yi. Online object tracking via sparse subspace selection. *Acta Automatica Sinica*, 2016, 42(7): 1077–1089

目标跟踪是对视频序列中的目标进行状态估计的过程, 它是特征提取、目标检测等底层视觉技术的输出, 同时为行为判定、语义理解等高级视觉分析提供输入信息, 在整个视觉处理流程中起到承上启下的作用. 尽管研究者们提出了大量的跟踪算法^[1–3], 但是由于形变、光照、遮挡等原因导致了目标表现的复杂变化, 使得目标跟踪仍是计算机视觉领域的研究难点.

一般来说, 一个跟踪系统最重要的两个组成部分是运动模型和观测模型. 其中运动模型对目标在当前情况下所有的可能状态进行预测, 常见的运动模型包括卡尔曼滤波^[4]、粒子滤波^[5]等. 观测模

型则用于计算每个候选状态与目标间的相似性, 通常根据目标表现建模的方式确定, 例如基于稀疏表达的跟踪方法常采用重建误差作为度量标准^[6–9], 而基于直方图的建模方法则用直方图相似进行计算^[10–11]. 由于观测模型与目标表现建模密切相关, 因此对目标表现进行精确而鲁棒的描述是成功跟踪的基础. 根据表现建模的方式不同可将现有的跟踪方法分为两类, 分别是生成式方法^[12–15]和判别式方法^[16–19].

生成式跟踪方法首先对目标表现建模, 然后在视频序列中搜索与目标模型最相似的区域来跟踪目标. 增量视觉跟踪 (Incremental visual tracking, IVT)^[12] 基于低维子空间的方法对目标建模, 并通过子空间的增量学习对模型进行自动更新以适应表现变化. 视觉跟踪分解 (Visual tracking decomposition, VTD)^[13] 将运动模型和观测模型分为若干个小模型, 通过这些小模型对目标的表现进行更精确

收稿日期 2015-07-30 录用日期 2016-01-23
Manuscript received July 30, 2015; accepted January 23, 2016
本文责任编辑 刘跃虎
Recommended by Associate Editor LIU Yue-Hu
1. 大连理工大学信息与通信工程学院 大连 116024
1. School of Information and Communication Engineering,
Dalian University of Technology, Dalian 116024

的描述,因此该方法在目标表观发生变化时仍能比较准确地捕捉到目标的位置.此外,文献[14]利用稀疏表示技术对目标区域进行稀疏编码来表征目标,并通过均值漂移搜索具有最小重建误差的区域作为跟踪结果;而超像素跟踪(Superpixel tracking, SPT)^[15]则采用中等视觉线索——超像素来描述目标表观,并定义置信图计算候选区域与目标模型间的相似性.生成式跟踪方法不考虑目标与背景之间的区分性,因此当目标处于复杂背景时,很难保持稳定的跟踪.判别式方法则将目标跟踪建模为二类的分类问题,通过训练目标表观分类器来跟踪目标.典型的判别式跟踪方法如多样例跟踪(Multiple instance learning, MIL)^[16]将目标附近的多个区域定义为正样本包,以此来缓解跟踪中的漂移问题.跟踪学习检测(Tracking learning detection, TLD)^[17]则更关注目标的结构性,它将检测与跟踪相结合,并在训练中根据样本结构性限制不断修正分类器从而得到更好的跟踪结果.文献[18]将图像的稀疏编码作为特征来表征样本,并训练一个线性支持向量机来区分目标与背景.判别式跟踪方法虽然考虑了目标与背景之间的判别性,但是在跟踪中常会由于累积误差而导致模型的漂移.在以上两种方法的基础上,文献[20]构建了一个基于稀疏表示的联合模型,利用生成式方法和判别式方法联合跟踪目标,跟踪的结果则由两种方法共同决定.

上述方法从不同角度对目标表观进行建模,针对跟踪中的难点提出各自的解决方法,但是这些方法都需要对所有的候选区域构建表观模型,然后从其中选择与目标最接近的作为跟踪结果.然而实际上,根据运动模型获得的候选状态中仅有少数与目标状态接近,大部分候选状态都与实际状态具有较大的偏离,因此对所有候选状态进行建模不仅造成大量的冗余计算,而且这些候选区域具有不同的表观特性,这种表观复杂性也加大了跟踪中表观建模和观测建模的难度.针对这一问题,本文在粒子滤波框架下提出一种两步的在线跟踪方法.在跟踪的第一步,该方法首先根据指定的目标区域构造目标数据;然后利用该数据张成的子空间对候选区域进行预处理,提取出少量与目标状态最接近的候选状态,这个预选的过程通过稀疏子空间选择算法实现.在跟踪的第二步,则对这些选出的候选状态构建简单有效的联合观测模型,最后根据最大后验概率估计目标状态.与现有的跟踪方法不同,本文提出的方法对采样得到的候选状态进行预处理,在减少候选状态个数的同时还提高了候选状态的“质量”,即:保留的候选状态比其他状态更接近目标.这使得该算法不必为了区分多样化的候选状态而对目标的表观进行复杂的建模,在避免了冗余计算的同时也使得

模型的设计更简约有效.在多个测试视频上的跟踪结果验证了本文算法的有效性.

1 基于相异性的稀疏子空间选择算法^[21]

稀疏子空间选择是指从大量的源数据中找到一个子空间,该子空间能够保留所有目标数据的关键性质,构成子空间的元素也称为“代表”或“样例”.在视觉系统中,尤其是在视频处理中,稀疏子空间选择算法能够将大量的图像有针对性地进行聚类并获得聚类中心^[22].基于相异性的稀疏子空间选择算法(Dissimilarity-based sparse subset selection, DS3)将该问题描述如下:已知源数据 $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ 和目标数据 $Y = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$, 利用某种度量标准计算 X 、 Y 中元素之间的相异性,根据二者的相异性矩阵计算 X 中数据对 Y 中元素的表达系数.首先介绍源数据 X 和目标数据 Y 之间的相异性矩阵 D , 该矩阵定义如下:

$$D = \begin{bmatrix} \mathbf{d}_1^T \\ \vdots \\ \mathbf{d}_M^T \end{bmatrix} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{M,1} & d_{M,2} & \cdots & d_{M,N} \end{bmatrix} \in \mathbf{R}^{M \times N} \quad (1)$$

式中, M 和 N 分别为源数据和目标数据中元素的个数, $d_{i,j}$ 表示 \mathbf{x}_i 与 \mathbf{y}_j 的相异程度, $d_{i,j}$ 的值越小则说明 \mathbf{x}_i 与 \mathbf{y}_j 越相似,在实际计算中,本文利用 \mathbf{x}_i 与 \mathbf{y}_j 之间的欧氏距离计算二者的相异值,即 $d_{i,j} = \|\mathbf{x}_i - \mathbf{y}_j\|_2$. 假设存在一个概率矩阵 $Z \in \mathbf{R}^{M \times N}$, Z 中的元素 $z_{i,j}$ 与 $d_{i,j}$ 一一对应, $z_{i,j}$ 表示 \mathbf{x}_i 能够代表 \mathbf{y}_j 的概率,将该矩阵记为:

$$Z = \begin{bmatrix} \mathbf{z}_1^T \\ \vdots \\ \mathbf{z}_M^T \end{bmatrix} = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ z_{M,1} & z_{M,2} & \cdots & z_{M,N} \end{bmatrix} \in \mathbf{R}^{M \times N} \quad (2)$$

式中, $z_{i,j} \in [0, 1]$, $\sum_{j=1}^N z_{i,j} = 1$, 矢量 \mathbf{z}_i 为 \mathbf{x}_i 对应的系数矢量.那么稀疏子空间选择可建模为:根据相异矩阵 D 求解概率矩阵 Z ,也就是从源数据 X 中寻找一个包含少量元素的子空间来表达目标数据 Y , 该问题可通过以下优化问题求解:

$$Z^* = \arg \min_{\{\mathbf{z}_i\}, \{e_j\}} \lambda \sum_{i=1}^M I(\|\mathbf{z}_i\|_p) + \sum_{j=1}^N \sum_{i=1}^M d_{i,j} z_{i,j} + \sum_{j=1}^N w_j e_j$$

$$\text{s.t. } \sum_{i=1}^M z_{i,j} + e_j = 1, \forall j; z_{i,j} > 0, \forall i, j; e_j \geq 0, \forall j \quad (3)$$

