

doi: 10.7690/bgzd.2015.05.010

## 一种加速的 PCA-L<sub>1</sub> 增量子空间学习跟踪方法

王兵学<sup>1,2</sup>, 康林<sup>3</sup>, 黄自力<sup>1</sup>

(1. 西南技术物理研究所研究生部, 成都 610041; 2. 总装备部重庆军事代表局, 重庆 400042;  
3. 总装备部重庆军事代表局驻绵阳地区军代室, 四川 绵阳 621000)

**摘要:** 为解决 PCA-L<sub>1</sub> 增量子空间学习跟踪方法运算量大、实时性不强的不足, 分析了 PCA-L<sub>1</sub> 的目标跟踪算法的运算瓶颈, 应用、改进了边界误差重采样 (bounded particle resampling, BPR) 算法, 引进了加速近似梯度法 (accelerated proximal gradient, APG), 并测试、比较了该方法与其他经典方法的跟踪效果及运算速度。实验结果表明: 该方法加速效果明显, 能提高跟踪的实时性, 有较强的工程应用价值。

**关键词:** PCA-L<sub>1</sub> 算法; L<sub>1</sub> 范数; APG; 粒子滤波

**中图分类号:** TP302 **文献标志码:** A

## Object Tracking Algorithm Based on Accelerated PCA-L<sub>1</sub> of Incremental Subspace Learning

Wang Bingxue<sup>1,2</sup>, Kang Lin<sup>3</sup>, Huang Zili<sup>1</sup>

(1. Graduate Student Department, Southwest Institute of Technical Physics, Chengdu 610041 China;  
2. PLA Military Representation Bureau of General Equipment Headquarters in Chongqing, Chongqing 400042, China;  
3. PLA Presentation Office in Mianyang District, PLA Representation Bureau of General Equipment Department in Chongqing, Mianyang 621000, China)

**Abstract:** For dealing with the large amount computation and not real time of increment subspace tracking algorithm, analyze computation bottleneck of L<sub>1</sub>-norm maximization target tracking algorithm, use and improve bounded particle re-sampling (BPR) algorithm, introduce accelerated proximal gradient (APG). Test and compare the tracking effect and speed with other classical methods. Experimental results show that, the method accelerate speed effectively, it can improve the real-time tracking, and has strong engineering application value.

**Keywords:** PCA-L<sub>1</sub> algorithm; L<sub>1</sub> norm; APG; particle filter

### 0 引言

近年来, 机器学习被广泛地应用到目标跟踪中, 通过模仿人类的学习过程, 对不断变化的数据进行分析和学习, 更新已有的外观模型, 从而提高跟踪算法的适应性。基于增量子空间学习跟踪算法是一种在线学习的跟踪方法, 是当前研究的重要方法之一, 这类方法的核心是对跟踪对象进行主成分分析, 建立目标子空间的一组特征基, 在跟踪过程中计算新获取的观测与该子空间的距离, 选择距离最小的观测为跟踪区域。

David 等人在文献[1]增量学习的基础上, 提出稳健的增量学习视频跟踪方法 (incremental learning for robust visual tracking, IVT), 该方法能很好地适应跟踪目标的外观变化, 提高了跟踪效果。但该方法能稳定跟踪的前提是目标模型不随时间明显变化, 当形状、亮度等突然变化时, 容易产生漂移。

主成分分析 (principle component analysis, PCA) 是一种最常用的线性降维方法。相关研究者 (Michael 等) 将 PCA 子空间方法推广到目标跟踪之

中。但在跟踪问题中, 目标的姿态、尺度以及光照条件可能同时变化, 多种因素的复杂组合使得数据特征在 PCA 特征空间线性不可分, 导致跟踪性能下降; 并且子空间方法将图像样本拉伸为向量形式进行运算, 丢失了目标图像信息的部分结构信息和邻域关系。KPCA 是 B.Scholkopf 等<sup>[2]</sup>人用核方法对 PCA 算法进行核推广得到的, 增强了对非线性数据的处理能力, 提高了跟踪的稳定性和适应性, 但在抗遮挡和抑制漂移方面还有欠缺。

