论 文

基于 MAP 多子空间增量学习的目标跟踪算法

李康, 何发智*, 陈晓

武汉大学计算机学院, 武汉 430000

* 通信作者. E-mail: fzhe@whu.edu.cn

收稿日期: 2015-07-15; 接受日期: 2015-08-28; 网络出版日期: 2016-03-15

国家自然科学基金 (批准号: 61472289) 和湖北省自然科学基金 (批准号: 2015CFB254) 资助项目

摘要 目标跟踪是计算机视觉中的一个重要领域. 基于 PCA 子空间学习的跟踪算法假设目标从子空间中生成并且误差项服从小方差的 Gauss 分布. 但是这些算法没有考虑样本在子空间中的先验分布, 在以重构误差最小化为目标进行优化时, 容易产生过度拟合测试样本现象. 为了解决过度拟合问题, 本文首先从最大似然估计 (ML) 角度分析了基于 PCA 子空间学习的算法原理, 然后推导出了样本投影在子空间中坐标方差的无偏估计, 最后提出了基于最大后验概率 (MAP) 的 PCA 子空间学习算法. 在处理跟踪过程中出现的遮挡问题时, 本文提出了一种局部跟踪策略. 首先, 将目标图像分成多个块, 假设每个块都从独立的子空间中生成. 其次, 根据每个分块对目标的重构误差大小判断该分块是否被遮挡. 最后, 在计算粒子权值和更新子空间时仅使用未被遮挡的分块, 从而避免了遮挡物对跟踪结果的干扰. 实验表明本文提出的算法能有效地解决目标的部分遮挡、运动模糊和背景干扰等问题, 在与其他算法的比较中体现了本算法的优越性.

关键词 目标跟踪 主成分分析 子空间方法 最大后验概率 最大似然估计

1 引言

目标跟踪在实时监控、运动捕捉、视频分析和娱乐等领域有着广泛的应用, 是计算机视觉中最为活跃的领域之一^[1~3]. 近年来国内外有大量的跟踪算法被开发出来^[4~6], 但是由于目标本身的信息不足, 以及光照变化、遮挡、目标旋转等各种因素的影响, 开发出具有鲁棒效果的跟踪算法仍然是一项具有挑战性的研究课题.

无模型 (model free) 的目标跟踪仅给出目标在第 1 帧中的位置, 跟踪算法负责在余下的视频序列中找出目标的位置. 文献 [7, 8] 关注目标的静态外观模型, 但是由于目标信息的不足, 当目标遭受剧烈形变时会跟踪失败. 自适应外观模型的跟踪算法^[9~20] 通常在跟踪过程中动态更新分类器, 能够适应目标新的变化, 具有更好的鲁棒性, 已经成为当前目标跟踪研究的主流方向.

随着近年来机器学习的发展, 大量的机器学习算法被应用到目标跟踪上来. 根据外观模型的不同, 跟踪算法可以分为判别模型和产生模型两类.

引用格式: 李康, 何发智, 陈晓. 基于 MAP 多子空间增量学习的目标跟踪算法. 中国科学: 信息科学, 2016, 46: 476–495, doi: 10.1360/N112015-00104

判别模型跟踪算法^[9~14]将跟踪问题看成分类问题,它使用训练样本训练分类器将目标与背景分开,然后在候选样本中选择具有最大分类响应的样本作为目标.多实例目标跟踪(MIL)^[9]使用在线 Boosting 算法训练分类器,能够处理非线性分类问题,提高了跟踪鲁棒性.但是基于 Boosting 的算法需要在每一帧迭代地对弱分类器进行选择,降低了跟踪速度.根据压缩感知中的压缩采样和稀疏重建理论^[21,22],压缩感知跟踪算法(CT)^[10]根据信号的稀疏特性,使用随机采样的方式从目标中提取类哈尔(Haar-like)特征,并在目标周围指定的区域内采集正样本和负样本训练 Bayes 分类器,最后选择具有最大分类响应的候选样本作为目标.类哈尔特征的计算速度非常快,使用积分图计算图像中任意尺寸的特征可以在常数时间内得出. Bayes 分类器的训练和更新时间复杂度低,使得 CT 在跟踪速度上较 MIL 有很大的提高.

产生模型跟踪算法^[15~20]通过训练样本学习到目标的外观,然后在候选样本中选择重构误差最小的样本作为目标.在产生模型的跟踪算法中一般使用粒子滤波跟踪框架^[23],它不受系统满足线性和 Gauss 分布的限制,能够处理多峰问题,并且使用灵活,易于实现.近年来,基于子空间学习的目标跟踪算法^[15,24~27]受到了专家学者的关注,产生了大量的科研成果.大量的实验表明,基于 PCA 子空间的学习算法能够有效应对目标缓慢的姿态变化、光照变化和旋转等影响因素带来的问题.增量跟踪算法(IVT)^[15]假设目标是从 PCA 基向量张成的子空间内生成,并选择重构误差最小的候选样本作为目标,在得到目标位置后通过 SKL^[28]算法增量训练 PCA 子空间. LSST^[29]算法假设目标从 PCA 子空间中生成,并假设噪声服从 Gauss-Laplace 分布.当候选目标中有异常点时,通过 LSS 距离能够更有效地测量候选样本与训练字典之间的相似程度.基于稀疏表示的跟踪算法(L1T)^[16]使用目标模板和琐碎模板的线性组合重构目标样本,通过求解 L1 最小化问题得到稀疏系数,最后选择重构误差最小的候选样本作为目标.由于稀疏系数需要迭代求解,因此算法整体时间复杂度较高,达不到实时跟踪的要求. L1APG^[18]在 L1T 的基础上增加了对琐碎模板系数的 L2 范数约束,并设计了快速求解算法,提高了跟踪速度和鲁棒性.文献[17]将 L1T 中的 L1 约束转变成 L2 约束,直接求得系数的解析解,提高了跟踪速度,但求解的系数不再具有稀疏性.文献[19]在 L1T^[16]的基础上引入了结构化的局部稀疏外观模型,在模板更新上使用从 PCA 子空间中生成的样本作为更新样本,在处理被遮挡的目标时提高了跟踪算法的精确度.文献[20]使用 PCA 基和琐碎模板的组合作为基向量,对琐碎模板的系数加上稀疏性约束来建模噪声,提高了算法应对部分遮挡和运动模糊问题时的精确度.以上基于 PCA 子空间产生模型算法仅使用极大似然估计求出候选样本在子空间中的位置,而没有考虑到样本在子空间中的先验分布,导致在对候选样本求解重构误差时发生过度拟合候选样本的情况.

在处理跟踪问题时,许多算法对目标区域进行了分块处理.文献[30]将目标图像分成多个图像块,通过比较各图像块与目标的相似度确定目标位置.文献[31]提出使用多窗口均值移动跟踪算法,将目标用多个窗口表示,每个窗口对应一个跟踪器,最后根据这些窗口之间的位置关系来估计目标状态.文献[32]在 Bayes 理论框架下提出了基于局部分块学习的在线视觉跟踪算法.

在上述研究的基础上,本文提出了一种在粒子滤波框架下基于分块策略的子空间增量学习算法.为了解决文献[15,19,20]中存在的过度拟合问题,本文从最大似然估计(ML)角度研究了基于 PCA 子空间的跟踪算法,根据 PCA 理论推导并证明了训练样本在子空间内的坐标分布,提出了基于最大后验概率估计(MAP)的 PCA 子空间增量学习跟踪算法.基于 MAP 的算法比基于 ML 的算法计算出的重构误差值更加精确,提升了跟踪算法的综合性能.在处理目标部分遮挡问题上,本文提出了一种分块选择策略.利用基于 MAP 的 PCA 子空间学习算法能够精确计算重构误差的特性,在计算粒子权值和训练子空间时仅使用重构误差较小的分块,有效地避免了使用被遮挡的目标训练子空间而导致的跟踪失败情况.实验表明,本文提出的算法能够有效处理跟踪中的遮挡、形变、运动模糊等情况,最终

图 1 (网络版彩图) 跟踪基本流程

Figure 1 (Color online) Basic flow of tracking

达到鲁棒的跟踪效果.