式中, $\|\cdot\|_p$ 表示 l_p 范数; $I(t)$ 表示指示函数, 当 $t = 0$ 时, 函数值为 0, 否则值为 1. 式 (3) 中, 目标函数的第 1 项用于约束表达 Y 所使用的元素个数, 即寻找的子空间中所包含的元素个数, 第 2 项约束的是 X 表达 Y 所需要的整体代价, 第 3 项约束的是 Y 中奇异元素的个数, 即 Y 中不能用 X 表达的元素的个数. $\lambda > 0$ 为平衡前两项约束的参数, $w_j > 0$ 是惩罚因子, $e_j \in [0, 1]$ 表示 y_j 是奇异元素的概率. 式 (3) 可以通过乘数交替方向算法 (Alternating direction method of multipliers, ADMM)^[23] 求解. 系数矢量 z_i 中只有少量元素非 0, 所以式 (3) 也可视为信号的稀疏表示问题. 由于子空间选择算法的目的就是从源数据中找到能够表达目标数据的子集合, 因此在根据式 (3) 求得优化解 Z^* 之后, 可以通过 $z_{i,j}^*$ 数值的大小来确定子集合中的元素. 如果 X 中的第 i 个元素 x_i 对应的系数矢量 z_i 满足 $\max(z_i) > \mu \|z_i\|_1$, 则 x_i 为子集合中的元素, 其中 μ 为约束系数, 本文中取值为 0.2. 为后文表述方便起见, 将子集合记为 $\{x_{i_1}, \dots, x_{i_c}, \dots, x_{l_c}\}$, 其中 l_c 为元素 x_{l_c} 在源数据集合 X 中的索引, C 为子集合中元素的个数. 从子集合的选择过程可知, C 的值与源数据和目标数据有关, 是根据式 (3) 的计算结果自动确定的. 那么自然地, 目标数据集合中的元素 y_j 可以用子集合中的元素稀疏地表示, y_j 可以通过下式进行聚类,

$$\delta_{y_j} = \arg \max_{i \in \{i_1, \dots, i_c\}} z_{i,j} \quad (4)$$

式中, $\delta_{y_j} \in [1, \dots, K]$ 表示 y_j 的类别, $K \leq C$ 为目标数据的聚类中心个数. 那么源数据集合中的元素 x_i 可以根据其对目标数据中元素的表达能力分配类别, 本文只对子集合中的元素进行分类, 并将

x_{l_c} 的类别定义为系数矢量 z_{l_c} 的最大值所对应的目标数据的类别.

从以上描述可知, 已知源数据 X 和目标数据 Y , 通过 DS3 算法不仅可以找到 X 的一个低维子空间去描述 Y , 还能够根据这种描述关系将源数据自动聚类, 并为目标数据中的每个元素分配类别. 本文将目标跟踪中的候选区域作为源数据 X , 利用目标模板与背景模板构建目标数据 Y , 根据上述 DS3 算法将候选区域聚类, 并为目标数据分配类别, 选出与目标模板同类别的候选区域进行后续跟踪处理, 以减少冗余计算并降低观测建模的复杂性.

2 基于稀疏子空间选择的两步目标跟踪方法

本节着重介绍提出的两步跟踪算法, 首先给出源数据与目标数据的构造方法; 然后介绍如何利用 DS3 算法对候选区域进行预处理; 之后提出本文使用的相似性函数, 最后给出目标数据的在线更新方法.

2.1 源数据与目标数据的构造方法

从上节可知, DS3 算法从源数据张成的空间中寻找一个低维子空间来表达目标数据. 本文将候选区域作为源数据, 具体做法为: 首先利用粒子滤波算法对目标状态进行采样, 然后将采样得到的候选区域进行归一化, 再对归一化后的图像进行列向量化就得到源数据 $X = \{x_1, x_2, \dots, x_M\} \in \mathbf{R}^{d \times m}$. 其中 x_i 是第 i 个候选区域经过列向量化后得到的向量, d 为 x_i 的维数, m 为候选区域的个数. 构造源数据的过程如图 1 所示, 图 1(a) 中矩形框内的图像为采样得到的候选区域, 图 1(b) 是经过归一化的候选图像.

目标数据是待表达的数据, 本文根据指定的目标区域和背景区域进行目标数据的构造. 从现有的跟踪方法可知, 在目标建模时加入判别信息后会得到更好的跟踪结果^[17-18]. 大部分方法中, 判别信息

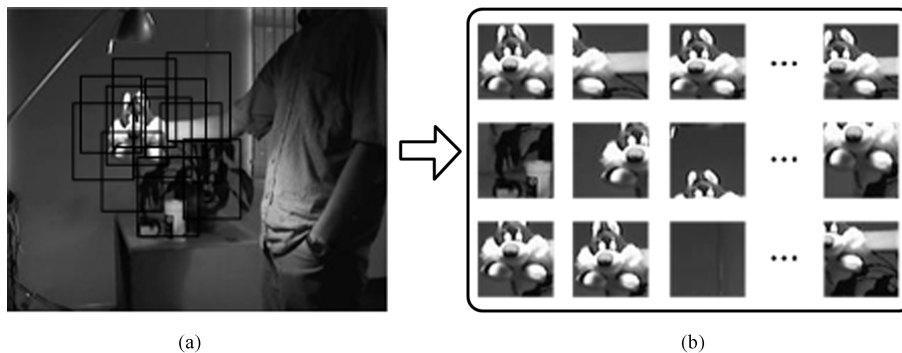


图 1 源数据 (样本) 示意图 ((a) 采样得到的候选区域; (b) 归一化的候选图像)

Fig. 1 Sketch map of source data (samples)

((a) Candidates obtained by sampling; (b) Normalized candidates)

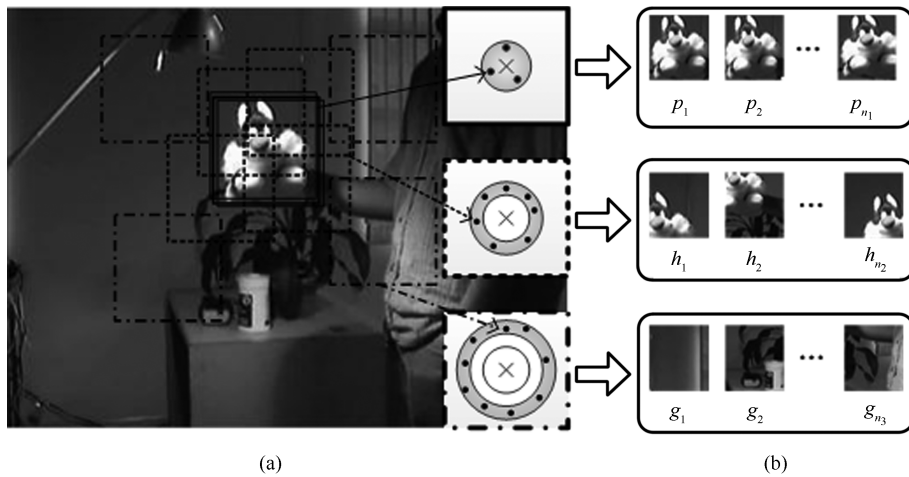


图2 目标数据(模板)示意图((a)模板分布示意图;(b)归一化的模板图像)
Fig.2 Sketch map of target data (templates)

((a) Sketch map of template distributions; (b) Normalized templates)

指的是目标和背景之间的差别信息,例如在基于稀疏表示的方法中,判别信息一般体现为利用目标模板和背景模板构造字典^[18-19].但是在实际中,目标区域与背景区域之间还存在过渡区域,如图2(a)中右侧上数第2个示意图所示.由于目标建模的方法不同,对过渡区域的处理也各不相同.在MIL跟踪中,将这部分区域作为正样本来包容跟踪中产生的模型漂移,而在文献[20]中则将其作为负模板以减少状态估计的误差.在本文提出的跟踪方法中,将过渡区域作为独立的一类,因此目标数据由三部分构成,分别是目标模板 $P = \{p_i | i = 1, \dots, n_1\}$ 、过渡模板 $H = \{h_i | i = 1, \dots, n_2\}$ 和背景模板 $G = \{g_i | i = 1, \dots, n_3\}$.这三类模板的位置分布分别如图2(a)中右侧三个示意图所示,具体的模板图像由左侧的矩形框表示.其中目标模板由目标区域向各个方向分别平移得到,背景模板从背景区域采样得到,过渡模板则在二者之间由采样得到,这三类模板在图2(a)中分别用实线、虚线和点划线的矩形框表示.对这些模板进行归一化和列向量化处理,即可得到由三类模板构成的目标数据 $Y = \{p_1, \dots, p_{n_1}, h_1, \dots, h_{n_2}, g_1, \dots, g_{n_3}\} \in \mathbf{R}^{d \times n}$,其中 $n = n_1 + n_2 + n_3$ 为模板个数的总和,经过归一化的模板如图2(b)所示.