文献[3-4]中提出基于稀疏表示的子空间跟踪的 L<sub>1</sub> 算法, 有效解决了光照、姿态变化及遮挡情况的目标识别问题。这种算法运算量很大, 降低运算量的方法是对目标模板稀疏采样后构成, 同时, 文献[5-6]对算法作了改进和优化。算法存在的缺点是目标模板无法全面反应原始数据信息, 跟踪不稳定, 特别是在跟踪对象发生比较大的表观变化时, 跟踪漂移增大且容易丢失。解决这一问题的思路之一就是选用更为鲁棒的数据描述方法代替稀疏采样模板。

收稿日期: 2015-02-12; 修回日期: 2015-03-24

基金项目: 总装预研 (402030203) 基金资助项目

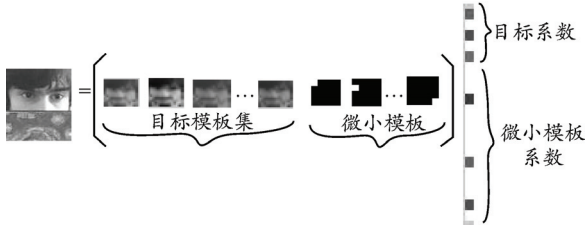
作者简介: 王兵学 (1975—), 男, 甘肃人, 博士, 工程师, 从事运动目标检测与跟踪研究。

### 1 基于 PCA-L<sub>1</sub> 的目标跟踪算法

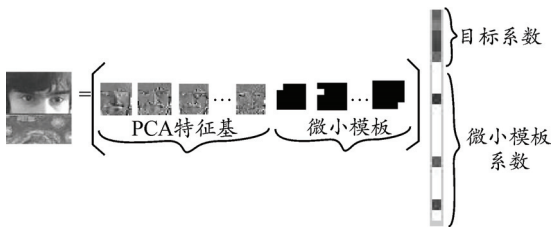
研究表明：用 PCA 特征基构造目标模板集，在降低数据量的同时，可以在最大程度上保存原图像的信息。因此有学者<sup>[7]</sup>提出将 PCA 和 L<sub>1</sub> 算法结合，提出了 PCA-L<sub>1</sub> 算法。PCA-L<sub>1</sub> 跟踪方法与 PCA、L<sub>1</sub> 跟踪方法的区别见图 1。L<sub>1</sub> 跟踪方法的目标模板和微小模板构成如图 1(b)所示，由于 L<sub>1</sub> 跟踪方法中的目标模板是相互关联的，其表达系数是稀疏的，而微小模板用于代表图像中的噪声和干扰部分，系数也是稀疏的；图 1(c)是 PCA-L<sub>1</sub> 方法，由于 PCA 特征基向量是正交的，故系数是非稀疏的，而微小模板系数仍是稀疏的。从图 1 可以看出：该方法一方面克服了 L<sub>1</sub> 跟踪方法对目标描述不够丰富的缺点，另一方面也避免了 PCA 完全不考虑噪声和遮挡干扰的缺点，并且可以通过 PCA 增量学习方法，对新样本的不断学习。



(a) PCA 方法示意图



(b) L<sub>1</sub> 方法示意图



(c) PCA-L<sub>1</sub> 方法示意图

图 1 3 种跟踪方法的原理对比

该算法的核心思想：用粒子滤波采样产生若干个候选目标，用一组 PCA 特征基和微小模板作为基函数对它们进行线性表示。当某一个候选目标为跟踪结果时，由基函数中的 PCA 特征基线性表示的重构误差最小，因此可以假设线性表示的系数是稀疏的，并利用 L<sub>1</sub> 范式最小化求解这些系数。采用在线学习的方式，即在每一时刻跟踪完成后，利用跟踪结果与目标模板之间的变化作为训练样本对基函

