

• 通信与信息工程 •

基于自适应同时稀疏表示的鲁棒性目标追踪

李厚彪, 樊庆宇, 耿广磊

(电子科技大学数学科学学院 成都 611731)

【摘要】综合考虑高斯噪声和拉普拉斯噪声, 并通过拉普拉斯噪声的能量大小自适应的选择稀疏模型, 该文提出了基于同时稀疏表示的自适应追踪算法。该算法可以更好的解决目标遮挡、姿势改变、光照变化和背景混杂等追踪问题, 且具有更强的鲁棒性。其次提出一种基于子空间学习和无监督学习(K-means)相结合的模板更新方法, 该方法一方面可以及时有效地反应目标的状态, 另一方面也可以避免模板更新过快而引入较大的误差。然后, 利用LASSO算法对该模型做了进一步的改进, 并将目前较好的9种追踪算法与该文提出的算法进行比较, 实验结果表明该算法在鲁棒性、精确性和实时性方面都得到了较好的改善。

关键词 拉普拉斯噪声; 鲁棒性; 同时稀疏表示; 模板更新; 无监督学习

中图分类号 TN911.73 **文献标志码** A **doi**:10.3969/j.issn.1001-0548.2018.01.001

Robust Visual Tracking Based on Adaptive Simultaneous Sparse Representation

LI Hou-biao, FAN Qing-yu, and GENG Guang-lei

(School of Mathematics Sciences, University of Electronic Science and Technology of China Chengdu 611731)

Abstract Sparse representation has already been successfully applied to the visual tracking, but it also faces with unstable factors. In this paper, we propose an adaptive tracking algorithm based on simultaneous sparse representation which considers the Gaussian noise and Laplace noise. The sparse model is chosen adaptively according to the energy of Laplace noise. The proposed algorithm can better solve the problem of targets such as occlusion, pose change, illumination variation and background clutter, and the algorithm has better robustness. Secondly, the template update method based on the subspace learning and unsupervised learning (K-means) is given to respond the target state in a timely and effective manner and to avoid the template update too fast and introduce significant error. Then, the LASSO algorithm is used to further improve the model. Finally, comparing the current nine state-of-the-art tracking algorithms with the proposed algorithm in this paper, the experimental results illustrate that the proposed algorithm have the better performance in terms of robustness, accuracy and real time.

Key words Laplace noise; robustness; simultaneous sparse representation; template update; unsupervised learning

目标追踪在计算机视觉领域占据着重要的地位。目标追踪的主要任务包括: 感兴趣运动目标的检测、视频帧到帧之间的连续追踪和追踪目标的行为分析等^[1-2]。

当前目标追踪所面临的一些挑战主要包括光照变化、背景混杂、部分遮挡、完全遮挡以及它们的混合。为了解决这些问题, 各种不同的追踪算法相继被提出来, 如在线学习方法(online learning)和基于稀疏表示(sparse representation)的追踪算法。在线学习方法大致又可分为: 生成方法(generative

approaches, GA) 和判别方法(discriminative approaches, DA)。GA是一种搜索与追踪目标最相似的区域方法, 如文献[3]使用积分直方图的健壮片段跟踪, 减弱了部分遮挡和姿势变化等因素对追踪效果的影响, 有效降低了计算代价; 另外, 为更好地实现鲁棒性追踪, 基于增量学习的目标追踪算法^[4]和基于核的目标追踪算法^[5]相继被提出来。尽管生成方法在一定程度上能实现追踪, 但是不能有效解决长时间部分遮挡、完全遮挡、目标外形剧烈变化等追踪问题。DA可看作是一种二分类问题, 主要利

收稿日期: 2016-5-25; 修回日期: 2017-11-01

基金项目: 国家自然科学基金(51175443); 中央高校基本科研业务费专项资金(ZYGX2016J131, ZYGX2016J138); 四川省科技支撑计划(2015GZX0002)

作者简介: 李厚彪(1976-), 副教授, 主要从事目标追踪与图像处理方面的研究。

用已知的训练样本训练出一个分类器,用于判别目标和背景。文献[6]提出在线多实例学习的目标追踪算法,该方法不同于传统的监督学习方法,主要区别为用于训练时的阳性样本(positive sample)数量大于阴性样本(negative sample)数量;文献[7]提出超像素追踪方法,该法使用超像素特征的结构信息能更好判别目标和背景,在一定程度上适应姿势改变、运动模糊及短时间的部分遮挡等情况,但该方法的计算代价较高且只用了简单的HIS颜色特征并未考虑更好的颜色特征;文献[8]提出基于核空间的稠密采样追踪算法,使用循环矩阵和快速傅里叶变换在核空间里实现了快速学习,大大提高了追踪系统的实时性。