2.2 基于DS3的候选区域预处理

按照上节的方法将源数据 X 和目标数据 Y 构建完成后,根据欧氏距离计算二者之间的相异性矩阵 D ,即 $d_{i,j} = \|\mathbf{x}_i - \mathbf{y}_j\|_2$.通过求解式(3)可找到 X 的一个低维子空间用于描述 Y ,同时源数据 X 可根据表达系数矩阵 Z 自动聚类,目标数据 Y 可根据式(4)进行分类,图3给出了这一过程的示意图.

图3(a)为样本图像,也即源数据 X ;图3(c)则是 X 的聚类结果,其中每个椭圆代表一类,不同线型的椭圆代表不同的类别.图3(b)表示的是目标数据 Y ,图像的外接矩形框给出 Y 中每个模板在图3(c)中所属的类别,例如目标模板 p_1 的外接矩形为实线,那么 p_1 与图3(c)中实线椭圆内的候选区域属于同一类别,其他模板的类别以此类推.从图3可知,经过DS3算法后,目标数据的每个元素都会在候选区域的聚类结果中找到对应的类别.由于目标数据由目标模板、过渡模板和背景模板构成,因此,根据图3所示的分类结果,可以得到与目标模板处于同一类别的候选区域,如图4所示.其中图4(a)为目标数据中的目标模板,它代表了目标在不同时刻的表现;图4(b)为图3(c)中与目标模板处于同一类别的候选区域,也就是经过预处理后保留下来的候选区域.

经过上述过程,候选状态中与目标状态接近的部分被筛选出来并根据观测模型进行最终的状态估计,而其余的则被抛弃.通过对候选区域的预处理,不仅减少了冗余计算,而且保留下来的候选区域的表现更趋于单一化,更有利于目标的状态估计.本文能成功地对候选区域进行预处理,主要有以下三个原因:一是根据稀疏表示系数筛选候选区域能够有效地适应目标的表现变化,由于大量基于稀疏表示的跟踪方法已经证明了稀疏模型能够处理局部遮挡、局部形变以及光流、尺度和位姿变化对目标表现的影响,而本文提出的粒子预判方法又是基于源数据对目标数据的稀疏表示系数,因此继承了基于稀疏表示的模型对目标表现变化的适应能力,从而使粒子预判的结果更加准确.二是采用DS3算法对候选区域进行聚类不需要预先规定类别个数,而是完全根据数据本身的特点自动聚类,避免了由于硬

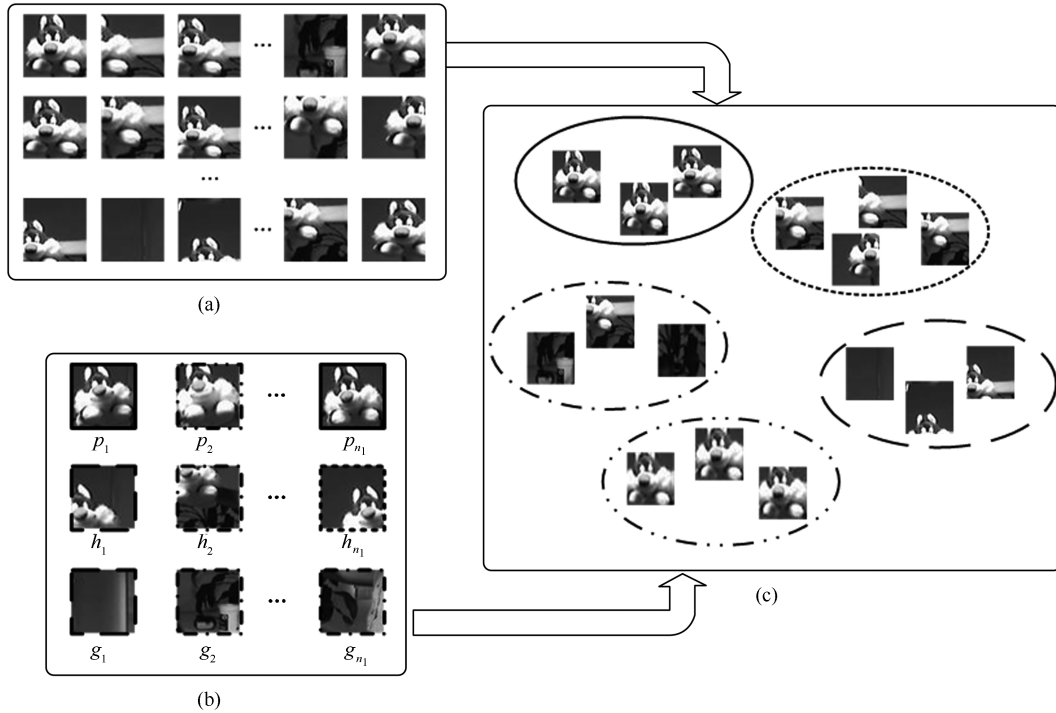


图 3 目标数据 (模板) 和源数据 (样本) 聚类示意图 ((a) 样本示意图; (b) 模板以及聚类结果; (c) 样本聚类结果)
 Fig. 3 Sketch map of target data (templates) clustering and source data (samples) clustering
 ((a) Sketch map of samples; (b) Templates and clustering results; (c) Samples clustering results)

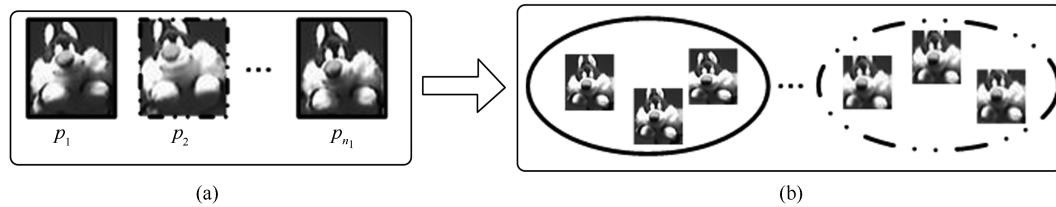


图 4 粒子预判示意图 ((a) 目标模板示意图; (b) 与目标模板属于同一类别的样本示意图)
 Fig. 4 Sketch map of particles pre-filter

((a) Sketch map of target templates; (b) Sketch map of samples that has the same classes with target templates)

性规定聚类个数而导致的聚类误差, 并使得同一类别内的候选区域具有更强的相似性. 三是将过渡模板加入到目标数据中, 使得偏离真实状态较小的候选区域也能被分离出来, 降低了这部分候选区域被保留的可能性, 这也保证了选出的候选区域更接近真实的目标状态.

2.3 相似性函数

在目标跟踪中, 观测模型用于计算候选状态与目标区域之间的相似性, 该相似性反映了候选状态是参考目标的概率. 为了更准确地估计当前的目标状态, 本文构建了一个联合相似性函数, 从整体和局部两个方面综合衡量候选区域与目标之间的相似性.