数进行更新，使得基函数能准确的反映目标外观的变化。

该算法具有稳定精确的跟踪效果，文献[7]对光照突变、遮挡和跳动等场景下的 12 个运动目标进行了跟踪测试。实验结果表明，该方法较 IVT、L<sub>1</sub>、PN、VTD、MIL 和 Frag 等算法在跟踪稳定性、精度方面都更优异。文献[8]也对该种算法进行了试验测试，表明在目标遮挡、视角变化等场景下具有较强的鲁棒性和抑制模型漂移的能力。虽然 PCA-L<sub>1</sub> 算法在抗遮挡和抑制漂移方面均有很好的表现，但与 IVT、KPCA 等算法相比，该算法运算速度慢，很难进行实时跟踪，需要对其进行加速处理。

### 2 PCA-L<sub>1</sub> 算法的加速求解方法

#### 1) PCA-L<sub>1</sub> 跟踪算法运算的瓶颈。

PCA-L<sub>1</sub> 是通过求解下面的最小化问题来求取任意一个候选目标  $y_i^i$  的表达系数：

$$\min_a \frac{1}{2} \|y_i^i - Ba\|_2^2 + \lambda \|a\|_1 + \frac{d}{2} \|a_T\|_2^2, \quad \text{s.t. } a \geq 0 \quad (1)$$

其中： $B=[U, I]$ ， $U$  是 PCA 分解的特征基、 $I$  是微小模板， $a_i^i = [a_T^i; a_I^i]$ ； $a_T$  是目标系数； $a_I$  是微小模板系数； $a$  是特征系数；参数  $d$  是控制微小模板能量大小的参数，它用于控制微小模板系数的贡献； $\lambda$  是协方差矩阵的特征值。

PCA-L<sub>1</sub> 跟踪算法引入了正、负单位模板，大大增加了问题的维数，使得运算效率降低。同时在关键过程求解出公式 (1) 中的系数  $a$  时，有 2 个运算量比较大的地方。一是在粒子滤波框架下，需要对大量粒子的目标状态进行预测，而每一个粒子都需要解  $l_1$  范数最小化问题；二是  $l_1$  范数最小化问题即公式 (1) 的解本身也是一个比较耗时的过程。因此，PCA-L<sub>1</sub> 跟踪算法运算量大、实时性较差，需要通过一定的优化措施提高求解速度。

#### 2) BPR 算法的应用与改进。

文献[6]对 L<sub>1</sub> 跟踪方法做出了一个重要的优化措施，即基于最小误差边界策略的边界误差重采样 (bounded particle resampling, BRP) 算法，有效提高了 L<sub>1</sub> 跟踪方法的计算效率。

最小误差边界的提出是基于以下观察：

$$\|T_i a - y\|_2^2 \geq \|T_i \hat{a} - y\|_2^2, \quad \forall a \in R^N \quad (2)$$

其中

$$\hat{a} = \arg \min_a \|T_i a - y\|_2^2 \quad (3)$$

因此，对任意一个采样粒子  $x_t^i$ ，它的观测概率符合以下边界约束：

$$p(z_t | x_t^i) \leq \frac{1}{F} \exp\{-\mathbf{a}^\top T_t \hat{\mathbf{a}} - y_t^i\}_2^\Delta = q(z_t | x_t^i) \quad (4)$$

其中  $q(y_t^i | x_t^i)$  是在状态  $x_t^i$  下能得到的观测概率的上界。基于以上分析，可以得知如果有  $q(z_t | x_t^i) < \frac{1}{2N} \sum_{j=1}^{i-1} p(z_t | x_t^j)$ ，则采样粒子  $x_t^i$  将不会出现在后续的重采样粒子集中，也就是说，对于采样粒子  $x_t^i$  可以不再进行后续观测概率的计算。这样一来，便可大大减少参与计算  $l_1$  范数最小的粒子个数。

有了最小误差边界的约束，BPR 算法得以提出。它在不牺牲粒子重采样精度的前提下显著提高了运算速度，该算法把计算每个粒子观测概率的过程分成 2 步：首先只是简单的将样本投影到样本模板空间进行重构，对计算出的观测概率进行排序，只选择具有较大观测概率的粒子分布，进入下一阶段的计算；第 2 步则是通过  $l_1$  范数最小问题精确求解重构误差。