文献[9]提出稀疏表示在人脸识别中的应用(sparse representation based classification, SRC),对噪声污染和人脸部分遮挡获得较好的识别率,相比之前提出的SVM、KNN、PCA和LDA等线性分类器,不仅表现出更优的识别效果,且有更强的抗干扰能力。受其启发,文献[10]提出稀疏表示在目标追踪方面的应用,此后不断得到发展,如稀疏表示追踪^[10-15](L₁ Tracker)、联合表示追踪^[16](L₂ Tracker)、同时稀疏表示追踪^[17](L_{p,q} Tracker)、核稀疏表示追踪^[18-19](KSR Tracker)和混合稀疏表示追踪^[20-21]等。

本文提出了同时稀疏表示的自适应追踪算法,该算法采用子空间学习(subspace learning)和无监督学习(K-means)相结合的模板更新方法,避免了模板更新太快而引入较大的误差;另外算法使用同时稀疏表示来刻画粒子之间的关系,克服了传统稀疏表示假设粒子之间是相互独立的缺陷;最后根据噪声(如遮挡,光照变化等)能量的大小自适应的选择模型。大量的实验结果表明该算法可实现鲁棒性追踪。

1 自适应同时稀疏追踪模型

1.1 同时稀疏L_{p,q}模型

目前,粒子滤波方法已被成功应用到目标追踪中:若追踪过程中第($t-1$)帧的追踪目标状态为粒子 q_{t-1} ,则第 t 帧的追踪目标可能的状态记为 $p(q_t | q_{t-1})$,且第 t 帧的观测为 $p(y_t | q_t)$,其中 y_t 为第 t 帧的观测值。稀疏表示追踪给出了 $p(y_t | q_t)$ 的具体形式,并根据最小重构误差得到目标的追踪结果 y_t 。

对于一个给定目标模板的图像集合 $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_n] \in \mathbf{R}^{d \times n}$ (d 表示图像的维数, n 表示模板基向量的个数), \mathbf{T} 的每一列都是通过零均值化后得到的向量,因此对于一个候选目标 \mathbf{y} 可以近似通过模板的线性组合表示为:

$$\mathbf{y} = \mathbf{T}\mathbf{z} + \mathbf{e} = [\mathbf{T}, \mathbf{I}] \begin{pmatrix} \mathbf{z} \\ \mathbf{e} \end{pmatrix} \quad (1)$$

式中, \mathbf{e} 表示高斯噪声; \mathbf{I} 表示琐碎模板。文献[15,22]提出的稀疏表示追踪模型为:

$$\begin{aligned} \min_{\mathbf{x}} \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \\ \text{s.t. } \mathbf{x} \geq \mathbf{0} \end{aligned} \quad (2)$$

式中, $\mathbf{D} = [\mathbf{T}, \mathbf{I}]$ 表示追踪模板; $\mathbf{x} = [\mathbf{z}, \mathbf{e}]^T$; λ 表示正则参数,用来平衡保真项和正则项。若考虑粒子之间的关系,则可给出稀疏追踪模型^[17]:

$$\min_{\mathbf{X}} \|\mathbf{D}\mathbf{X} - \mathbf{Y}\|_{\text{F}}^2 + \lambda \|\mathbf{X}\|_{p,q} \quad (3)$$

式中, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m]$ 表示所有候选目标; $\|\mathbf{X}\|_{p,q}^q = \sum_{i=1}^{n+d} (\|\mathbf{X}_i\|_p)^q$ ($p \geq 1, q \leq 1$); $\|\cdot\|_p$ 是L_p范数;正则项 $\|\mathbf{X}\|_{p,q}$ 能更好地提取粒子之间的相似性并能有效去除模板的冗余信息。

尽管上述模型在一定程度上解决了部分遮挡、光照变化、姿势改变和背景混杂等影响,但太过简单地考虑噪声的分布情况,因此面对一些复杂的噪声分布情况可能会出现跟踪失败。为此下面假设噪声服从高斯拉普拉斯分布,即:

$$\mathbf{Y} = \mathbf{T}\mathbf{Z} + \mathbf{S} + \mathbf{E} \quad (4)$$

式中, \mathbf{S} 表示拉普拉斯噪声; \mathbf{E} 表示高斯噪声。给出同时稀疏追踪模型:

$$\begin{aligned} \min_{\mathbf{X}, \mathbf{S}} \|\mathbf{Y} - \mathbf{D}\mathbf{X} - \mathbf{S}\|_{\text{F}}^2 + \lambda_1 \|\mathbf{X}\|_{1,1} + \lambda_2 \|\mathbf{S}\|_{1,1} \\ \text{s.t. } \mathbf{X} \geq \mathbf{0} \end{aligned} \quad (5)$$