令 $\{s_t^1, \dots, s_t^L\} \in \mathbf{R}^{d \times L}$ 表示经过预处理后保留下来的候选区域, s_t^i 为第 i 个候选区域, T 为目标模板, 那么 s_t^i 与 T 之间的相似性可通过下式计算:

$$f(s_t^i, T) = f_h \times f_l = \exp(-\alpha \times \theta) \sum_{w=1}^W \sum_{j=1}^J \frac{\min(s^j, q^j)}{q^j} \quad (5)$$

式 (5) 中, $\theta = \arccos(s_t^i, T)$ 为候选区域与目标模板之间的夹角, α 为归一化参数, $f_h = \exp(-\alpha \times \theta)$ 为整体相似性函数. $f_l = \sum_{w=1}^W \sum_{j=1}^J \frac{\min(s^j, q^j)}{q^j}$ 为局部相似性函数, 它将候

选区域和目标区域分成若干个大小相同的子区域, 其中 W 表示子区域的个数, J 为每个子区域的维数, $\mathbf{s} = \mathbf{s}_t^{i,w}$ 表示第 i 个候选区域的第 w 个子区域, $\mathbf{q} = T^w$ 表示目标区域的第 w 个子区域, s^j 与 q^j 分别表示 \mathbf{s} 与 \mathbf{q} 的第 j 个元素. f_h 表示候选区域与目标模板在整体结构上的相似度, f_l 则是利用直方图相交来计算二者间的局部相似. 为了适应目标在跟踪中的表观变化, 本文采用两个目标模板来计算相似性, 分别是首帧指定的目标区域 T_0 和与上一帧的跟踪结果 $\mathbf{s}_{(t-1)}$ 最相似的目标模板 $T^{(t-1)}$, 即:

$$T^{(t-1)} = \arg \min_{i=1, \dots, n_1} (\arccos(\mathbf{p}_i, \mathbf{s}_{(t-1)})) \quad (6)$$

式中, \mathbf{p}_i 为第 i 个目标模板. 则本文提出的相似性函数定义如下:

$$F(\mathbf{s}_t^i, T_0, T^{(t-1)}) = \beta_1 f(\mathbf{s}_t^i, T_0) + \beta_2 f(\mathbf{s}_t^i, T^{(t-1)}) \quad (7)$$

式中, β_1, β_2 为平衡系数, 式 (7) 的第 1 项表示候选状态与原始表观的相似性, 该项可以减轻目标跟踪中由于累积误差而产生的漂移; 第 2 项则表示候选状态与当前表观的相似性, 用于增强观测模型对目标表观变化的适应性. 由于本文的相似性函数同时考虑了整体和局部的相似性, 因此能更全面地衡量候选区域与目标区域的关系; 而同时将两个目标模板作为相似性计算的参照使得该函数能更好地适应表观变化, 从而更准确地估计目标状态.

2.4 目标数据在线更新

目标的表观在跟踪中会由于光流、视角以及本身形变等原因而改变. 随着时间的推移, 根据首帧信息构造的目标模型不能完全适应这些变化, 从而导致跟踪失败. 因此有效的模型更新是跟踪算法能否长期稳定跟踪的一个重要因素, 本文对目标数据进行在线更新以适应目标的表观变化. 由于目标数据是由目标模板、过渡模板和背景模板三部分组成, 因此需要分别进行更新. 为了使更新方法更加高效, 本文每 5 帧进行一次判断, 如果目标模板能够很好地描述目标当前的表观, 则不需要更新. 如果目标模板不能适应目标当前的表观, 则用当前的跟踪结果取代目标模板中权重最小的模板, 权重的计算方法与文献 [6] 相同. 同时在目标区域与背景区域之间进行重新采样以更新过渡模板, 并在背景区域采样更新背景模板. 在目标模板更新时, 保留首帧中由指定的目标区域构成的目标模板以减轻跟踪过程中的漂移. 实验结果证明该更新方法与本文提出的两步跟踪方法相结合, 能够准确地捕捉目标在跟踪过程中的表观变化, 获得更稳定的跟踪结果.

3 两步跟踪算法的实现

本文提出的两步跟踪方法基于粒子滤波框架, 粒子滤波中包含两个重要的模型, 分别是观测模型和运动模型. 令 \mathbf{o}_t 表示目标在第 t 帧的观测, \mathbf{s}_t 表示目标在第 t 帧的状态, 本文将观测模型定义为:

$$P(\mathbf{o}_t | \mathbf{s}_t^i) \propto F(\mathbf{s}_t^i, T_0, T^{(t-1)}) \quad (8)$$

此外, 运动模型建模为具有 6 个独立参数的高斯分布, 即:

$$P(\mathbf{s}_t | \mathbf{s}_{t-1}) = N(\mathbf{s}_t; \mathbf{s}_{t-1}, \Sigma) \quad (9)$$

两式中目标状态矢量定义为 $\mathbf{s}_t = [x_t^c, y_t^c, w_t, h_t, r_t, \varphi_t]$, x_t^c 和 y_t^c 分别是目标中心位置的横、纵坐标, w_t 和 h_t 分别是目标区域在横、纵坐标轴上的尺度, r_t 和 φ_t 分别是目标与横、纵坐标轴之间的角度. $N(\cdot)$ 表示均值为 $\mathbf{s}_{(t-1)}$ 、方差为 Σ 的高斯分布. 方差矩阵 Σ 的形式为对角阵并且对角元素为目标状态矢量对应的变换参数, 这些变换包括目标在中心位置上的平移变换、目标区域在横轴和纵轴上的尺度缩放变换以及目标区域的倾斜和剪切变换. 式 (8) 所示的观测模型反映了采样样本与目标之间的似然概率, 而式 (9) 则描述了目标在相邻两帧之间的状态变化, 即通过式 (9) 和目标在第 $t-1$ 帧的状态 $\mathbf{s}_{(t-1)}$ 可对目标在第 t 帧的状态 \mathbf{s}_t 进行预测, 从而获得目标的候选区域. 那么根据最大后验概率, 第 t 帧的目标状态可由下式估计:

$$\mathbf{s}_t^* = \arg \max_{i=1, \dots, L} \mathbf{s}_t^i \quad (10)$$

在上述框架下, 本文跟踪方法的第 1 步包括根据粒子滤波方法对当前状态进行采样、利用 DS3 算法对候选状态进行预处理. 而第 2 步则包括对保留下来的候选区域构建观测模型, 根据最大后验概率估计目标状态. 为了将本文算法表述的更加清楚, 本节总结了该算法的各个步骤, 并概括如算法 1 所示.

算法 1. 两步跟踪算法流程

初始化阶段 $t = 1$:

在指定目标区域周围提取目标模板, 在背景区域采样得到背景模板, 在目标区域与背景区域之间采样得到过渡模板, 利用三类模板构造目标数据 Y ;

跟踪阶段:

For $t = 2 : \text{FrameNum}$

第 1 步:

1) 根据第 $t-1$ 帧的目标状态, 利用式 (9) 进行状态预测, 并用候选区域构造源数据 X ;

- 2) 利用 DS3 算法对源数据聚类, 并根据式 (4) 对目标数据中的每个模板进行分类;
- 3) 选出与目标模板属于同一类别的候选区域, 进行第 2) 步的处理;

第 2 步:

- 1) 根据式 (5) ~ (7) 计算保留下来的候选区域与目标模板间的相似性;
- 2) 根据式 (8) 和式 (10) 估计当前目标状态;

更新:

If t 是 5 的整数倍
 根据第 2.4 节方法更新目标数据 Y ;

End

End

此外, 值得指出的是, 本文在粒子滤波框架下实现目标的跟踪, 主要是因为粒子滤波是一种比较有效的搜索策略, 它采用多个随机的采样粒子来逼近目标真实的密度分布, 可以不受状态矢量非线性和非高斯的限制. 由于粒子滤波的限制条件少, 并且能够提供目标所有可能的状态, 因此与本文提出的粒子预判方法相结合后, 能够使保留下来的粒子更加接近目标的真实状态, 从而提高跟踪的准确程度. 实际上, 本文提出的粒子预判方法和目标表现模型也可以和其他的搜索方法相结合实现目标跟踪, 如 Kalman 滤波和穷尽搜索等. 但是 Kalman 滤波要求目标的运动模型和观测模型均为线性, 并且要求系统噪声满足高斯分布, 因此当目标不满足上述条件时容易导致跟踪失败. 而穷尽搜索虽然能够提供有效的采样粒子, 但是计算量太大, 无法应用于自然场景下的目标跟踪.