2 粒子滤波框架

本章主要介绍目标跟踪的粒子滤波框架. 目标跟踪问题可以看作是隐 Markov 模型中隐状态变量的 Bayes 最大后验概率估计问题. 设 $\mathbf{x}_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t\}$ 为 t 时刻目标的仿射运动参数变量^[15], 其中 x_t, y_t 表示目标相对图像边界在 x 轴和 y 轴方向的平移; θ_t 表示目标在第 t 帧的旋转角度; s_t 表示目标在第 t 帧的尺度; α_t 表示目标在第 t 帧变换的纵横比; ϕ_t 表示目标在第 t 帧的倾斜角度. 在获得从开始时刻至 t 时刻的观测值 $\mathbf{I}_{1:t}$ 后, 由 Bayes 推论得

$$p(\mathbf{x}_t | \mathbf{I}_{1:t}) \propto p(\mathbf{I}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{I}_{1:t-1}) d\mathbf{x}_{t-1}, \quad (1)$$

其中 $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ 表示连续两帧的状态转移模型, $p(\mathbf{I}_t | \mathbf{x}_t)$ 表示观测似然模型. 一般假设状态转移模型服从 Gauss 分布, 即 $p(\mathbf{x}_t | \mathbf{x}_{t-1}) = N(\mathbf{x}_t | \mathbf{x}_{t-1}, \Omega)$, 其中 $\Omega = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2)$ 是一个对角矩阵, 它的对角线元素对应仿射参数的方差, 这些方差描述了目标在帧与帧之间状态变化的剧烈程度, 更复杂的状态转移模型参考文献 [33]. 跟踪算法的基本流程如图 1 所示. 在已知第 1 到 $t-1$ 时刻目标观测值 $\mathbf{I}_{1:t-1}$ 的情况下, 在第 t 时刻根据状态转移模型散布粒子, 根据观测似然模型得出每个粒子的置信度作为该粒子的权值, 最后选择权值最高的粒子作为目标在 t 时刻的状态, 即

$$\hat{\mathbf{x}}_t = \arg \max_{\mathbf{x}_t} p(\mathbf{I}_t | \mathbf{x}_t). \quad (2)$$

在每一帧跟踪完成后, 根据粒子的权值进行重采样, 防止粒子退化.

3 基于 ML 的 PCA 子空间学习跟踪算法

本章从最大似然估计 (ML) 的角度分析 PCA 子空间学习跟踪算法^[15], 这是后面章节提出基于最大后验概率 (MAP) 的 PCA 子空间学习跟踪算法的基础.

设 $T = [\mathbf{I}_1 \cdots \mathbf{I}_n]$ 是 $d \times n$ 维的矩阵, 其中 $\mathbf{I}_i$ 表示训练样本. 令 $A = [\mathbf{I}_1 - \bar{\mathbf{I}}_T \cdots \mathbf{I}_n - \bar{\mathbf{I}}_T]$, 其中 $\bar{\mathbf{I}}_T = \frac{1}{n} \sum_{i=1}^n \mathbf{I}_i$. A 的奇异值分解为 $A \stackrel{\text{SVD}}{=} U \Sigma V^T$, 其中 $U \in \mathbb{R}^{d \times r}$ 为左奇异矩阵, $\Sigma \in \mathbb{R}^{r \times r}$ 是对角线为奇异值的对角矩阵, $V \in \mathbb{R}^{r \times n}$ 为右奇异矩阵. 给出候选目标块 $\mathbf{I}_t$, 令 $\mathbf{y} = \mathbf{I}_t - \bar{\mathbf{I}}_T$, 假设 $\mathbf{y}$ 是从 U 张成的子空间中生成的, 中心点在 $\bar{\mathbf{I}}_T$, $\mathbf{z}$ 是对应的坐标向量, 式 (3) 解释了目标图像与子空间

坐标之间的关系:

$$\mathbf{y} = U\mathbf{z} + \boldsymbol{\varepsilon}, \quad (3)$$

其中 $\boldsymbol{\varepsilon} = [\varepsilon_1 \cdots \varepsilon_d]^T$ 为重构误差. 在 PCA 中, 假设 $\varepsilon_i \sim N(0, \sigma^2)$, 其中 σ 表示数值很小的方差. 则

$$p(\mathbf{y}|\mathbf{z}) = \prod_{i=1}^d \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\varepsilon_i^2}{2\sigma^2}\right) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^d \exp\left(-\frac{\|\boldsymbol{\varepsilon}\|_2^2}{2\sigma^2}\right). \quad (4)$$

$\mathbf{z}$ 的似然函数为 $l(\mathbf{z}) = p(\mathbf{y}|\mathbf{z})$, $\mathbf{z}$ 的最大似然估计相当于求解式 (5),

$$\begin{aligned} \hat{\mathbf{z}}_{\text{ML}} &= \arg \max_{\mathbf{z}} l(\mathbf{z}) \\ &= \arg \max_{\mathbf{z}} \log \left(\prod_{i=1}^d \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\varepsilon_i^2}{2\sigma^2}\right) \right) \\ &= \arg \max_{\mathbf{z}} \left(-d \log \sqrt{2\pi}\sigma - \frac{1}{2\sigma^2} \sum_{i=1}^d \varepsilon_i^2 \right) \\ &= \arg \min_{\mathbf{z}} \|\boldsymbol{\varepsilon}\|_2^2 \\ &= \arg \min_{\mathbf{z}} \|\mathbf{y} - U\mathbf{z}\|_2^2. \end{aligned} \quad (5)$$

对式 (5) 中的变量 $\mathbf{z}$ 求导得

$$\frac{d\left(\|\mathbf{y} - U\mathbf{z}\|_2^2\right)}{d\mathbf{z}} = 2(-U^T)(\mathbf{y} - U\mathbf{z}). \quad (6)$$

令式 (6) 等于 0 可得

$$\hat{\mathbf{z}}_{\text{ML}} = U^T \mathbf{y}. \quad (7)$$

式 (7) 为 $\mathbf{z}$ 的最大似然估计, 其中 $U^T U = I_r$ 为 $r \times r$ 维的单位矩阵. 则其重构误差为 $\hat{\boldsymbol{\varepsilon}}_{\text{ML}} = \mathbf{y} - U\hat{\mathbf{z}}_{\text{ML}}$. 在基于 PCA 子空间学习的跟踪算法中, 令 $p(\mathbf{I}_t|\mathbf{x}_t) = p(\mathbf{y}; \mathbf{z} = \hat{\mathbf{z}}_{\text{ML}})$, 其中

$$p(\mathbf{y}; \mathbf{z} = \hat{\mathbf{z}}_{\text{ML}}) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^d \exp\left(-\frac{\|\hat{\boldsymbol{\varepsilon}}_{\text{ML}}\|_2^2}{2\sigma^2}\right), \quad (8)$$

可得观测似然函数与重构误差之间的关系:

$$p(\mathbf{I}_t|\mathbf{x}_t) \propto \exp\left(-\frac{\|\hat{\boldsymbol{\varepsilon}}_{\text{ML}}\|_2^2}{2\sigma^2}\right). \quad (9)$$

在 PCA 理论中, 样本集映射到第一主成分坐标的方差大于第二主成分, 以此类推, 越次要的主成分其对应坐标的方差就越小. 基于 ML 的算法没有对候选样本映射到子空间的坐标变化进行约束, 在求解重构误差时, 次要主成分所对应的坐标大幅变化拟合候选样本, 发生了过度拟合. 若保留的主成分数量过多, 基于 ML 的算法求得的重构误差会非常小, 这时候重构误差的值便不能反映候选目标样本与训练模型之间的差异. 当子空间维度 (即保留的主成分数量) 等于样本维度的时候, 误差 $\|\hat{\boldsymbol{\varepsilon}}_{\text{ML}}\|_2^2$ 将恒等于 0, 此时基于 ML 的跟踪算法将失效. 此外, 在基于 PCA 子空间学习的目标跟踪算法中, 假设误差项服从小方差的 Gauss 分布^[15], 但是在跟踪中当目标发生被遮挡等情况时, 该假设不成立. 因此, 基于 PCA 子空间学习的跟踪算法对目标的部分遮挡较为敏感. 另一方面, 基于 PCA 子空间学习的跟踪算法在更新子空间时只是简单地使用新的观测目标作为训练样本, 而没有考虑到观测目标是否被遮挡^[15]. 当使用被遮挡的目标样本进行增量训练时, 会造成误差的积累, 从而降低跟踪效果.