式中, $\mathbf{X} = [\mathbf{Z}, \mathbf{E}]^T$ 。但对于一个给定的视频序列,目标并不总处于被污染状态,设已追踪到目标对应的拉普拉斯噪声为 \mathbf{S}_t (表示 \mathbf{S} 的第 t 列),因此可给出自适应的同时稀疏追踪模型为:

1) 噪声 $\|\mathbf{S}_t\|_2 \leq \tau$ 时,模型变为:

$$\begin{aligned} \min_{\mathbf{X}, \mathbf{S}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_{\text{F}}^2 + \lambda_1 \|\mathbf{X}\|_{1,1} \\ \text{s.t. } \mathbf{X} \geq \mathbf{0} \end{aligned} \quad (6)$$

2) 噪声 $\|\mathbf{S}_t\|_2 > \tau$ 则模型变为:

$$\begin{aligned} \min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X} - \mathbf{S}\|_{\text{F}}^2 + \lambda_1 \|\mathbf{X}\|_{1,1} + \lambda_2 \|\mathbf{S}\|_{1,1} \\ \text{s.t. } \mathbf{X} \geq \mathbf{0} \end{aligned} \quad (7)$$

式中, τ 为给定噪声能量的阈值; λ_1 和 λ_2 为模型的正则参数,分别用来控制表示系数 \mathbf{X} 的稀疏程度以及噪声 \mathbf{S} 的能量大小。

模型求解:式(6)和式(7)的目标函数是一个凸优化。使用交替方向乘子方法(alternating direction method multipliers, ADMM)求解优化问题(7)如下:

首先,将约束问题变为无约束问题:

$$\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X} - \mathbf{S}\|_{\text{F}}^2 + \lambda_1 \|\mathbf{X}\|_{\text{L}_1} + \lambda_2 \|\mathbf{S}\|_{\text{L}_1} + \tau_+(\mathbf{X}) \quad (8)$$

式中, $\tau_+(\mathbf{X}) = \sum_{i=1}^d \tau_+(x_i)$ 是一个指示函数。因此优化问题(7)有如下的等价形式:

$$\min_{\mathbf{X}, \mathbf{S}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3} \frac{1}{2} \|\mathbf{Y} - \mathbf{V}_1 - \mathbf{S}\|_{\text{F}}^2 + \lambda_1 \|\mathbf{V}_2\|_{\text{L}_1} + \lambda_2 \|\mathbf{S}\|_{\text{L}_1} + \tau_+(\mathbf{V}_3) \quad (9)$$

s.t. $\mathbf{D}\mathbf{X} = \mathbf{V}_1, \mathbf{X} = \mathbf{V}_2, \mathbf{X} = \mathbf{V}_3$

式中, $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3$ 为对偶变量, 式(9)进一步优化为:

$$\min_{\mathbf{V}, \mathbf{X}, \mathbf{S}} g(\mathbf{V}, \mathbf{X}, \mathbf{S}) \quad (10)$$

s.t. $\mathbf{G}\hat{\mathbf{X}} + \mathbf{B}\mathbf{V} = 0$

式中,

$$g(\mathbf{V}, \mathbf{X}, \mathbf{S}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{V}_1 - \mathbf{S}\|_{\text{F}}^2 + \lambda_1 \|\mathbf{V}_2\|_{\text{L}_1} + \lambda_2 \|\mathbf{S}\|_{\text{L}_1} + \tau_+(\mathbf{V}_3)$$

$$\mathbf{G} = \begin{pmatrix} \mathbf{D} & & \\ & \mathbf{I} & \\ & & \mathbf{I} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} -\mathbf{I} & 0 & 0 \\ 0 & -\mathbf{I} & 0 \\ 0 & 0 & -\mathbf{I} \end{pmatrix}, \hat{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \mathbf{X} \\ \mathbf{X} \end{pmatrix}, \mathbf{V} = \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \end{pmatrix}$$

式(10)的增广拉格朗日函数为:

$$L(\mathbf{X}, \mathbf{S}, \mathbf{U}, \mathbf{V}) = g(\mathbf{X}, \mathbf{S}, \mathbf{V}) + \frac{\beta}{2} \|\mathbf{G}\mathbf{X} + \mathbf{B}\mathbf{V} - \mathbf{U}\|_{\text{F}}^2 \quad (11)$$