加速算法中也借鉴了 BPR 方法的思路，只是在第一阶段的计算方法有所不同。在此的计算方法：用  $nSample$  个粒子采样状态对应的候选目标图像和 PCA 的特征基对跟踪目标进行重构，将重构误差排序，选择少量的粒子继续计算。通过试验，选择 200 个粒子即可满足跟踪要求，较其他文献中的 600 个粒子的运算量大大降低。通过最小误差边界约束可以大大减少后续参与运算的粒子个数，有效解决了 PCA-L<sub>1</sub> 跟踪算法的第 1 个计算瓶颈。

3) 加速近似梯度法方法的应用。

对于第 2 个计算瓶颈，可以用加速近似梯度法 (accelerated proximal gradient, APG) 快速计算方法加快求解参数  $\mathbf{a}$  的过程。2008 年，有学者<sup>[9]</sup>提出了 APG 方法，它是一种利用 Nesterov 技巧的一阶算法，其收敛速度很有竞争力。该方法最早用于解决如下的无约束最小化问题<sup>[10]</sup>：

$$\min F(\mathbf{a}) + G(\mathbf{a}) \quad (5)$$

式中： $F(\mathbf{a})$  是一个具有 Lipschitz 连续梯度的可微凸函数； $G(\mathbf{a})$  是一个非光滑的凸函数。由于 APG 方法适用于求解无约束最小化问题，而公式 (1) 是一个有约束的最小化问题，因此 APG 快速计算方法不能直接用于式 (1) 的计算。文献[5]中对公式 (1) 进行修改，使其转变为一个无约束的最小化问题，具体

方法是在公式 (1) 中添加了一项  $1_{R^+}(\mathbf{a})$ ，其定义为

$$1_{R^+}(\mathbf{a}) = \begin{cases} 0, & \mathbf{a} > 0 \\ +\infty, & \text{otherwise} \end{cases} \quad (6)$$

公式 (1) 则可以改写为

$$\arg \min_{\mathbf{a}} \frac{1}{2} \|y_t^i - B\mathbf{a}\|_2^2 + \lambda \mathbf{1}_T^\top \mathbf{a}_T + \|\mathbf{a}_I\|_1 + \frac{d_t}{2} \|\mathbf{a}_I\|_2^2 + 1_{R^+}(\mathbf{a}_T) \quad (7)$$

继续改写为

$$F(\mathbf{a}) = \frac{1}{2} \|y - B\mathbf{a}\|_2^2 + \lambda \mathbf{1}_T^\top \mathbf{a}_T + \frac{d_t}{2} \|\mathbf{a}_I\|_2^2$$

$$G(\mathbf{a}) = \|\mathbf{a}_I\|_1 + 1_{R^+}(\mathbf{a}_T)$$

公式 (6) 的最小化问题有如下解法：

$$\mathbf{a}_{k+1}|_T = \max(0, \mathbf{g}_{k+1}|_T) \quad (8)$$

$$\mathbf{a}_{k+1}|_I = S_{\lambda/L}(\mathbf{g}_{k+1}|_I)$$

其中： $\mathbf{g}_{k+1} = \beta_{k+1} - \frac{\nabla F(\beta_{k+1})}{L}$ ； $S$  是一个软阈值算子：

$$S_\lambda(x) = \text{sign}(x) \max(|x| - \lambda, 0) \quad (9)$$

通过下面的迭代过程可以求出系数  $\mathbf{a}$ ：

1) 设  $\mathbf{a}_0 = \mathbf{a}_{-1} = \mathbf{0} \in R^N$ ，以及  $t_0 = t_{-1} = 1$ ；