4 实验结果及分析

本节首先对粒子预判算法的有效性进行分析, 然后从定性和定量两个角度对本文算法的跟踪性能进行分析. 本算法由 Matlab 2011a 实现, 并在 Intel Pentium Duo 2.60 GHz 处理器, 内存为 2.0 GB 的计算机上执行, 此外选取 9 种当前跟踪领域内比较流行的跟踪算法在多个测试视频上进行跟踪结果的对比. 为了验证本文算法对跟踪中难点问题的处理效果, 实验中选取的测试视频包含遮挡、运动模糊、光流与尺度变化以及复杂背景等. 由于测试视频中包含彩色视频, 而本文提出的跟踪算法并没有利用图像的颜色信息, 因此将彩色视频中的每帧图像都转化为灰度图进行目标的跟踪. 本文选取的跟踪对比方法包括: IVT^[12]、VTD^[13]、Frag^[10]、MIL^[16]、TLD^[17]、L1^[6]、SCM (Sparsity-based collaborative model)^[20]、DS SM (Discriminative sparse similarity map)^[19] 和 LSK (Local sparse k -selection)^[15]. 所有方法均采用作者提供的程序代码, 并赋给相同的初始状态. 此外, 本文对于模板图像和候选区域的归一化包括两

步: 一是对图像尺寸的归一化, 即利用仿射变换将获得的模板图像和样本图像归一化为相同的尺寸, 本文将该尺寸定义为 32 像素 \times 32 像素; 二是对每个图像像素值的归一化, 即将图像中每个像素的值减去该图像像素值的均值, 再除以该图像的方差. 经过归一化处理的模板图像和样本图像都具有相同的图像尺寸, 并且每幅图像的像素值都满足均值为 0、方差为 1 的概率分布. 本文中其他的参数设置如下: 粒子个数 $m = 300$, 目标模板、过渡模板和背景模板的个数分别为 $n_1 = 10$ 、 $n_2 = 60$ 、 $n_3 = 60$, 相似性函数中的归一化参数 $\alpha = 0.5$, 平衡系数 $\beta_1 = 0.3$, $\beta_2 = 0.7$, 局部相似性函数中子区域的尺寸为 8 像素 \times 8 像素, 子区域的个数 $M = 49$, DS3 算法中的迭代次数设为 50, 参数 $\lambda = 0.05$.

4.1 粒子预判算法的有效性分析

本节选取四个测试视频进行粒子预判算法的有效性分析, 分别是 Faceocc、Singer1、Dollar 和 Owl 序列. 本节的实验包括两部分, 一是分析算法对多余粒子的滤除情况, 二是分析算法对跟踪性能的增强情况. 图 5 是测试视频在粒子预判算法对多余粒子的滤除性能, 其中横轴为初始的粒子数目, 纵轴为经过粒子预判后保留下来的粒子数目, 该数据是测试视频在所有帧的平均值. 从图中可知, 经过粒子预判后, 保留的粒子数目有了大幅度的减少, 证明了粒子预判的有效性. 表 1 是测试视频在粒子预判前后的平均跟踪结果, 包括中心位置误差和重合面积参数, 中心位置误差 PosErr 定义为当前跟踪结果的中心位置坐标与参考值之间的均方根, 面积重合参数定义为 $SC = \frac{R_g \cap R_t}{R_g \cup R_t}$, 其中 R_g 和 R_t 分别表示目标区域面积的参考值和实际跟踪结果. 表 1 中的数据是在初始粒子数目为 300 的条件下得到跟踪结果. 无

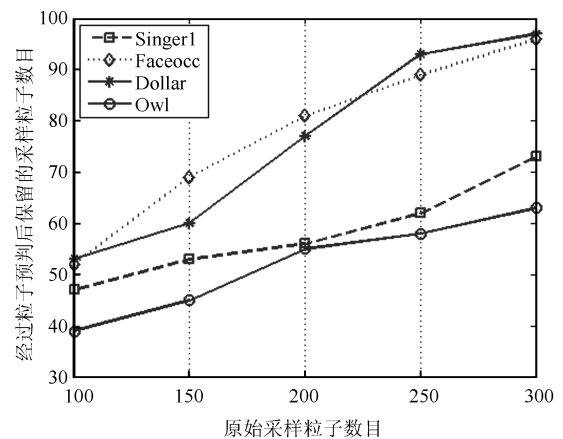


图 5 粒子预判算法对多余粒子的滤除性能

Fig. 5 Performance of the particles pre-filter algorithm for filtering the redundant particles

表 1 有无粒子预判处理的跟踪结果对比

Table 1 Tracking results comparing between the method with particle pre-filter and the method without particle pre-filter

	平均中心位置误差 (pixel)				平均面积重合参数			
	Singer1	Faceocc	Dollar	Owl	Singer1	Faceocc	Dollar	Owl
无预判	6.6	10.2	6.4	120.1	0.67	0.80	0.83	0.06
有预判	3.6	5.1	5.5	28.8	0.85	0.93	0.85	0.47

预判是没有经过粒子预判, 直接对所有粒子都进行观测. 建模的跟踪结果; 有预判则是本文提出的两步跟踪算法的跟踪结果. 从表中可知, 经过粒子预判算法后, 跟踪结果的准确性都得到了不同程度的提高, 这是因为本文提出的粒子预判算法利用稀疏表示来建模目标和样本之间的结构关系, 因此继承了稀疏表示模型对目标表观变化的适应能力, 能够有效地去除距离目标真实位置较远的粒子, 从而提高了算法的跟踪性能.

4.2 定性分析

本节给出本文算法以及其他 9 种对比算法在测试视频上的跟踪结果, 并从四个角度对这些算法的性能进行测试与比较. 为了使图片中的结果更清晰, 本文只给出效果最好的 5 种方法的跟踪结果并只对这 5 种方法的性能进行讨论与分析.

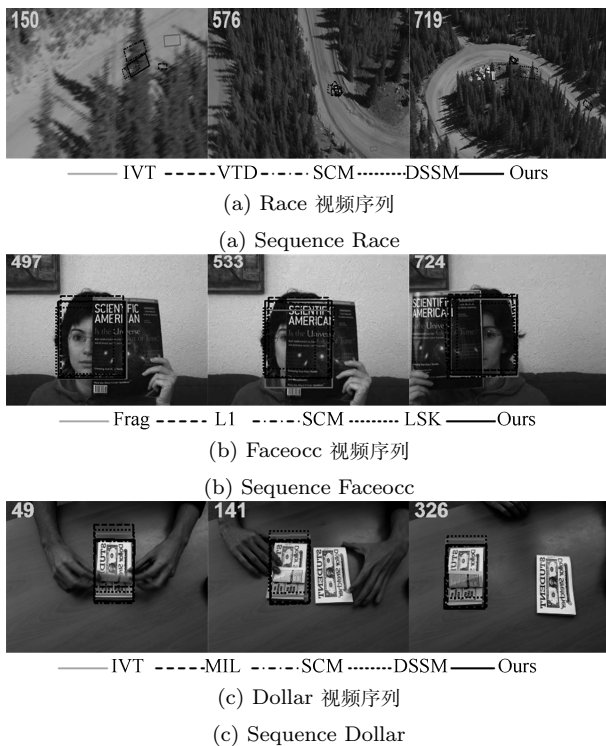


图 6 目标存在遮挡时的跟踪结果

Fig. 6 Tracking results when targets are occluded

测试 1. 目标存在遮挡时本文算法的跟踪性能. 图 6 (a) 所示的 Race 序列中, 目标不仅经历严重的遮挡而且目标的尺度和外观都发生了巨大的变化, 从结果图像中可知, 只有本文的算法能够在整个序列中保持稳定准确的跟踪. 在 150 帧, 当目标几乎被完全遮挡时, 其他算法都不能对目标进行准确定位. 而在 576 帧, 尽管除了 IVT 外的方法都捕捉到目标的位置, 但是只有本文算法能够对目标的尺度进行正确的估计. 随着时间推移, 到了 719 帧, 除本文算法外的其余算法都已经丢失了目标, 跟踪失败. 图 6 (b) 所示的 Faceocc 序列中, 人脸图像被频繁地遮挡, 尽管图中所示的 5 种方法都能成功跟踪目标, 但是本文的算法在尺度以及定位精度上要优于其他方法, 具体数值如表 3 所示. 图 6 (c) 的 Dollar 序列中, 目标纸币经过折叠后, 从目标呈现的表现来看相当于该纸币被部分遮挡. IVT、MIL 和 DSSM 在纸币被折起时, 就产生了漂移, 并且除 MIL 外其他方法并没有在后续图像中恢复正确的跟踪. 而本文的算法在根据稀疏子空间选择算法对候选区域进行预处理时, 将目标区域与背景区域间的过渡区域单独定义为一类, 因此减少了这部分区域被保留下来参与最终状态估计的可能性, 避免了漂移的发生.