式中, β 表示拉格朗日乘子; $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]^{\text{T}}$ 。式(11)可分解为3个子优化问题:

$$\mathbf{V}_1^* = \arg \min_{\mathbf{V}_1} \frac{1}{2} \|\mathbf{Y} - \mathbf{V}_1 - \mathbf{S}\|_{\text{F}}^2 + \frac{\beta}{2} \|\mathbf{D}\mathbf{X} - \mathbf{V}_1 - \mathbf{U}_1\|_{\text{F}}^2$$

$$\mathbf{V}_2^* = \arg \min_{\mathbf{V}_2} \lambda_1 \|\mathbf{V}_2\|_{\text{L}_1} + \frac{\beta}{2} \|\mathbf{X} - \mathbf{V}_2 - \mathbf{U}_2\|_{\text{F}}^2$$

$$\mathbf{V}_3^* = \arg \min_{\mathbf{V}_3} \tau_+(\mathbf{V}_3) + \frac{\beta}{2} \|\mathbf{X} - \mathbf{V}_3 - \mathbf{U}_3\|_{\text{F}}^2$$

$$\mathbf{S}^* = \arg \min_{\mathbf{S}} \frac{1}{2} \|\mathbf{Y} - \mathbf{V}_1 - \mathbf{S}\|_{\text{F}}^2 + \lambda_2 \|\mathbf{S}\|_{\text{L}_1}$$

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \frac{\beta}{2} (\|\mathbf{D}\mathbf{X} - \mathbf{V}_1 - \mathbf{U}_1\|_{\text{F}}^2 + \|\mathbf{X} - \mathbf{V}_2 - \mathbf{U}_2\|_{\text{F}}^2 + \|\mathbf{X} - \mathbf{V}_3 - \mathbf{U}_3\|_{\text{F}}^2)$$

因此根据极值原理, 只需要对上述子问题求一阶导数, 可获得式(11)的最优解:

$$\mathbf{V}_1^* = [\beta(\mathbf{D}\mathbf{X} - \mathbf{U}_1) + (\mathbf{Y} - \mathbf{S})] / (1 + \beta)$$

$$\mathbf{V}_2^* = \text{shrink}(\mathbf{X} - \mathbf{U}_2, \lambda_1 / \beta)$$

$$\mathbf{V}_3^* = \max(0, \mathbf{X} - \mathbf{U}_3)$$

$$\mathbf{S}^* = \text{shrink}(\mathbf{Y} - \mathbf{V}_1, \lambda_2)$$

$$\mathbf{X}^* = (\mathbf{D}^{\text{T}}\mathbf{D} + 2\mathbf{I})^{-1} [\mathbf{D}^{\text{T}}(\mathbf{V}_1 + \mathbf{U}_1) + \mathbf{V}_2 + \mathbf{U}_2 + \mathbf{V}_3 + \mathbf{U}_3]$$

同理, 对于式(6)得到以下解:

$$\mathbf{V}_1^* = [\beta(\mathbf{D}\mathbf{X} - \mathbf{U}_1) + \mathbf{Y}] / (1 + \beta)$$

$$\mathbf{V}_2^* = \text{shrink}(\mathbf{X} - \mathbf{U}_2, \lambda_1 / \beta)$$

$$\mathbf{V}_3^* = \max(0, \mathbf{X} - \mathbf{U}_3)$$

$$\mathbf{X}^* = (\mathbf{D}^{\text{T}}\mathbf{D} + 2\mathbf{I})^{-1} [\mathbf{D}^{\text{T}}(\mathbf{V}_1 + \mathbf{U}_1) + \mathbf{V}_2 + \mathbf{U}_2 + \mathbf{V}_3 + \mathbf{U}_3]$$

这样通过对子问题的分析与求解, 获得了式(6)和式(7)解的一般形式, 数值求解见算法1和2。

算法1: 问题(6)的ADMM求解算法

初始化: $k = 0$, $\mathbf{X}^{(0)}$, \mathbf{Y} , 正则参数 λ_1 , λ_2 , 对偶变量 $\mathbf{V}_1^{(0)}$, $\mathbf{V}_2^{(0)}$, $\mathbf{V}_3^{(0)}$, 拉格朗日乘子 $\mathbf{U}_1^{(0)}$, $\mathbf{U}_2^{(0)}$, $\mathbf{U}_3^{(0)}$, 收敛误差 ε , 惩罚因子 β 。