2) 当  $k = 0, 1, \dots$  时进行迭代计算，直至计算结果收敛：

$$\begin{cases} \gamma_{k+1} := \mathbf{a}_k + \frac{t_{k-1} - 1}{t_k} (\mathbf{a}_k - \mathbf{a}_{k-1}); \\ \mathbf{g}_{k+1}|_T := \gamma_{k+1}|_T - (B^\top (B\gamma_{k+1} - y))|_T / L - \lambda \mathbf{1}_T \\ \mathbf{g}_{k+1}|_I := \gamma_{k+1}|_I - (B^\top (B\gamma_{k+1} - y))|_I / L - d\gamma_{k+1}|_I / L \\ \mathbf{a}_{k+1}|_T := \max(0, \mathbf{g}_{k+1}|_T) \\ \mathbf{a}_{k+1}|_I := \max(0, \mathbf{g}_{k+1}|_I) \\ t_{k+1} := (1 + \sqrt{1 + 4t_k^2}) / 2 \end{cases} \quad (10)$$

这样就可以用 APG 对式 (6) 进行快速计算。

### 3 算法小结

加速 PCA-L<sub>1</sub> 跟踪算法首先要建立起目标的观测模型，用粒子滤波对目标的状态进行预测，整个跟踪算法的步骤可以描述如下：

输入：视频图像  $I_t$ ，上一帧目标位置  $X_{t-1}$ 。

输出：当前帧跟踪到的目标位置  $X_t$ 。

1) 目标初始化。假设在初始时刻  $t_0$ ，已知目标模板及初始运动参数，初始化粒子集  $\{X_0^n = x_0, n = 1, 2, \dots, N\}$ ，设置各粒子的权值为  $1/N$ ，初始化目标特征空间特征向量  $\mathbf{U}_\Phi$  为空；

2) 状态转移。进入下一帧图像后,在动态模型下,以上一帧的 6 维目标状态(仿射变换参数)为均值,以预定的仿射变换参数为基准,用高斯分布生成  $t$  时刻的粒子集  $\{X_t^n\}$ ;

3) 对于目标模板,通过降维获得 PCA 特征集;

4) 计算粒子的观测概率并进行排序,保存观测概率较大的前 200 个粒子进入下一阶段的计算;

5) 获取保存的每个粒子的目标位置  $X_t^i$  所对应的候选区域,归一化并拉伸为向量,记做  $y_t^i$ ;

6) 对于每个候选的目标  $y_t^i$ ,通过过完备基  $B$  构造  $L_1$  最小范数问题,通过求解公式 (1) 得到目标模板相关系数  $a_T$  和微小模板相关系数  $a_I$ ,即得系数向量  $a$ ;

7) 计算使用过完备基  $B$  对每个候选目标  $y_t^i$  的匹配效果为  $d_t^i = \|y_t^i - Ba\|_2$ ;

8) 根据  $d_t^i$  更新粒子权值  $w_t^i$ ,使得  $\sum_i w_t^i = 1$ ;

9) 获取当前帧目标位置  $X_t = \sum_i X_t^i w_t^i$ ;

10) 粒子重采样;

11) 判断是否更新特征空间。当新的观测样本到达预定数量时,对新样本进行增量学习以更新特

征空间,转至步骤 2)。

#### 4 实验结果与分析

为验证本算法的有效性,分别对几种典型复杂场景的视频目标进行跟踪测试,并与近年来提出的 2 种比较具有代表性的方法: IVT 和  $L_1$  在跟踪效果和时间上进行了比较。实验在 Matlab R2010a 环境下,以 Pentium@Dual-Core 3.50 GHz CPU, 3.50 GB 内存的台式机为工具。跟踪算法的模板大小是  $32 \times 32$ ,特征基个数是 15,特征基更新步长是 5,  $\lambda$  设为 0.05,原 PCA- $L_1$  的粒子数为 600,加速 PCA- $L_1$  的粒子数取前 200 个。实验中点线框为  $L_1$  跟踪方法结果、虚线框为 IVT 跟踪方法结果,其中实线框为文中算法。