测试 2. 目标由于快速运动而产生运动模糊时本文算法的跟踪性能. Jumping 序列中的目标由于快速跳跃而产生模糊, 而背景区域的复杂性也增加了跟踪的难度. IVT 和 Frag 在构建目标模型时不考虑目标与背景间的区别信息, 同时根据模糊目标图像构建的表观模型又不能准确地描述目标, 从而导致跟踪失败. DSSM 算法虽然利用了判别信息, 但是对过渡区域所属类别的定义并不明确, 因此当目标区域存在模糊时, 极易导致模型的漂移. 如图 7 (a) 所示, DSSM 的跟踪结果总是在实际位置周围, 这说明该方法构建的目标模型虽然能在一定程度上区分目标与背景, 但是模型已经产生漂移以至于不能准确地定位目标. 此外 SCM 算法则由于其采用的联合模型而使得跟踪比较准确. 相对于以上几种方法, 本文算法在跟踪的第 1 步利用了目标与背景间的判别信息, 而且将过渡区域明确规定为一个独立的类别, 在减少漂移的同时也保证了筛选出的候

选区域更接近目标的真实状态; 此外, 算法第 2 步建立的观测模型不仅从整体与局部两个方面综合衡量候选状态与目标之间的相似性, 而且采用原始目标状态和经过表观变化的目标状态作为相似性函数的参考标准, 因此本文的观测模型在存在模糊时仍能对目标当前的状态进行准确地估计. 图 7 (b) 所示的 Owl 序列中, 目标区域的准确轮廓在快速的运动下几乎不能用肉眼分辨. 从结果图像可见, 除本文算法外, 其余几种方法都有不同程度的漂移, 这说明本文算法能够有效地处理跟踪中由于快速运动而产生的模糊现象.

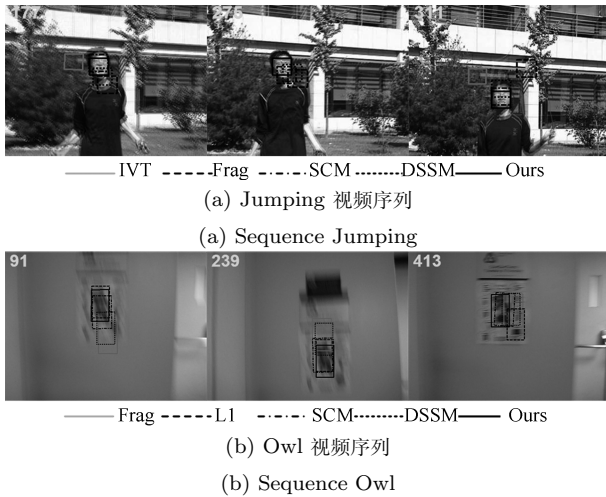


图 7 目标快速运动产生模糊时的跟踪结果
Fig. 7 Tracking results when targets appearance is blurry because of quick movement

测试 3. 光流变化时本文算法的跟踪性能. 图 8 (a) 所示的 Singer1 序列中, 场景的光流与目标的尺度都发生巨大的变化. L1 和 LSK 不考虑目标与背景间的判别信息, 因此不能很好地应对光流和尺度的变化. VTD、SCM 和本文算法都能在整个序列中成功地跟踪目标, 但是本文算法采用两步跟踪, 不仅能够减少漂移, 而且由于观测模型的有效性而使跟踪的结果更加准确. 在 Car11 序列中, 目标车辆在低照度情况下行驶, 并且在运动中伴随着光流的变化以及由于相机抖动引起的轻微模糊. 由于目标的表观在整个序列中只发生了刚性变化, 因此大多数方法都能很好地跟踪目标. 只有 TLD 方法从跟踪开始不久就产生漂移, 并且随着时间推移, 这种漂移越来越严重, 如图 8 (b) 所示.

测试 4. 目标表观发生变化或场景中不存在与目标相似区域时本文算法的跟踪性能. 在图 9 (a) 所示的 Sylv 序列中, 目标在不同光照下由于位姿变化而导致目标表观的变化. 虽然这几种方法都成功地对目标进行了跟踪, 但是 DSSM 和 TLD 算法在 355 帧到 542 帧之间发生了轻微的漂移, 而 SCM 在 613 帧也偏离了目标的真实位置, 本文算法一直保持着

稳定的跟踪, 并且取得了最好的跟踪结果, 如表 2 和表 3 所示. 图 9 (b) 所示的 Football 序列中, 目标经历快速运动并且场景中不存在与目标相似的区域. 从结果图像可见, IVT 与 Frag 在 181 帧开始产生偏移, 在 309 帧, VTD 与 DSSM 也产生漂移, 只有本

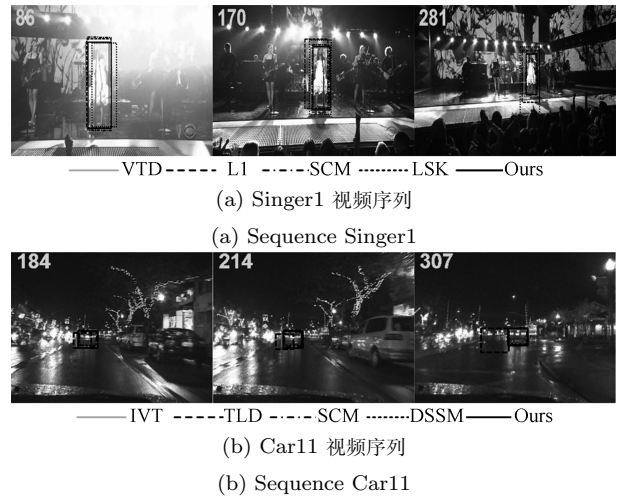


图 8 光流发生变化时的跟踪结果
Fig. 8 Tracking results when targets undergo illumination changes

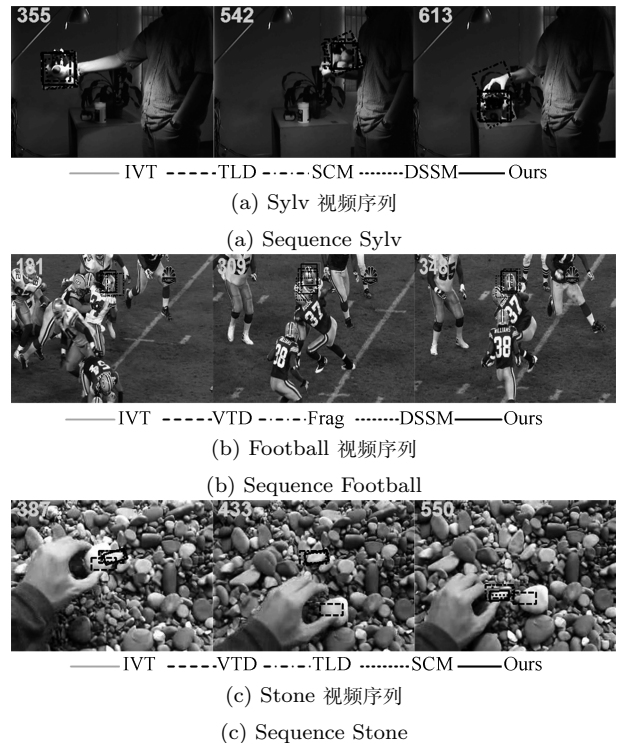


图 9 目标发生形变 ((a), (b)) 以及场景中不存在相似区域 ((b), (c)) 时的跟踪结果
Fig. 9 Tracking results when targets occur deforms ((a) and (b)) and results when there are similar regions in scenes ((b) and (c))