1) 迭代

$$2) \mathbf{V}_1^{(k+1)} \leftarrow [\beta(\mathbf{D}\mathbf{X}^{(k)} - \mathbf{U}_1^{(k)}) + \mathbf{Y}] / (1 + \beta)$$

$$3) \mathbf{V}_2^{(k+1)} \leftarrow \text{shrink}(\mathbf{X}^{(k)} - \mathbf{U}_2^{(k)}, \lambda_1 / \beta)$$

$$4) \mathbf{V}_3^{(k+1)} \leftarrow \max(0, \mathbf{X}^{(k)} - \mathbf{U}_3^{(k)})$$

$$5) \mathbf{U}_1^{(k+1)} \leftarrow \mathbf{U}_1^{(k)} - (\mathbf{D}\mathbf{X}^{(k+1)} - \mathbf{V}_1^{(k+1)})$$

$$6) \mathbf{U}_2^{(k+1)} \leftarrow \mathbf{U}_2^{(k)} - (\mathbf{X}^{(k+1)} - \mathbf{V}_2^{(k+1)})$$

$$7) \mathbf{U}_3^{(k+1)} \leftarrow \mathbf{U}_3^{(k)} - (\mathbf{X}^{(k+1)} - \mathbf{V}_3^{(k+1)})$$

$$8) \mathbf{X}^{(k+1)} \leftarrow (\mathbf{D}^{\text{T}}\mathbf{D} + 2\mathbf{I})^{-1} [\mathbf{D}^{\text{T}}(\mathbf{V}_1^{(k+1)} + \mathbf{U}_1^{(k+1)} + \mathbf{V}_2^{(k+1)} + \mathbf{U}_2^{(k+1)} + \mathbf{V}_3^{(k+1)} + \mathbf{U}_3^{(k+1)})]$$

9) 直到 $\|\mathbf{G}\mathbf{X}^{(k+1)} + \mathbf{B}\mathbf{V}^{(k+1)}\| \leq \varepsilon$

输出: 最优解 $\mathbf{X}^{(k+1)}$

算法2: 问题(7)的ADMM求解算法

初始化: $k = 0$, $\mathbf{X}^{(0)}$, $\mathbf{S}^{(0)}$, \mathbf{Y} , 正则参数 λ_1 , λ_2 , 对偶变量 $\mathbf{V}_1^{(0)}$, $\mathbf{V}_2^{(0)}$, $\mathbf{V}_3^{(0)}$ 、拉格朗日乘子 $\mathbf{U}_1^{(0)}$, $\mathbf{U}_2^{(0)}$, $\mathbf{U}_3^{(0)}$, 收敛误差 ε , 惩罚因子 β 。

1) 迭代

$$2) \mathbf{V}_1^{(k+1)} \leftarrow [\beta(\mathbf{D}\mathbf{X}^{(k)} - \mathbf{U}_1^{(k)}) + \mathbf{Y} - \mathbf{S}^{(k)}] / (1 + \beta)$$

$$3) \mathbf{V}_2^{(k+1)} \leftarrow \text{shrink}(\mathbf{X}^{(k)} - \mathbf{U}_2^{(k)}, \lambda_1 / \beta)$$

$$4) \mathbf{V}_3^{(k+1)} \leftarrow \max(0, \mathbf{X}^{(k)} - \mathbf{U}_3^{(k)})$$

$$5) \mathbf{S}^{(k+1)} \leftarrow \text{shrink}(\mathbf{Y} - \mathbf{V}_1^{(k)}, \lambda_2)$$

$$6) \mathbf{U}_1^{(k+1)} \leftarrow \mathbf{U}_1^{(k)} - (\mathbf{D}\mathbf{X}^{(k+1)} - \mathbf{V}_1^{(k+1)})$$

$$7) \mathbf{U}_2^{(k+1)} \leftarrow \mathbf{U}_2^{(k)} - (\mathbf{X}^{(k+1)} - \mathbf{V}_2^{(k+1)})$$

$$8) \mathbf{U}_3^{(k+1)} \leftarrow \mathbf{U}_3^{(k)} - (\mathbf{X}^{(k+1)} - \mathbf{V}_3^{(k+1)})$$

$$9) \mathbf{X}^{(k+1)} \leftarrow (\mathbf{D}^{\text{T}}\mathbf{D} + 2\mathbf{I})^{-1} [\mathbf{D}^{\text{T}}(\mathbf{V}_1^{(k+1)} + \mathbf{U}_1^{(k+1)} + \mathbf{V}_2^{(k+1)} + \mathbf{U}_2^{(k+1)} + \mathbf{V}_3^{(k+1)} + \mathbf{U}_3^{(k+1)})]$$