4 基于 MAP 的 PCA 子空间学习跟踪算法

为了解决基于 ML 法求解重构误差时产生过度拟合的问题, 本章首先推导并证明了训练样本在子空间中坐标的分布, 进而提出了基于最大后验概率估计的 PCA 子空间学习目标跟踪算法. PCA 假设输入的数据在主成分的方向上符合 Gauss 分布, 且数据投影在第一主成分上的方差大于其投影在第二主成分的方差, 依次类推. 在式 (3) 中, 假设 $\mathbf{z}$ 是子空间中的坐标向量, 将 $\mathbf{z}$ 的各个维度当作随机变量, 则它的分布可以用矩阵 A 在子空间中的投影 $U^T A$ 的统计特征估计. 在下面的两个定理中, 各符号的含义与前述相同, 这里只给出定理的结果和证明.

定理1 $U^T A$ 的行元素相加之和为 0.

证明 令 $A = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, $U^T = \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_r^T \end{bmatrix}$, 其中 $\mathbf{a}_i, \mathbf{u}_i$ 为 d 维列向量, 并且 $(\mathbf{a}_1 + \dots + \mathbf{a}_n) = \mathbf{0}$.

则 $U^T A = \begin{bmatrix} \mathbf{u}_1^T \mathbf{a}_1 & \dots & \mathbf{u}_1^T \mathbf{a}_n \\ \vdots & \ddots & \vdots \\ \mathbf{u}_r^T \mathbf{a}_1 & \dots & \mathbf{u}_r^T \mathbf{a}_n \end{bmatrix}$, 其第 i 行之和为 $\mathbf{u}_i^T (\mathbf{a}_1 + \dots + \mathbf{a}_n) = 0$.

定理2 假设 $\mathbf{z} \sim N(0, W^2)$, 其中 $W \in \mathbb{R}^{r \times r}$ 是对角矩阵, 其对角线为输入数据投影在主成分方向上的标准差. 则 W 的无偏估计为 $\frac{\Sigma}{\sqrt{n-1}}$.

证明 令 $\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}$, $V = \begin{bmatrix} v_{11} & \dots & v_{1r} \\ \vdots & \ddots & \vdots \\ v_{n1} & \dots & v_{nr} \end{bmatrix}$, 则 $\Sigma V^T = \begin{bmatrix} \sigma_1 v_{11} & \dots & \sigma_1 v_{n1} \\ \vdots & \ddots & \vdots \\ \sigma_r v_{1r} & \dots & \sigma_r v_{nr} \end{bmatrix}$.

令 $b_i = \sigma_i [v_{1i} \dots v_{ni}]$ 表示 ΣV^T 的第 i 行元素, 因为 $U^T A = \Sigma V^T$, 由定理 1 可知: $\bar{b}_i = \frac{\sigma_i}{n} \sum_{j=1}^n v_{ji}$ 为 0.

由于 $\sigma_i \neq 0$, 所以 $\bar{v}_i = \frac{1}{n} \sum_{j=1}^n v_{ji} = 0$. 又因为 V 是列正交矩阵, 可以得出 $\sum_{j=1}^n (v_{ji} - \bar{v}_i)^2 = \sum_{j=1}^n (v_{ji})^2 = 1$, 则 b_i 的标准差的无偏估计为

$$\sigma_i \sqrt{\frac{1}{n-1} \sum_{j=1}^n (v_{ji} - \bar{v}_i)^2} = \frac{\sigma_i}{\sqrt{n-1}},$$

得出 W 的无偏估计为 $\frac{\Sigma}{\sqrt{n-1}}$.

由定理 2 获得随机变量 $\mathbf{z}$ 的先验分布后, $\mathbf{z}$ 的最大后验估计为

$$\hat{\mathbf{z}}_{\text{MAP}} = \arg \max_{\mathbf{z}} p(\mathbf{y}|\mathbf{z})p(\mathbf{z}). \quad (10)$$

对式 (10) 化简得

$$\begin{aligned} \hat{\mathbf{z}}_{\text{MAP}} &= \arg \max_{\mathbf{z}} p(\mathbf{y}|\mathbf{z})p(\mathbf{z}) \\ &= \arg \max_{\mathbf{z}} \log(p(\mathbf{y}|\mathbf{z})p(\mathbf{z})) \\ &= \arg \max_{\mathbf{z}} \log \left(\prod_{i=1}^d \frac{1}{\sqrt{2\pi}\sigma} \exp \left(-\frac{(y_i - \mathbf{u}_i \mathbf{z})^2}{2\sigma^2} \right) \frac{1}{(2\pi)^{r/2}} \frac{1}{W} \exp \left(-\frac{1}{2} \mathbf{z}^T W^{-2} \mathbf{z} \right) \right) \end{aligned}$$

图 2 (网络版彩图) $\hat{z}_{ML}$ 和 $\hat{z}_{MAP}$ 的变化Figure 2 (Color online) The change of $\hat{z}_{ML}$ and $\hat{z}_{MAP}$

$$\begin{aligned}
 &= \arg \max_z d \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \|\mathbf{y} - U\mathbf{z}\|_2^2 - \log((2\pi)^{r/2}) - \log(W) - \frac{1}{2} \|W^{-1}\mathbf{z}\|_2^2 \\
 &= \arg \min_z \|\mathbf{y} - U\mathbf{z}\|_2^2 + \lambda \|W^{-1}\mathbf{z}\|_2^2,
 \end{aligned} \quad (11)$$

其中 $\lambda = \sigma^2$ 是调节参数, 本实验中取 $\lambda = 1$. 对式 (11) 求导得

$$\frac{d(\|\mathbf{y} - U\mathbf{z}\|_2^2 + \lambda \|W^{-1}\mathbf{z}\|_2^2)}{d\mathbf{z}} = -2U^T\mathbf{y} + 2U^TU\mathbf{z} + 2\lambda W^{-2}\mathbf{z}. \quad (12)$$

令式 (12) 等于 0 可得 $\mathbf{z}$ 的 MAP 估计为

$$\hat{\mathbf{z}}_{MAP} = (U^TU + \lambda W^{-2})^{-1}U^T\mathbf{y}, \quad (13)$$

其中, $U^TU = I_r$. 则目标的重构误差为

$$\hat{\mathbf{e}}_{MAP} = \mathbf{y} - U\hat{\mathbf{z}}_{MAP}. \quad (14)$$

类似于式 (9) 构建观测似然模型的方法, 基于 MAP 的 PCA 子空间学习跟踪算法构建观测似然模型为

$$p(\mathbf{I}_t|\mathbf{x}_t) \propto \exp(-\|\hat{\mathbf{e}}_{MAP}\|_2^2). \quad (15)$$

由于 $\hat{\mathbf{z}}_{MAP}$ 有解析解, 在已知 U, Σ 的情况下, 根据式 (15) 计算单个候选样本置信度的时间复杂度为 $O(d^4r)$, 其中 d, r 为 U 的行数和列数.

图 2 显示了使用 ML 和 MAP 跟踪测试序列 Boy 时坐标向量 $\hat{\mathbf{z}}_{ML}$ 和 $\hat{\mathbf{z}}_{MAP}$ 中的分量 (每个坐标向量含有 9 个分量) 随时间的变化. 从图中可以看出使用本文的方法时, 越是次要的主成分, 其波动范围越小, 结果符合 PCA 中次要主成分方差依次减小的原理. 而在传统的基于 ML 的 PCA 子空间学习跟踪算法中, 次要的主成分波动范围依然很大. 图 3 显示了 Boy 序列中第 156 帧至 175 帧的实验结果, 每一列为一帧的跟踪结果数据. 其中上面方框是基于 ML 的跟踪算法结果, 下面方框是基于 MAP 跟踪算法的结果. 每个方框中包含 4 行, 其中每一行表示一种结果, 它们依次为: 当前时刻目标训练样本的均值; 算法跟踪到的目标图像; 重构误差; 重构图像. 从图 3 中可以看出, 基于 MAP 的跟

图 3 (网络版彩图) Boy 序列部分跟踪结果

Figure 3 (Color online) Some tracking results of Boy sequence

踪算法重构误差比基于 ML 的跟踪算法大 (轮廓比较明显). 结合实际情况看, 由于目标的头部在不停地变动, 基于 MAP 的跟踪算法的重构误差能够准确地反映新目标外观与学习到的目标外观模型之间的差异, 而基于 ML 的跟踪算法的重构误差基本为 0 (黑色占绝大部分数据), 这是因为基于 ML 的跟踪算法产生了过度拟合. 基于 MAP 的跟踪算法符合 Occam 剃刀原理, 它相当于在目标函数上加入了正则化项, 提高了模型的泛化能力.