图 2 展示了对视频序列“shaking”的跟踪结果,该视频中跟踪对象为一乐手的头部,整个跟踪过程中随着乐手不断的抬头、低头和身体姿态改变,目标出现了显著的外观变化,在第 60 帧左右还出现了强光干扰,目标几乎被淹没。图 2 中第 20 帧时目标有明显运动和强光干扰,IVT 方法和  $L_1$  跟踪器出现了漂移;第 60 帧文中方法抵抗住了剧烈的光照变化,第 158、248、358 帧目标人物的姿态迥异,可以看出文中方法一直保持了对目标区域的跟踪。



图 2 3 种方法对 shaking 视频跟踪效果对比

图 3 是对视频序列“singer”的跟踪结果，该视频中的难点是目标出现了非常明显的尺度变化，同时还伴随着目标位置移动、目标姿态变换和剧烈的光照变换。如在第 90 帧时图像整体变亮，目标变小且与初始位置相比有较大位移，此时 3 种跟踪方法均能保持对目标的稳定跟踪，之后目标持续向右移

动并继续变小，到第 222 帧时本章方法和 L<sub>1</sub> 跟踪方法仍然跟踪的比较好，但 IVT 方法在目标尺度和位置计算上有了较大的偏差，虽然目标还在跟踪框内，但跟踪框变大，表明跟踪向背景漂移。到了第 321 帧，IVT 方法跟踪误差变大，文中方法和 L<sub>1</sub> 跟踪方法跟踪稳定。



图 3 3 种方法对 singer 视频跟踪效果对比

图 4 展示了对视频序列‘Deer’的跟踪效果，视频中目标处于快速奔跑过程中，背景和目标姿态变化很快。如在第 8、29 帧，目标在短时间内出现

大幅度位移，第 41 帧目标因运动导致图像模糊，第 55 帧出现了相似目标干扰等复杂情况。文中算法与 IVT 方法、L<sub>1</sub> 跟踪方法均实现了稳定跟踪。

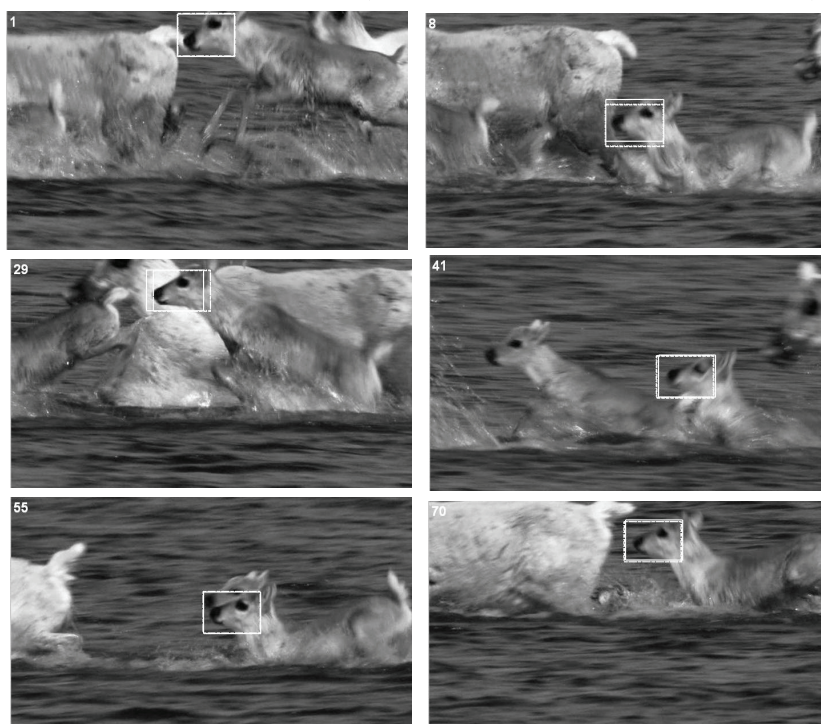


图 4 3 种方法对 Deer 视频跟踪效果对比

表 1 列出了 3 种算法对文中测试视频的平均处理时间，文中算法的运算效率比 L<sub>1</sub> 和 IVT 跟踪方

法都高。