10) 直到 $\|\mathbf{G}\mathbf{X}^{(k+1)} + \mathbf{B}\mathbf{V}^{(k+1)}\| \leq \varepsilon$

输出: 最优解 $\mathbf{X}^{(k+1)}$, $\mathbf{S}^{(k+1)}$

1.2 模板更新

模板更新的好坏直接影响追踪的性能。在稀疏追踪模型中提出了用目标模板和琐碎模板相结合的模板更新方法, 对于每一个追踪的目标 \mathbf{y} , 如果 \mathbf{y} 与模板的相似性很高, 则不更新, 否则将 \mathbf{y} 引入模板中并剔除模板中权重较小的向量。引入琐碎模板是为了更好地解决目标遮挡问题, 这种模板更新方法虽然在一定程度上可以减弱跟踪漂移现象, 但是模

板维数较高,提高了模型的计算代价,从而降低了追踪系统的实时性。下面用子空间学习和无监督学习相结合的模板更新方法来降低模板的维数,并且能避免模板更新过快而引入较大的误差。

若给定当前模板 \mathbf{T} , 已追踪到的目标 \mathbf{y} 以及噪声能量相似度 τ , 则拉普拉斯噪声 $\mathbf{S} = \mathbf{Y} - \mathbf{T}\mathbf{Z} - \mathbf{E}$ 。如果 $\|\mathbf{s}\|_2 \leq \tau$, 则对模板 \mathbf{T} 进行更新: 首先分别对 \mathbf{T} 和 \mathbf{y} 进行奇异值分解:

$$\mathbf{T} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \mathbf{y} = \mathbf{u}\mathbf{v}^T \quad (12)$$

进一步给出 τ 的等价定义,令 $\mathbf{c} = \text{mean}(\mathbf{T})$, 则:

$$\|\tau\|_2 \propto \arccos \left\langle \frac{(\mathbf{c}, \mathbf{y})}{\sqrt{(\mathbf{c}, \mathbf{c})}\sqrt{(\mathbf{y}, \mathbf{y})}} \right\rangle \quad (13)$$

式中, τ 等价为目标与模板均值的反余弦,即余弦夹角。文献[4]提出增量子空间学习追踪算法,同理可使用奇异向量 $\mathbf{u}, \mathbf{s}, \mathbf{v}$ 去增量更新 $\mathbf{U}, \mathbf{S}, \mathbf{V}$, 从而得到新的奇异向量 $\mathbf{U}_*, \mathbf{S}_*, \mathbf{V}_*$ 。新的模板可表示为:

$$\mathbf{T}_* = \mathbf{U}_*\mathbf{S}_*\mathbf{V}_*^T \quad (14)$$

考虑模板的维数较大,给定初始类个数为 k , 使用无监督学习K-means方法训练模板:

$$\mathbf{J} = \sum_{i=1}^{n+1} \sum_{k=1}^K r_{ik} \|\mathbf{T}_{i*} - \mathbf{u}_k\|^2, \mathbf{u}_k = \frac{\sum_{n+1} r_{ik} \mathbf{T}_{i*}}{\sum_{n+1} r_{ik}} \quad (15)$$

式中, i 表示第 i 个样本; 当 \mathbf{T}_{i*} 属于类 k 时, $r_{ik} = 1$, 否则 $r_{ik} = 0$; \mathbf{u}_k 为所有属于类 k 的样本的平均值。因此新的模板变为 $\mathbf{T}_{\text{new}} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k]$ 。

本文提出的模板更新不同于传统的模板更新,它强调选择对目标追踪具有重要贡献的模板,而避免使用琐碎模板,并通过K-means算法对模板进行无监督训练,大大剔除了模板的冗余信息,从而提高了追踪的实时性。

1.3 追踪算法

在目标追踪过程中,假设第 $(t-1)$ 帧的目标状态为 q_{t-1} , 第 t 帧的所有观测为 \mathbf{Y}_t 。目标状态更新和观测更新可以表示为:

$$p(q_t | q_{t-1}) \propto w_t G(q_{t-1}, \delta^2) \quad (16)$$

$$p(\mathbf{Y}_t | q_t) \propto \exp(-\|\mathbf{Y}_t - \mathbf{D}\mathbf{X} - \gamma(\mathbf{S})\|) \quad (17)$$