5 多子空间目标跟踪算法

本章介绍利用分块策略解决部分遮挡问题, 提出了多子空间的目标跟踪算法. 部分遮挡是影响跟踪效果的重要因素. 在基于 PCA 子空间学习的跟踪算法中, 由于被遮挡的部分不能够用服从小方差 Gauss 分布的误差项建模, 所以算法对部分遮挡较为敏感. 传统的处理遮挡问题的方法是将目标分成多个图像块进行处理. 本文提出一种新的分块跟踪算法. 首先, 在训练阶段以分块为单位训练子空间, 假设所有子空间是相互独立的; 然后, 根据分块的重构误差判断该块是否被遮挡, 在构建观测似然模型时选择未被遮挡的分块计算粒子权值, 有效地避免了跟踪框漂移到遮挡物的情况; 最后, 在更新阶段, 将新目标区域中未被遮挡的分块添加到训练样本, 防止算法学习到被遮挡物的特征.

在对目标图像进行分块时, 如果图像块过小, 容易产生过度拟合现象; 如果图像块过大, 则难以精确描述被遮挡的区域. 本文使用的分块策略如图 4 所示, 首先将目标对齐至 32×32 像素的图像块, 然后从左上角依次采集 $l = 9$ 个 16×16 的图像块, 步长为 8 个像素, 其中相邻的小图像块有部分重叠. 这种分块策略既保证了每个图像块对应的子空间包含足够的信息, 又保证了图像块能够比较精确地描述被遮挡区域.

使用分块策略后, 对每个分块分别训练子空间. 将文中的符号重新定义为

$$T^{(i)} = \left[\mathbf{I}_1^{(i)}, \dots, \mathbf{I}_{n_i}^{(i)} \right], \bar{\mathbf{I}}_{T^{(i)}} = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{I}_j^{(i)}, A^{(i)} = \left[\mathbf{I}_1^{(i)} - \bar{\mathbf{I}}_{T^{(i)}} \cdots \mathbf{I}_{n_i}^{(i)} - \bar{\mathbf{I}}_{T^{(i)}} \right],$$

其中上标 (i) 表示第 i 个分块, n_i 表示第 i 个分块训练样本的个数. 训练样本的奇异值分解为 $A^{(i)} \stackrel{\text{SVD}}{=} U^{(i)} \Sigma^{(i)} (V^{(i)})^T$, 给出候选样本 $\mathbf{I}_t$, 令 $\mathbf{y}^{(i)} = \mathbf{I}_t^{(i)} - \bar{\mathbf{I}}_{T^{(i)}}$, 则根据式 (14) 可以得出第 i 块

图 4 (网络版彩图) 目标分块方法

Figure 4 (Color online) Local patch model

图 5 (网络版彩图) 重构误差变化

Figure 5 (Color online) Changes of reconstruction error

的重构误差为

$$\hat{\epsilon}_{\text{MAP}}^{(i)} = \mathbf{y}^{(i)} - U^{(i)} \left(\left(U^{(i)} \right)^T U^{(i)} + \lambda \left(W^{(i)} \right)^{-2} \right)^{-1} \left(U^{(i)} \right)^T \mathbf{y}^{(i)}. \quad (16)$$

在使用分块策略时, 系统维持了一个表示当前目标未被遮挡的分块序号的集合 S_t . 图 5 显示了测试序列 Woman 中目标被遮挡部分时各个分块的重构误差曲线图. 可以看出, 目标被遮挡的分块 (第 4, 5, 6, 7, 8, 9 块) 重构误差明显比未被遮挡的分块 (第 1, 2, 3 块) 重构误差大. 假设重构误差大于某个阈值的分块被判断为被遮挡了, S_t 中的元素表示在第 t 帧中重构误差不超过阈值 H 的块序号. 假设第 $t-1$ 帧中估计的目标状态为 $\hat{\mathbf{x}}_{t-1}$, 其对应的各个分块的误差项为 $\hat{\epsilon}_{\text{MAP},t-1}^{(i)}$, 则

$$S_{t-1} = \left\{ i \mid \left\| \hat{\epsilon}_{\text{MAP},t-1}^{(i)} \right\|_2^2 < H, i \in \{1, \dots, l\} \right\}, \quad (17)$$

其中 H 是阈值参数 (本文中取 $H = 10$), 下标 $t-1$ 对应当前跟踪到的帧数. 假设各个子空间是相互

独立的, 则有

$$p(\mathbf{I}_t | \mathbf{x}_t) \propto \prod_{i \in S_{t-1}} \exp \left(- \left\| \hat{\mathbf{e}}_{\text{MAP}}^{(i)} \right\|_2^2 \right). \quad (18)$$

式 (18) 并不是每次计算粒子的权值都使用所有的分块¹⁾, 而是使用上一帧中跟踪比较准确的分块. 在 PCA 子空间学习中, 假设目标是从同一个子空间中生成, 它的重构误差非常小. 反之, 若某个分块的重构误差非常大, 则不能认为它是从目标子空间中生成的. 这时候则不使用重构误差过大的分块计算粒子权值和更新子空间.

6 子空间更新

由于目标在跟踪过程中不断地变化, 为了使跟踪算法能够适应目标新的外观形态, 需要进行在线学习. 在文献 [15, 16] 中, 直接使用当前的目标观测值进行增量学习. 在本研究中若使用分块策略, 并不是每次都使用全部的分块更新子空间. 假设第 t 帧目标的状态为 $\hat{\mathbf{x}}_t$, 对于 $i \in S_t$, 设其对应的分块为 $\hat{\mathbf{I}}_t^{(i)}$. 设分块后的训练数据集为 $\tilde{A}^{(i)} = [\mathbf{I}_1^{(i)} - \bar{\mathbf{I}}_{\tilde{T}^{(i)}}, \dots, \mathbf{I}_{n_i}^{(i)} - \bar{\mathbf{I}}_{\tilde{T}^{(i)}} \hat{\mathbf{I}}_t^{(i)} - \bar{\mathbf{I}}_{\tilde{T}^{(i)}}]$. 其中 $\bar{\mathbf{I}}_{\tilde{T}^{(i)}} = (\sum_{j=1}^{n_i} \mathbf{I}_j^{(i)} + \hat{\mathbf{I}}_t^{(i)}) / (n_i + 1)$, 以及 $\tilde{A}^{(i)} \stackrel{\text{SVD}}{=} \tilde{U}^{(i)} \tilde{\Sigma}^{(i)} (\tilde{V}^{(i)})^T$ 中的 $\tilde{U}^{(i)}$, $\tilde{\Sigma}^{(i)}$ 可以使用 SKL 算法 [15, 28] 增量计算得出. 相对于直接计算 PCA 子空间基向量, 使用 SKL 算法增量计算 PCA 子空间的基向量, 能够将时间复杂度从 $O(d_i(n_i + 1)^2)$ 降低到 $O(d_i)$, 其中 d_i 表示第 i 个分块内的像素数量, n_i 表示第 i 个分块训练样本的个数.

算法 1 描述了增量训练子空间的算法, 其中 $f \in [0, 1]$ 是遗忘因子 [15]. 它表示降低以前训练样本的权重, 使得跟踪算法更关注最近观测到的目标状态. 算法 2 描述了 PMAPI 的算法流程.