表 2 跟踪结果的平均中心位置误差 (像素)
Table 2 Average center location errors of tracking results (pixel)

视频序列	IVT	VTD	Frag	MIL	TLD	L1	SCM	DSSM	LSK	Ours
Car11	2.2	30.1	40.1	44.7	17.6	33.7	2.1	<i>2.0</i>	73.3	1.9
Faceocc	11.2	9.5	5.7	18.6	16.0	6.5	3.9	42.7	7.7	<i>5.1</i>
Sylv	6.0	21.6	7.5	7.0	5.6	22.0	5.5	<i>5.1</i>	59.9	5.0
Dollar	7.0	65.1	71.0	6.6	11.8	68.9	<i>6.1</i>	11.0	47.2	5.5
Football	6.9	<i>6.6</i>	7.9	209.4	13.0	58.4	26.4	11.7	15.9	6.1
Jumping	34.8	111.9	21.2	41.8	–	38.0	<i>6.1</i>	12.7	63.5	3.9
Owl	114.7	190.7	40.6	292.1	–	33.1	<i>32.2</i>	50.7	230.7	28.8
Stone	2.4	26.7	67.4	31.0	5.6	32.6	3.8	43.9	105.4	<i>2.9</i>
Race	176.4	82.2	221.4	–	–	214.7	28.7	<i>4.3</i>	217.2	3.9
Singer1	9.1	<i>3.7</i>	42.1	241.0	27.5	5.6	<i>3.7</i>	11.9	7.7	3.6

表 3 跟踪结果的平均面积重合误差 (像素)
Table 3 Average overlap scores of tracking results (pixel)

视频序列	IVT	VTD	Frag	MIL	TLD	L1	SCM	DSSM	LSK	Ours
Car11	0.80	0.38	0.06	0.17	0.36	0.46	<i>0.81</i>	<i>0.81</i>	0.0	0.82
Faceocc	0.81	0.84	0.89	0.73	0.67	0.87	<i>0.91</i>	0.48	0.82	0.93
Sylv	0.72	0.57	0.69	<i>0.73</i>	0.72	0.48	0.72	<i>0.73</i>	0.35	0.75
Dollar	0.81	0.36	0.29	0.80	0.64	0.31	<i>0.83</i>	0.75	0.37	0.85
Football	0.72	<i>0.73</i>	0.71	0.06	0.60	0.32	0.40	0.61	0.41	0.74
Jumping	0.24	0.15	0.34	0.22	–	0.20	<i>0.63</i>	0.43	0.17	0.73
Owl	0.15	0.06	0.40	0.08	–	<i>0.46</i>	0.47	0.28	0.05	0.47
Stone	<i>0.56</i>	0.40	0.18	0.36	0.48	0.42	<i>0.56</i>	0.13	0.07	0.65
Race	0.02	0.30	0.05	–	–	0.05	0.51	<i>0.55</i>	0.01	0.64
Singer1	0.50	<i>0.82</i>	0.30	0.32	0.35	0.66	0.85	0.72	0.62	0.85

文算法在整个序列中都能正确地估计目标的状态,从而保持着准确的跟踪结果. 在 Stone 序列的背景中存在大量与目标类似的区域, 这些区域增大了发生错误跟踪的可能性. 在 387 帧, 当目标被部分遮挡时, VTD 将遮挡物误认为是目标, 并在后续跟踪中无法恢复. 而 IVT、TLD 与 SCM 在跟踪中不能很好地处理目标的尺度变化, 对比而言, 本文算法的跟踪结果取得了较小的定位误差, 同时与参考目标具有最大的重合面积.

4.3 定量分析

本节采用中心位置误差 PosErr 和面积重合参数 SC 对所有跟踪算法的性能进行定量分析. 表 2、表 3 分别给出包括本文算法在内的 10 种跟踪方法在 10 个测试视频上的跟踪结果, 分别为跟踪结果的平均中心位置误差和平均面积重合参数, 其中最优秀以及次优的跟踪结果分别用粗体和斜体表示, “–”

表示跟踪出现中断. 从表 2 可知, 本文提出的两步跟踪算法在这些测试视频上均取得最优或次优的结果, 而在表 3 所示的重合面积参数中, 本文算法在跟踪的准确性上优于其他所有的跟踪方法.

为了更直观地表现所有对比方法的跟踪结果, 图 10 给出了上述两个对比结果的曲线图, 点划线为本文算法的跟踪结果. 图 10(a) 中的中心误差越小越好, 而图 10(b) 中的重合参数越大越好. 图中横轴测试视频的序号代表的视频序列分别为: 1. Car11; 2. Faceocc; 3. Sylv; 4. Dollar; 5. Football; 6. Jumping; 7. Owl; 8. Stone; 9. Race; 10. Singer1. 从图 10 中可知本文算法的跟踪性能优于其他方法.

除了上述平均跟踪结果, 本文还给出了 Benchmark^[2] 中用于衡量算法整体跟踪性能的曲线: OPE (One-pass evaluation) 的曲线下面积 (Area under curve, AUC), 如图 11 所示. 其中图 11(a) 为精确度曲线, 图 11(b) 为跟踪成功率曲线.

从图 11 可知, 当误差阈值设为 20 个像素时, 本文算法的跟踪精确度达到 90.25%, 而当成功跟踪的阈值设为重合面积之比为 0.6 时, 本文算法的成功跟踪率达到 75.38%, 这充分说明本文算法在整个跟踪过程中均能够保持比较准确而稳定的跟踪。

4.4 算法的运行时间分析

本节对本文算法的运行时间进行分析. 由于 LSK 方法要求运行环境为 64 位的处理器, 无法与其他跟踪方法在同一实验平台上实现, 因此本节只

对包括本文算法在内的 9 种跟踪方法在相同实验平台上的运行时间进行对比. 表 4 给出了这些跟踪方法的运行时间, 从表中的数据可知, 本文提出的两步跟踪算法虽然未能达到实时的跟踪, 但是其运行时间在表中的跟踪方法中仍然处于中上水平. 此外, 本文算法采用文献 [21] 的作者提供的代码进行粒子预判的计算, 该代码并没有经过优化或并行处理, 是本文算法中最耗时的部分. 在测试过程中, 大概有 3/4 的运行时间耗费在粒子预判阶段, 因此本文算法的计算效率仍有提升的空间。

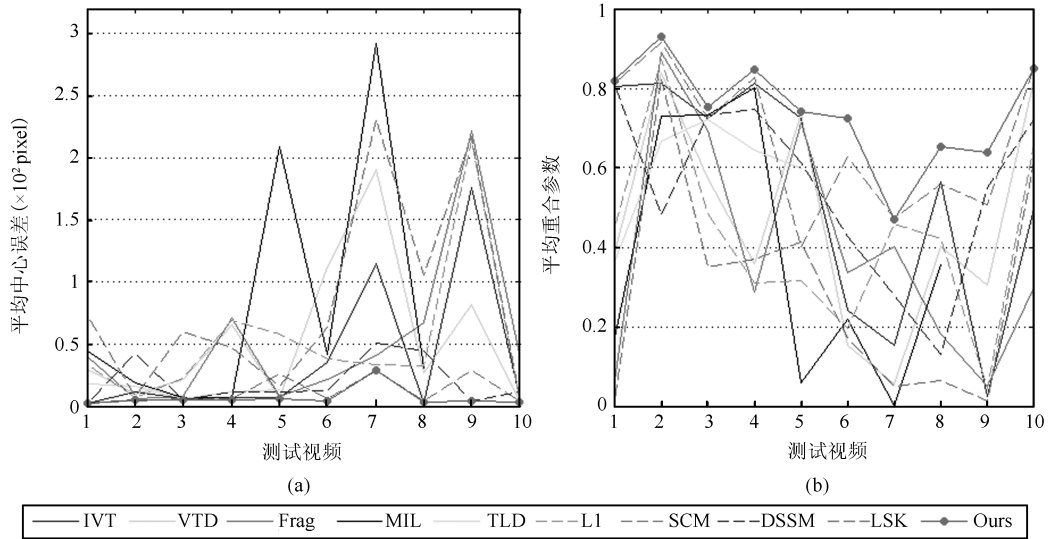


图 10 所有跟踪方法在全部测试视频上的跟踪性能 ((a) 平均中心误差曲线; (b) 平均重合面积参数曲线)