式中, 状态 q_t 由6个仿射参数决定且彼此之间相互独立, 即 $q_t = [t_x, t_y, \theta, \varepsilon, \mu, \rho]$, θ 、 ε 、 μ 、 ρ 分别表示旋转角度、规模大小、长宽比和倾斜度; $[t_x, t_y]$ 表示二维旋转参数; w_t 表示第 t 帧的粒子权重; $G(q_{t-1}, \delta^2)$ 表示第 t 帧的目标状态, 可以由服从均值为 q_{t-1} 、方差为 δ^2 的高斯分布函数决定; 而 $p(\mathbf{Y}_t | q_t)$ 可由同时稀疏 $L_{(p,q)}$ 的最小重构误差得到:

$$\gamma(\mathbf{S}) = \begin{cases} \mathbf{S} & \|\mathbf{S}_i\|_2 \leq \tau \\ 0 & \|\mathbf{S}_i\|_2 > \tau \end{cases} \quad (18)$$

在实际中 w_t 通过粒子滤波算法自适应更新得到, 细节见文献[15]; 方差 $\delta = 0.05$, 仿射参数由仿射矩阵逆变换得到。对应的自适应同时追踪算法, 见算法3。

算法3: 自适应同时稀疏追踪算法

输入: 第 t 帧所有候选粒子 $\mathbf{Y}_t = [\mathbf{y}_t^1, \mathbf{y}_t^2, \dots, \mathbf{y}_t^m]$ 、初始追踪目标 $\mathbf{y}_t^{(0)}$ 、模板 \mathbf{D}_t 、最大迭代次数 Loop、收敛误差 tol、余弦夹角阈值 α 、正则参数 λ_1, λ_2 。

1) 根据式(13)计算追踪目标 \mathbf{y}_t 与模板均值的相似性, 记为 sim

2) 判断 sim 与 α 的大小, 并自适应地选择模型进行追踪

3) IF sim < α

4) 选择算法1进行求解并获得稀疏系数矩阵 \mathbf{X}

5) ELSE sim $\geq \alpha$

6) 选择算法2求解得到稀疏系数 \mathbf{X} 和噪声 \mathbf{S}

7) 根据式(12), 式(14)~式(15)自适应更新模板

8) 追踪目标 $\mathbf{y}_t^j = \arg \max_{1 \leq j \leq m} p(\mathbf{y}_t^j | q_t)$

输出: 追踪目标 \mathbf{y}_t^j 和新的模板 \mathbf{D}_{new} 。

2 实验结果

2.1 数据和实验说明

下面通过数值试验把本文提出的算法与其他5种具有很好追踪性能的算法进行比较, 这5种追踪算法分别为核技巧的循环矩阵追踪(circulant structure of kernels, CSK)^[8]、加速梯度追踪(accelerated proximal gradient, L1APG)^[14]、多任务追踪(multi-task tracking, MTT)^[17]、稀疏原型追踪(sparse prototype tracking, SPT)^[23]以及稀疏联合追踪(sparse collaborative method, SCM)^[21]。所有的实验均基于 Matlab 2012a, 计算机内存为2 GB, CPU为Intel(R) Core(TM)i3。实验数据来源于文献[20]。

本次实验选择了14种不同的具有追踪挑战性的视频, 其中包括遮挡、光照变化、背景混杂、姿势改变、低分辨率和快速运动等影响追踪结果的因素, 如表1所示。其中OV表示目标丢失, BC表示背景混杂, OCC表示完全遮挡, OCP表示部分遮挡, OPR表示旋转出平面, LR表示低分辨率, FM表示快速运动, SV表示大小变化。

在实验中参数设置如下: 正则参数 $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, 惩罚因子 $\beta = 0.1$, 余弦角度阈值 $\alpha_{\min} = 20$, $\alpha_{\max} = 35$, 模板最大基向量个数为15, 粒子采样数

为600, 图像块的大小为 25×25 , 实验最大迭代次数 $Loop=20$ 。参数 λ_1, λ_2 均通过交叉验证方法得到, 且 λ_2 参数的调节满足如下规则, 若噪声 S 的能量较大(即目标遭受较大的遮挡、外形变化或光照变化), 此时 λ_2 的值应该较小, 反之则较大。

表1 各种不同具有挑战性的追踪视频

视频序列	帧数	噪声/s	视频序列	帧数	噪声/s
Walking2	495	SV, OCP, LR	Suv	945	OCC, OV, BC
Car4	659	IV, SV	CarDark	393	IV, BC, LR
Car2	913	IV, SV, BC	Deer	71	FM, LR, BC
Girl	500	OPR, OCC, LR	Singer2	366	IV, OPR, BC
FaceOcc2	812	OCC, OPR, IV	Skater2	435	SV, OPR
Football	362	OCC, OPR, BC	Dudek	1145	OCC, BC, OV
FaceOcc1	892	OCC	Subway	175	OCC, BC