算法 1 增量训练子空间的 SKL 算法

输入: $A^{(i)}$ 的 SVD 分解 $U^{(i)}$, $\Sigma^{(i)}$, 以及 $\bar{\mathbf{I}}_{T^{(i)}}$, $\hat{\mathbf{I}}_t^{(i)}$;

输出: $\bar{\mathbf{I}}_{\tilde{T}^{(i)}}$, $\tilde{U}^{(i)}$, $\tilde{\Sigma}^{(i)}$;

- 1: $\bar{\mathbf{I}}_{\tilde{T}^{(i)}} = \frac{fn}{fn+1} \bar{\mathbf{I}}_{T^{(i)}} + \frac{1}{fn+1} \hat{\mathbf{I}}_t^{(i)}$.
 - 2: 定义 $\hat{B} = \begin{bmatrix} \mathbf{0} & \sqrt{\frac{n}{n+1}} (\bar{\mathbf{I}}_{T^{(i)}} - \hat{\mathbf{I}}_t^{(i)}) \end{bmatrix}$, 计算 $\tilde{B} = QR(\hat{B} - U^{(i)}(U^{(i)})^T \hat{B})$, 其中 $\tilde{B}$ 为 QR 分解中的正交矩阵.
 - 3: 构建 $R = \begin{bmatrix} f\Sigma^{(i)} & (U^{(i)})^T \tilde{B} \\ 0 & \tilde{B}^T (\hat{B} - U^{(i)}(U^{(i)})^T \hat{B}) \end{bmatrix}$, 计算 $R \stackrel{\text{SVD}}{=} U' \Sigma' V'^T$, 则 $\tilde{U}^{(i)} = \begin{bmatrix} U^{(i)} & \tilde{B} \end{bmatrix} U'$, $\tilde{\Sigma}^{(i)} = \Sigma'$.
-

算法 2 PMAPI 算法流程

- 1: 初始化第一帧的目标位置. 初始化粒子状态为第一帧目标的状态. 初始化 $t = 1$, $S_t = \{1, \dots, l\}$, $\bar{\mathbf{I}}_{A^{(i)}} = \mathbf{I}_1^{(i)}$, $U^{(i)} = [1, 0, 0, \dots, 0]^T$, $\Sigma^{(i)} = [10^{-5}]$, $i \in \{1, \dots, 9\}$ 为对应分块的参数.
 - 2: 进入 $t = t + 1$ 帧, 根据状态转移方程传播粒子.
 - 3: 根据每个粒子表示的状态提取出对应当前帧的窗口, 对窗口进行标准化后分块. 并根据式 (18) 计算粒子的权重.
 - 4: 将权重最大的粒子作为目标在当前帧的状态. 根据该粒子分块对应的重构误差选择出 S_{t+1} , 对 S_{t+1} 中的分块根据算法 1 更新对应分块的 $U^{(i)}$, $\Sigma^{(i)}$, $\bar{\mathbf{I}}_{A^{(i)}}$.
 - 5: 若已经到达最后一帧则跟踪结束, 否则转到步骤 2.
-

1) 假设目标在任何时刻都不会整体从场景中消失, 所以本算法至少有 1 个分块小于阈值 H .

表 1 测试视频特征

Table 1 Characteristic of test sequences

	Complex background	Background interference	Occlusion	Pose variation	Scale change	Illumination change	Camera shaking	Motion blur
Basketball	✓	✓	✓	✓	×	✓	×	×
Boy	×	×	×	✓	×	×	✓	✓
David	×	×	×	✓	✓	×	×	×
David2	✓	×	×	✓	×	×	×	×
David3	×	✓	✓	✓	×	×	×	×
Fish	×	✓	×	×	×	✓	×	×
Jumping	×	×	×	×	×	×	✓	✓
Bike	×	✓	×	✓	×	×	×	×
Woman	×	✓	✓	×	✓	×	×	×

7 实验与分析

7.1 实验配置

在本实验中, 目标的初始位置由用户手工指定, 在处理图像时, 首先将每一帧的彩色图像转化为灰度图像, 然后将图像中每个像素的灰度值归一化至 $[0, 1]$. 实验基本参数设置如下: $\sigma_x = 8$, $\sigma_y = 8$, $\sigma_\theta = 0.001$, $\sigma_s = 0.001$, $\sigma_\alpha = 0.002$, $\sigma_\phi = 0.001$, $\lambda = 1$, 粒子数为 600. 每个粒子在计算前都被归一化为 32×32 像素的正方形图像, 然后分成 16×16 的 9 个小块, 阈值 $H = 10$ 为经验参数. 子空间的最大维度为 9 维, 在跟踪过程中如果子空间维度超过了 9, 则仅保留前 9 个主成分方向, 超出的维度作舍去处理.

本文将基于最大后验概率的增量学习跟踪算法简称为 MAPI; 在 MAPI 上加入分块策略后的跟踪算法简称为 PMAPI. 本文将通过在不同测试视频序列上的模拟实验, 将所提出的 MAPI 和 PMAPI 与世界先进的跟踪算法对比, 评估本文提出算法的有效性. 进行对比的算法为 CT^[10], FRAG^[30], TLD^[13], ASLA^[19], L1APG^[18], MIL^[9], VTS^[34], IVT²⁾^[15], SCM^[35], LSST^[29], ONNDL^[36]. 进行对比的测试视频序列为 Basketball, Boy, David, David2, David3, Fish, Jumping, Bike, Woman. 这些测试序列同时含有实验室环境下和现实生活采集的视频序列; 既包含室内环境也含有室外环境; 囊括了目标跟踪过程中常遇到的光照变化、遮挡、运动模糊、复杂背景等各种情况, 能够比较全面地测试跟踪算法的性能. 测试序列以及对应的人工标定目标位置文件可以从文献^[37]获得.

表 1 列出了各个测试视频中目标具有的特点, 这也是跟踪的主要难点. 其中, 复杂背景表示背景中包含与目标颜色相似的物体; 背景干扰表示目标跟踪框中包含部分背景区域; 遮挡表示目标的某些部分被遮挡物遮挡了; 形变表示目标在跟踪过程中发生的旋转, 以及目标本身的变化; 尺度变化表示目标距离相机的远近导致的近大远小现象; 相机抖动表示相机的焦点短时间内剧烈晃动造成整帧图像的模糊; 运动模糊表示由于目标运动速度过快而产生的目标本身的模糊; 光照变化表示目标周围光照环境的变化, 这种影响因素容易导致目标阴影的剧烈变动, 从而影响跟踪效果. 在这些影响因素中, 部分遮挡最难处理. 在遮挡的影响下, 目标的被遮挡部分信息丢失, 如果直接使用跟踪到的目标作为训练样本, 分类器会因学习到遮挡物的信息而影响算法的性能. 从表 1 中可以看出, 本实验所选的测试

2) IVT 为基于 ML 的 PCA 子空间学习的目标跟踪算法.

图 6 (网络版彩图) 中心距离误差

Figure 6 (Color online) Center position error

视频囊括了跟踪中所遇到的各种情况, 能够全面地测试算法的有效性。

本文实验的硬件环境为 Pentium Dual-Core 2.70 GHz CPU, 4 GB RAM, 软件环境为 Windows 7, Matlab 2012a, 编程语言为 Matlab 和 C 语言混合编程。在没有做任何优化的情况下 MAPI 能够达到每秒跟踪 15 帧, PMAPI 能够达到每秒跟踪 4 帧。

7.2 评价方法

为了全面分析各种算法的跟踪效果, 本文采用 3 种评测方法对算法进行评比。定义某一帧的距离误差为在该帧中算法跟踪到的目标中心位置与人工标定的目标中心位置之间的距离, 单位为像素。图 6 是距离误差示意图。图中标题为测试视频序列的名称; 图中左上角方框内为对比算法的名称; 算法名称后面的数值表示该视频所有帧的距离误差之和与视频总帧数的比值, 称为平均距离误差, 单位为像素/帧。跟踪距离误差图能够精确地显示随着跟踪的进行算法跟踪目标位置的精确程度, 但是此图仅反映了跟踪位置的准确程度, 而没有反映跟踪框的大小的影响。

文献 [37] 提出了一种新的评比跟踪结果的算法, 并对当前流行的各种跟踪算法进行了横向对比。本实验采用该文献提出的重合度成功率作为跟踪效果的主要比较方法, 以距离误差成功率作为辅助评

图 7 (网络版彩图) 重合度成功率

Figure 7 (Color online) Success rate of overlap

价方法. 跟踪成功率是指跟踪成功的帧数占视频总帧数的比例. 评判某一帧是否跟踪成功一般有两种评价标准, 第 1 种为重合度判别方法, 第 2 种为中心距离误差判别方法. 在重合度判别方法中定义 $\text{score} = \frac{\text{area}(R \cap G)}{\text{area}(R \cup G)}$ 为重合度, 其中 R 表示算法跟踪的结果区域; G 表示人工标定的目标区域; $\cup$ 和 $\cap$ 分别表示两种区域的并集和交集; 函数 $\text{area}(\bullet)$ 返回区域中的像素个数. 若某帧的 score 值大于给定阈值, 则认为该帧的跟踪是成功的. 在传统的评价方法中一般取阈值为 0.5. 中心距离误差判别方法根据距离误差来判断跟踪是否成功. 若某帧中距离误差小于给定的阈值, 则认为该帧的跟踪是成功的. 在传统的评价方法中一般取该阈值为 20 个像素.