Fig. 10 Performance of all the tracking methods in test sequences ((a) Average center error curve; (b) Average overlap area parametric curve)

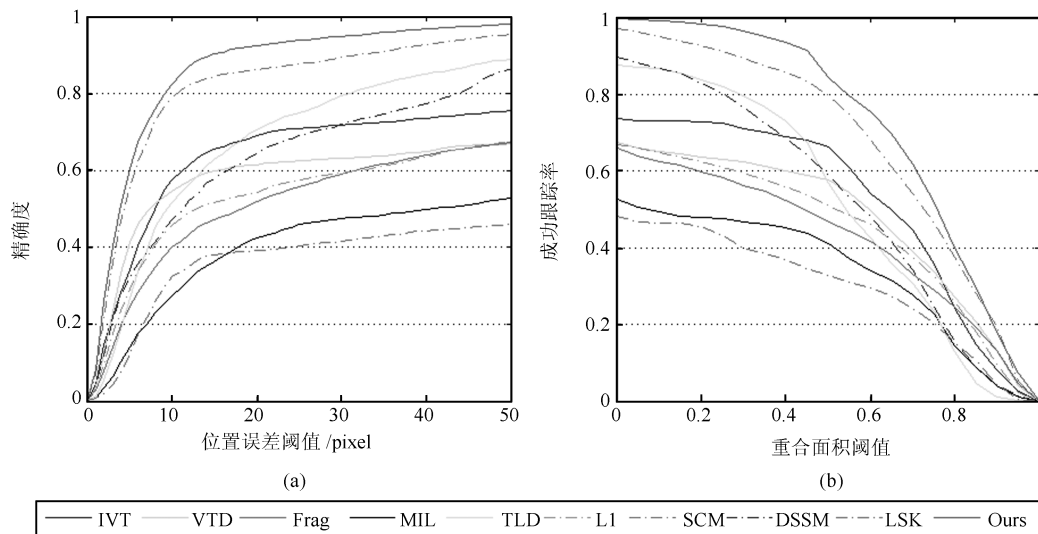


图 11 OPE 曲线 ((a) 跟踪精度的统计曲线; (b) 跟踪成功率的统计曲线)

Fig. 11 One-pass evaluation curve ((a) Precision plot of OPE; (b) Success plots of OPE)

表 4 本文跟踪算法与对比跟踪算法的平均运行时间比较

Table 4 Average running times comparing between the proposed method and other methods

跟踪方法	IVT	VTD	Frag	MIL	TLD	L1	SCM	DSSM	LSK	Ours
运行时间 (s/f)	0.12	6.74	1.28	1.3	0.34	5.25	3.33	1.82	–	0.86

5 总结

本文在粒子滤波框架下提出一种基于稀疏子空间选择的两步跟踪算法. 该算法对采样粒子进行预处理, 减少候选区域个数的同时提高了剩余候选状态与目标真实状态间的近似程度, 从而减少冗余计算并降低了目标建模的复杂度. 定义的相似性函数综合考虑候选区域与目标状态在整体与局部特征上的相似性, 并且将目标的原始状态和经过表观变化后的状态同时作为参考, 因此能够提供更为准确的观测模型, 提高目标估计的准确性. 此外, 该算法采用的更新策略也增强了其对目标表观变化的适应能力. 与其他跟踪方法的对比实验结果证明本文算法的有效性.

References

- 1 Yilmaz A, Javed O, Shah M. Object tracking: a survey. *ACM Computing Surveys (CSUR)*, 2006, **38**(4): 1–45
- 2 Wu Y, Lim J, Yang M H. Online object tracking: a benchmark. In: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, USA: IEEE, 2013. 2411–2418
- 3 Smeulders A W M, Chu D M, Cucchiara R, Calderara S, Dehghan A, Shah M. Visual tracking: an experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014, **36**(7): 1442–1468
- 4 Kalman R E. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 1960, **82**: 35–45
- 5 Doucet A, Freitas N, Gordon N. *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
- 6 Mei X, Ling H B. Robust visual tracking using L_1 minimization. In: Proceedings of the 2009 International Conference on Computer Vision. Kyoto, Japan: IEEE, 2009. 1436–1443
- 7 Zhang T Z, Ghanem B, Liu S, Ahuja N. Robust visual tracking via multi-task sparse learning. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence, USA: IEEE, 2012. 2042–2049
- 8 Hong Z B, Mei X, Prokhorov D, Tao D C. Tracking via robust multi-task multi-view joint sparse representation. In: Proceedings of the 2013 International Conference on Computer Vision. Sydney, NSW: IEEE, 2013. 649–656
- 9 Zhang S P, Yao H X, Zhou H Y, Sun X, Liu S H. Robust visual tracking based on online learning sparse representation. *Neurocomputing*, 2013, **100**: 31–40
- 10 Adam A, Rivlin E, Shimshoni I. Robust fragments-based tracking using the integral histogram. In: Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition. New York, USA: IEEE, 2006. 798–805
- 11 He S F, Yang Q X, Lau R W H, Wang J, Yang M H. Visual tracking via locality sensitive histograms. In: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, USA: IEEE, 2013. 2427–2434
- 12 Ross D A, Lim J, Lin R S, Yang M-H. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 2008, **77**(1): 125–141
- 13 Kwon J, Lee K M. Visual tracking decomposition. In: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition. San Francisco, USA: IEEE, 2010. 1269–1276
- 14 Liu B Y, Huang J Z, Yang L, Kulikowsk C. Robust tracking using local sparse appearance model and k -selection. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. Providence, USA: IEEE, 2011. 1313–1320
- 15 Yang F, Lu H C, Yang M H. Robust superpixel tracking. *IEEE Transactions on Image Processing*, 2014, **23**(4): 1639–1651
- 16 Babenko B, Yang M H, Belongie S. Visual tracking with online multiple instance learning. In: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami FL, USA: IEEE, 2009. 983–990
- 17 Kalal Z, Matas J, Mikolajczyk K. P-N learning: bootstrapping binary classifiers by structural constraints. In: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition. San Francisco, USA: IEEE, 2010. 49–56
- 18 Xie Y, Zhang W S, Li C H, Lin S Y, Qu Y Y, Zhang Y H. Discriminative object tracking via sparse representation and online dictionary learning. *IEEE Transactions on Cybernetics*, 2014, **44**(4): 539–553

- 19 Zhuang B H, Lu H C, Xiao Z Y, Wang D. Visual tracking via discriminative sparse similarity map. *IEEE Transactions on Image Processing*, 2014, **23**(4): 1872–1881
- 20 Zhong W, Lu H C, Yang M H. Robust object tracking via sparsity-based collaborative model. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence, USA: IEEE, 2012. 1838–1845
- 21 Elhamifar E, Sapiro G, Sastry S. Dissimilarity-based sparse subset selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. DOI: 10.1109/TPAMI.2015.2511748
- 22 Elhamifar E, Vidal R. Sparse subspace clustering: algorithm, theory, and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, **35**(11): 2765–2781
- 23 Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 2010, **3**(1): 1–122



黄丹丹 大连理工大学信息与通信工程学院博士研究生. 2007 年获得长春理工大学电子信息工程学院电子信息科学与技术系学士学位. 主要研究方向为视频序列中的目标检测与目标跟踪.

E-mail: dlut.huang@163.com

(**HUANG Dan-Dan** Ph.D. candidate at the School of Information and Communication Engineering, Dalian University of Technology. She received her bachelor degree from Changchun University of Science and Technology in 2007. Her research interest covers object detection and object tracking.)



孙怡 大连理工大学信息与通信工程学院教授. 1986 年获得大连理工大学电子系学士学位. 主要研究方向为图像处理, 模式识别与无线通信. 本文通信作者. E-mail: lslwf@dlut.edu.cn

(**SUN Yi** Professor at Dalian University of Technology. She received her bachelor degree from Dalian University of Technology in 1986. Her research interest covers image processing, pattern recognition, and wireless communication. Corresponding author of this paper.)