图 7 是重合度成功率示意图. 其中横轴表示重合度的阈值, 纵轴表示重合度大于该阈值的帧数占总帧数的比例; 算法名称后面的数字表示曲线与坐标轴围成的面积值 (area under curve, AUC). 重合度成功率图能够从整体上显示算法跟踪结果的准确程度, 其中 AUC 值能够综合地反映算法对测试视频的跟踪效果. 使用重合度作为评判标准能够准确反映跟踪结果与标定结果的相似程度. 图 8 是距离误差成功率示意图. 其中横轴表示距离误差阈值, 纵轴表示距离误差小于该阈值的帧数占总帧数的比例. 算法名称后面的数字为 AUC 值. 距离误差成功率图的 AUC 值能够从整体上反映跟踪结果在跟踪目

图 8 (网络版彩图) 距离误差成功率

Figure 8 (Color online) Precision of location error

标位置的精确程度. 例如, 如果在阈值为 20 像素时成功率还没有达到 1 的话, 就说明跟踪过程中某个时间段内发生了严重的漂移. 图 9 是平均重合度成功率示意图和平均距离误差成功率示意图, 它们表示在给定阈值下算法在 9 个测试视频中的跟踪成功率的平均值; 算法名称后面的数字为 AUC 值. 图 10 是本文算法与其他算法的部分跟踪结果.

7.3 实验结果分析

从图 9 的两张图中可以看出, 本文提出的 MAPI 和 PMAPI 算法的 AUC 值均高于其他算法, 说明了本文提出的算法在整体性能上达到了先进水平. 下面结合图 10, 根据各个测试视频的特点作针对性的分析.

Basketball. 该测试视频中的目标为非刚性物体, 目标区域是非凸的, 即, 在目标的包围框中包含有较多的背景区域, 如目标头部两侧的区域以及目标跑动时两腿之间的区域. 这些包围框中的非目标区域在目标运动时随着背景的变动而变动, 使得训练的跟踪器在确定目标时产生混淆, 造成跟踪困难. 从图 6 中可以看出, PMAPI, MAPI, VTS, LSST 和 FRAG 能够比较准确地跟踪到目标, 在跟踪过程中

图 9 (网络版彩图) 平均重合度成功率和平均距离误差成功率

Figure 9 (Color online) Average success rate of overlap and average precision of location error

距离误差没有大的波动. ONNDL 在第 650 帧之后丢失了目标. 由于在第 200 帧至第 400 帧之间目标发生了非刚性运动, 从图 6 中可以看出, IVT 在这段时间内产生了动荡随后丢失了目标, 而 MAPI 在这段时间内能够准确地跟踪到目标. 从图 7 中的 AUC 值上可以看出 VTS 跟踪效果最好, AUC 值在 0.5 以上的算法有 PMAPI, MAPI, FRAG, VTS 和 ONNDL. 从图 8 中可以看出, VTS 跟踪最为准确, 在距离误差为 15 个像素时达到了跟踪成功率 1. 这是由于 VTS 能够根据当前跟踪环境组建跟踪器, 能够有效应对目标短时间被遮挡和光照变化的情况. 从图 10 中可以看出, 第 23 帧, 目标被短时间遮挡的情况下, TLD 丢失了目标. 第 303 帧 L1APG 漂移到了非目标队员的身上, 这是由于该算法中使用的目标模板与背景队员颜色非常相近, 对目标的选择造成了干扰. 第 650, 700 帧中, 由于光照的变化, ONNDL 丢失了目标, MIL 也产生了一定的漂移. 从测试结果中可以看出, 本文算法在有背景干扰和光照变化的情况下仍然能够保证跟踪精度, 体现了算法的鲁棒性.

Boy. 该测试视频是在实验室环境下采集的, 其主要难点在于: 目标的快速移动形成的运动模糊; 相机的抖动形成的整体模糊; 以及目标本身的形变. 从图 6 中可以看出, PMAPI, MAPI, VTS, L1APG, TLD, CT, ONNDL 能够比较准确地跟踪到目标位置. 从图 7 的 AUC 值中可以看出 PMAPI, MAPI, L1APG, ONNDL 跟踪效果最好. 从图 8 的 AUC 值中可以看出 PMAPI, MAPI, L1APG, VTS, CT, TLD, ONNDL 跟踪精确度都很高. 从图 10 中可以看出, 第 205 帧, 目标本身发生了形变, IVT, MIL 发生了漂移. 第 289 帧, 目标快速移动产生了运动模糊, ALSA, L1APG, SCM 发生了漂移, 这主要是因为运动模糊的图像使得琐碎系数不具备稀疏性, 影响了这些算法的跟踪效果. 随着跟踪的进行, 第 353 帧发生了相机抖动, FRAG 也发生了漂移. 这是因为模糊使得目标区域的直方图与算法训练的直方图模板之间产生了巨大的差异, 导致了漂移. 从各项数据可以看出本文算法能够处理目标同时发生的运动模糊、相机抖动和形变等难点.

David. 该测试视频是在实验室环境下采集的, 其主要难点在于: 目标转到侧身过程中尺度由大变; 之后目标又转回正面, 尺度由小变大; 目标本身的形变. 从图 6 中可以看出除 FRAG, LSST 算法外, 其余算法均能够比较准确地跟踪到目标. 从图 7 中的 AUC 值中可以看出, ASLA, TLD 算法跟踪效果最好. 结合图 6 可以看出, 很多算法中心误差距离不大, 但是 AUC 值却不高, 这主要是由于跟踪算法对测试视频中目标的尺度的变化跟踪不够精确. 其中 ASLA, TLD 能够比较好地应对目标的尺

图 10 (网络版彩图) 部分跟踪结果
Figure 10 (Color online) Some tracking results

度变化, MAPI, PMAPI 在处理尺度变化上仅次于它们. 从图 8 中可以看出除了 FRAG, LSST 外, 其余算法都没有发生明显的漂移. 从图 10 中可以看出第 166 帧时, FRAG 发生了漂移, 这是由于 FRAG 没有在线更新机制, 无法适应目标在跟踪过程中的外观变化; PMAPI 在尺度跟踪上稍微不足, 跟踪框中包含了一部分背景区域. 第 212 帧, 目标转过脸时尺度由小变大, 不能适应尺度变化的 CT 算法跟踪结果略微漂移.

David2. 该测试视频是在实验室条件下采集的, 其主要难点在于: 目标区域较小, 难以提取有效的特征; 背景区域与目标有部分相似. 从图 6 中可以看出除 CT, ONNDL 和 FRAG 外, 其余算法均能够比较准确地跟踪到目标. 从图 7 中可以看出除 FRAG, MIL, CT, ONNDL, LSST 外, 其余算法的

AUC 值都超过了 0.6. 这主要是因为该视频序列中目标的影响因素较少. 从图 8 中可以看出, 除 CT, FRAG, MIL, ONNDL 外, 其余算法的 AUC 值都超过了 0.9. 从图 10 中可以看出, 第 114 帧, CT 发生了漂移. 第 172 帧, FRAG 发生了漂移. 这是由于目标区域过小, FRAG 算法中使用的直方图不能很好地描述目标的外观模型.

David3. 该测试视频的难点在于: 目标被短时间大面积的遮挡, 当目标经过树木时被遮挡部分面积远超目标区域面积. 从图 6 中可以看出, PMAPI, FRAG 能够比较准确地跟踪到目标. 从图 7 中的 AUC 值中可以看出, PMAPI 跟踪准确度最高, FRAG 次之. 这主要是因为 PMAPI 和 FRAG 算法都使用了专门处理部分遮挡问题的分块策略, 从结果上看, PMAPI 的处理效果更为优异. 从图 8 中可以看出, PMAPI 在距离误差阈值为 15 像素时跟踪成功率达到了 100%, 说明本文算法一直能准确定位目标位置, 而其他算法在误差为 50 像素时也未达到 100%, 说明其他算法发生了严重的漂移. 从图 10 中可以看出, 第 34 帧, 当目标被电线杆遮挡时, TLD 漂移到了遮挡物上. 第 92 帧, 目标被一棵树大面积遮挡, MAPI, SCM 丢失了目标. 而由于使用了被遮挡的目标更新分类器, 降低了分类器的性能, CT 和 MIL 在第 128 帧时都发生了漂移. 在第 197 帧当目标再次被大面积遮挡时, VTS 丢失了目标, 这是由于 VTS 的外观模型使用整体模板训练 SPCA (稀疏主成分分析) 模型, 因而无法处理目标遭受大面积遮挡的情况. 从整体跟踪结果来看, 只有 PMAPI 完全地跟上了目标, 这也体现了本文提出的分块算法在应对短时大面积遮挡问题时的有效性.

Fish. 该视频中目标为一条鱼的模型, 整个视频中相机一直在不停晃动, 并且右上方的灯光发生了两次变化. 从图 6 中可以看出 PMAPI, MAPI, IVT, VTS, ASLA, TLD, CT, LSST 能够比较准确地跟踪到目标. 由于 CT, MIL 使用对光照敏感的类哈尔特征训练分类器, 在每次光照变化时, 这两种算法的中心误差曲线都会产生波动. 从图 7 中可以看出 PMAPI, MAPI, TLD, ASLA, LSST 的 AUC 值都达到了 0.8 以上. 从图 8 中可以看出, PMAPI, MAPI, IVT, ALSA, LSST 的 AUC 值都达到了 0.9 以上. 从图 10 中可以看出, 第 27 帧时, 由于相机抖动引起了整幅图像模糊, L1APG 发生了漂移. 第 46 帧时, 鱼右上方的灯光发生了光照变化, FRAG 发生了漂移. 第 197 帧中, ONNDL 发生了漂移. 在整个过程中, 本文提出的两种算法都能精确地跟踪到目标位置.

Jumping. 在本测试视频中, 含有运动模糊的帧占了很大比例, 是跟踪的主要难点. 从图 6 中可以看出, PMAPI, MAPI, MIL, TLD, FRAG, LSST 能够比较准确地跟踪到目标位置. 从图 7 的 AUC 值可以看出 PMAPI, MAPI, TLD, FRAG, LSST 跟踪效果最好. 从图 8 中可以看出 LSST 的跟踪精确度最高, 能够有效应对运动模糊情况. 这可以归结为两个原因: (1) LSST 提出的 LSS 距离考虑了异常点的影响. (2) LSST 的更新机制从新的目标中移除异常点来避免模型退化. 从图 10 中可以看出, 第 17 帧由于相机的抖动产生了图像的整体模糊, SCM, ONNDL, CT, IVT, VTS, L1APG 发生了漂移. 其中 IVT 在第 17 帧时跟踪到了目标残影, 这主要是由于基于 ML 的 PCA 跟踪算法在计算候选样本的重构误差时不够精确, 影响了观测模型的计算. 而 MAPI 由于对样本在子空间中的坐标分布加入了先验约束, 能够精确计算候选目标与训练模型之间的差异, 提高了对运动模糊的抗干扰能力. 第 56 帧中, 目标由于跳动产生了运动模糊, ALSA 算法也产生了漂移. 从各种实验数据中可以看出, 本文算法在处理视频中运动模糊和相机抖动时具有鲁棒性.

Bike. 该测试视频中目标的初始框包含了大量的背景区域, 这些背景区域面积约等于目标面积. 由于目标框中背景区域过大, 使得算法学习到了大量背景特征, 随着跟踪的进行, 算法容易漂移到背景. 从图 6 中可以看出, PMAPI, MAPI, IVT, VTS, L1APG, ASLA, SCM, ONNDL 算法能够比较准确地跟踪到目标位置. 从图 7 中可以看出, ONNDL 的跟踪精度最高, 这主要是因为 ONNDL 能够自动检测和排除复杂背景的干扰. 从图 8 中可以看出, PMAPI, MAPI, IVT, VTS, L1APG, ASLA, SCM,

ONNDL 的跟踪准确度都在 0.8 以上. 从图 10 中可以看出, 该测试视频受到背景干扰影响严重, 目标框中含有大量的背景信息. 在目标飞出跳台时 (第 74 帧) FRAG, CT, TLD, LSST 相继发生了漂移. 在第 167 帧, 由于背景的干扰, MIL 也发生了漂移. 从各种实验数据中可以看出, 本文算法能够有效处理背景干扰问题.

Woman. 该测试视频中目标经过第一辆汽车时下半身被车体遮挡, 被遮挡面积约为目标总面积的一半, 遮挡过程持续了将近 70 帧. 如果未有效地处理目标的被遮挡区域, 算法会学习到车辆的特征, 导致跟踪框漂移到车辆上. 从图 6 中可以看出, 只有 PMAPI, SCM 能够比较准确地跟踪到目标. 但是在 560 帧左右, 由于相机的抖动和镜头的突然拉近导致目标产生了模糊和尺度变化的双重影响, PMAPI 算法的跟踪精度降低. 在图 7 中, 只有 PMAPI, SCM 的跟踪结果在 0.6 以上. 可以看出, SCM 使用局部稀疏表示模型能够有效应对跟踪中长时间的部分遮挡问题, 而使用整体目标作为模板的 L1APG 算法在处理遮挡问题时则容易丢失目标. 从图 8 中可以看出, 只有 PMAPI 跟踪准确度达到了 0.9 以上. 从图 10 中可以看出, 当目标未被遮挡时, 所有算法都能够准确地跟踪到目标. 在第 129 帧, 目标穿过了一辆白色车, 下半身被车体遮挡, 除了 PMAPI 和 SCM 外的所有算法都未能跟踪到目标. 在 173, 213, 298, 390 帧中, 目标连续穿过了数辆车, PMAPI, SCM 都能够准确地跟踪到目标, 说明 PMAPI 使用的多子空间策略在处理遮挡影响时具有鲁棒性.

8 结束语

本文首先从最大似然角度分析了基于 PCA 子空间学习的目标跟踪算法, 推导了目标样本投影在子空间中坐标方差的无偏估计, 最后提出了基于 MAP 的 PCA 子空间学习跟踪算法, 解决了过度拟合测试样本的问题, 增强了算法的泛化能力. 针对跟踪被遮挡的目标, 提出了一种分块跟踪策略. 将目标分成多个图像块, 分别训练子空间. 假设各个子空间相互独立, 本文提出融合多个子空间求粒子权值的方法. 根据目标在各个子空间上的重构误差来判断目标的被遮挡部分, 在计算粒子权值和更新子空间时仅使用未被遮挡的分块, 避免了遮挡对跟踪效果的影响. 实验结果表明, 本文提出的算法能够有效地应对目标的遮挡问题, 具有较强的鲁棒性. 在未来的研究中, 将会关注目标在视野中消失后又重现的目标跟踪.

致谢 在此, 我们向对本文的工作给予支持和建议的同行表示感谢.

参考文献

- 1 Yilmaz A, Javed O, Shah M. Object tracking: a survey. *ACM Comput Surv*, 2006, 38: 1-45
- 2 Yang H, Shao L, Zheng F, et al. Recent advances and trends in visual tracking: a review. *Neurocomputing*, 2011, 74: 3823-3831
- 3 Zhang H L, Hu S Q, Yang G S. Video object tracking based on appearance models learning. *J Comput Res Dev*, 2015, 52: 177-190 [张焕龙, 胡士强, 杨国胜. 基于外观模型学习的视频目标跟踪方法综述. *计算机研究与发展*, 2015, 52: 177-190]
- 4 Danelljan M, Khan F S, Felsberg M, et al. Adaptive color attributes for real-time visual tracking. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 2014. 1090-1097
- 5 Lee D Y, Sim J Y, Kim C S. Visual tracking using pertinent patch selection and masking. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 2014. 3486-3493

- 6 Zhang T, Jia K, Xu C, et al. Partial occlusion handling for visual tracking via robust part matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Columbus, 2014. 1258–1265
- 7 Isard M, MacCormick J. BraMBLe: a Bayesian multiple-blob tracker. In: Proceedings of IEEE International Conference on Computer Vision, Vancouver, 2001, 2: 34–41
- 8 Nummiaro K, Koller-Meier E, van Gool L. A color-based particle filter. In: Proceedings of the 1st International Workshop on Generative-Model-Based Vision, Copenhagen, 2002, 1: 53–60
- 9 Babenko B, Yang M H, Belongie S. Visual tracking with online multiple instance learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami, 2009. 983–990
- 10 Zhang K, Zhang L, Yang M H. Real-time compressive tracking. In: Proceedings of European Conference on Computer Vision, Firenze, 2012. 864–877
- 11 Hare S, Saffari A, Torr P H S. Struck: structured output tracking with kernels. In: Proceedings of IEEE International Conference on Computer Vision, Barcelona, 2011. 263–270
- 12 Grabner H, Grabner M, Bischof H. Real-time tracking via on-line boosting. In: Proceedings of the British Machine Vision Conference, Edinburgh, 2006, 1: 1–10
- 13 Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. Pattern Anal Mach Intell, 2012, 34: 1409–1422
- 14 Henriques J F, Caseiro R, Martins P, et al. Exploiting the circulant structure of tracking-by-detection with kernels. In: Proceedings of European Conference on Computer Vision, Firenze, 2012. 702–715
- 15 Ross D A, Lim J, Lin R S, et al. Incremental learning for robust visual tracking. Int J Comput Vision, 2008, 77: 125–141
- 16 Mei X, Ling H. Robust visual tracking using l1 minimization. In: Proceedings of IEEE International Conference on Computer Vision, Kyoto, 2009. 1436–1443
- 17 Jiang M X, Wang H Y, Wang J, et al. Visual object tracking algorithm based on ML and L2-norm. Acta Electron Sin, 2013, 41: 2307–2313 [姜明新, 王洪玉, 王洁, 等. 基于 ML 和 L2 范数的视频目标跟踪算法. 电子学报, 2013, 41: 2307–2313]
- 18 Bao C, Wu Y, Ling H, et al. Real time robust l1 tracker using accelerated proximal gradient approach. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Rhode, 2012. 1830–1837
- 19 Jia X, Lu H, Yang M H. Visual tracking via adaptive structural local sparse appearance model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Rhode, 2012. 1822–1829
- 20 Wang D, Lu H, Yang M H. Online object tracking with sparse prototypes. Image Process, 2013, 22: 314–325
- 21 Yin B C, Shi Y H, Ding W P, et al. Compressive sampling and sparse reconstruction of images/videos. Sci Sin Inform, 2013, 43: 226–242 [尹宝才, 施云惠, 丁文鹏, 等. 图像视频信号的压缩采样与稀疏重建. 中国科学: 信息科学, 2013, 43: 226–242]
- 22 Luo Q, Wei Q, Miao X J. Blocked image compression and reconstruction algorithm based on compressed sensing. Sci Sin Inform, 2014, 44: 1036–1047 [罗琦, 魏倩, 缪昕杰. 基于压缩感知思想的图像分块压缩与重构方法. 中国科学: 信息科学, 2014, 44: 1036–1047]
- 23 Arulampalam M S, Maskell S, Gordon N, et al. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. Signal Process, 2002, 50: 174–188
- 24 Wang T, Gu I Y H, Shi P. Object tracking using incremental 2D-PCA learning and ML estimation. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, 2007, 1: 933–936
- 25 Li G, Liang D, Huang Q, et al. Object tracking using incremental 2D-LDA learning and Bayes inference. In: Proceedings of IEEE International Conference on Image Processing, San Diego, 2008. 1568–1571
- 26 Wang D, Lu H, Chen Y W. Incremental MPCA for color object tracking. In: Proceedings of International Conference on Pattern Recognition, Istanbul, 2010. 1751–1754
- 27 Hu W, Li X, Zhang X, et al. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. Int J Comput Vision, 2011, 91: 303–327
- 28 Levey A, Lindenbaum M. Sequential Karhunen-Loeve basis extraction and its application to images. Image Process, 2000, 9: 1371–1374

- 29 Wang D, Lu H, Yang M H. Least soft-threshold squares tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Portland, 2013. 2371–2378
- 30 Adam A, Rivlin E, Shimshoni I. Robust fragments-based tracking using the integral histogram. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New York, 2006, 1: 798–805
- 31 An G C, Zhang F J, Wang H G, et al. Multi-window target tracking. J Comput Res Dev, 2011, 48: 2023–2030 [安国成, 张凤军, 王宏安, 等. 多窗口目标跟踪算法. 计算机研究与发展, 2011, 48: 2023–2030]
- 32 Yu W S, Tian X H, Hou Z Q, et al. Online visual tracking based on local patch learning. Acta Electron Sin, 2015, 43: 74–78 [余旺盛, 田孝华, 侯志强, 等. 基于局部分块学习的在线视觉跟踪. 电子学报, 2015, 43: 74–78]
- 33 North B, Blake A. Learning dynamical models using expectation-maximisation. In: Proceedings of IEEE International Conference on Computer Vision, Bombay, 1998. 384–389
- 34 Kwon J, Lee K M. Tracking by sampling trackers. In: Proceedings of IEEE International Conference on Computer Vision, Barcelona, 2011. 1195–1202
- 35 Zhong W, Lu H, Yang M H. Robust object tracking via sparsity-based collaborative model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Rhode, 2012. 1838–1845
- 36 Wang N, Wang J, Yeung D Y. Online robust non-negative dictionary learning for visual tracking. In: Proceedings of IEEE International Conference on Computer Vision, Sydney, 2013. 657–664
- 37 Wu Y, Lim J, Yang M H. Online object tracking: a benchmark. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Portland, 2013. 2411–2418

Visual tracking algorithm based on MAP multi-subspace incremental learning

Kang LI, Fazhi HE* & Xiao CHEN

School of computer science, Wuhan University, Wuhan 430000, China

*E-mail: fzhe@whu.edu.cn

Abstract Object tracking is an important area of computer vision. Tracking algorithms based on principal component analysis (PCA) in linear subspace learning assume that the object is generated from the same subspace, and that the error term is subject to a Gaussian distribution with a small variance. However, these algorithms do not consider the distribution of samples in the subspace, which often results in over-fitting when minimizing the reconstruction error. To solve the over-fitting problem, we first analyze PCA subspace learning algorithms from the maximum-likelihood-estimation perspective. We then deduce an unbiased estimation of the variance of the subspace, and finally we propose a new PCA subspace learning algorithm based on the maximum posteriori probability (MAP) method. We propose a local patch model to deal with the occlusion problem. First, we divide the object image into several patches. It is assumed that every patch is generated from an independent subspace. Then, we use the reconstruction error to judge whether a patch is occluded. Finally, we compute the particles' weight and train the new subspace using only the un-occluded patches, which avoids the adverse effect caused by occlusion. Experiments show that the proposed algorithm can effectively solve the influencing factors such as partial occlusion, motion blur and background interference. The proposed approach shows excellent performance in comparison with other state-of-the-art trackers.

Keywords object tracking, principal component analysis, subspace methods, maximal posterior, maximum likelihood estimation

Kang LI was born in 1986. He received a M.S. degree in Computer Science from the Central China Normal University, Wuhan in 2012. Currently, he is a Ph.D. candidate in the School of Computer Science, Wuhan University. His research interests include pattern recognition, image processing and computer vision.

Fazhi HE was born in 1968. He received a Ph.D. degree from Wuhan University of Technology. Now he is a professor in the School of Computer Science, Wuhan University. His research interests include computer graphics, computer-aided design, image processing and computer-supported cooperative work.

Xiao CHEN was born in 1980. He received a M.S. degree in computer science from Three Gorges University in 2010. He is currently a Ph.D. candidate in the School of Computer Science, Wuhan University. His research interests include machine learning, image matting and computer vision.