2.2 评价准则

本文实验采用的评价方法有3种, 且每种评价方法都能在一定程度上解释追踪性能的好坏, 分别为局部中心误差(center local error)、重叠率(overlap ratio)、曲线下的面积(area under curve, AUC)。

给定帧的真实目标框 R_g (ground truth) 和追踪目标框 R_t (tracked target bounding), 设它们的中心位置分别为: $p_g = (x_g, y_g)$ 和 $p_t = (x_t, y_t)$, 则局部中心误差为 $CLE = \|p_g - p_t\|_2$, 重叠率为:

$$OR = \frac{\text{area}(R_t \cap R_g)}{\text{area}(R_t \cup R_g)}$$

式中, $\text{area}(\cdot)$ 表示在该区域的所有像素; AUC每一点的值表示重叠率大于给定阈值 η 时该视频追踪的成功率。特别地设定 $\eta = 0.5$, 当重叠率 $OR > 0.5$ 时则认为该帧追踪成功。相关追踪结果如表2~4所示, 表2中AOR表示总的平均重叠率, 表3中ACLE表示总的平均中心误差, 表4中ASR表示总的平均成功率, 表中最好的两个结果分别用红色和蓝色表示。各算法追踪性能如图1~3所示, 图1中平均重叠率越大表示追踪性能越好, 图2中平均中心误差越小表示追踪性能越好, 图3中曲线与x轴围成的面积越大表示追踪性能越好。

表2 基于平均重叠率的各种不同算法性能的对比

Sequences	CSK	L1APG	MTT	SPT	SCM	本文算法
Walking2	0.47	0.71	0.60	0.75	0.80	0.80
Car4	0.48	0.22	0.17	0.88	0.87	0.90
Car2	0.69	0.88	0.89	0.87	0.87	0.90
Girl	0.38	0.70	0.71	0.27	0.25	0.67
Football	0.56	0.41	0.65	0.60	0.57	0.65
FaceOcc1	0.80	0.78	0.66	0.73	0.80	0.75
FaceOcc2	0.78	0.66	0.70	0.80	0.75	0.75
Suv	0.52	0.49	0.03	0.51	0.62	0.79
CarDark	0.75	0.55	0.83	0.86	0.81	0.86

(续表)

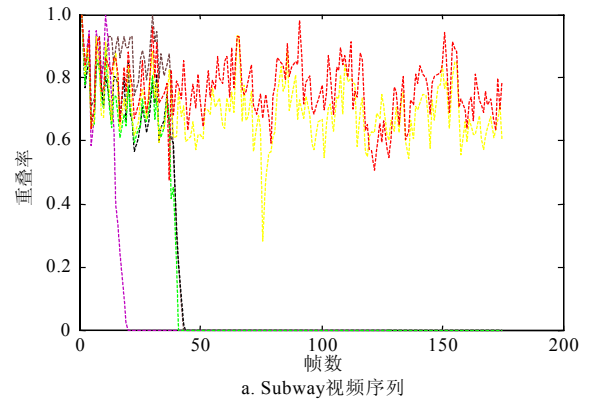
Sequences	CSK	L1APG	MTT	SPT	SCM	本文算法
Deer	0.76	0.68	0.74	0.72	0.66	0.78
Singer2	0.04	0.03	0.04	0.04	0.27	0.58
Skater2	0.59	0.28	0.18	0.36	0.25	0.58
Dudek	0.72	0.78	0.49	0.77	0.78	0.81
Subway	0.20	0.16	0.07	0.16	0.70	0.77
AOR	0.55	0.52	0.48	0.59	0.64	0.76

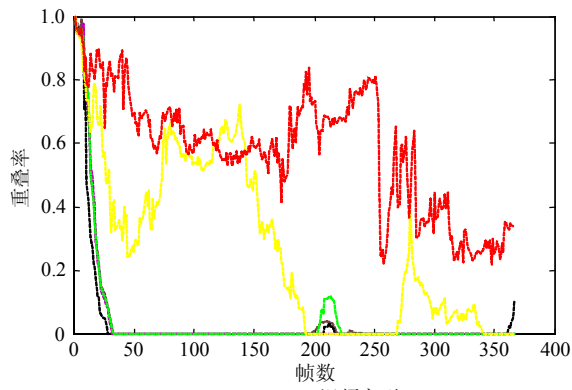
表3 基于平均局部中心误差的各种不同算法性能的对比

Sequences	CSK	L1APG	MTT	SPT	SCM	本文算法
Walking2	17.78	7.12	11.31	2.42	1.94	2.29
Car4	19.13	102.26	425.56	3.02	2.92	1.57
Car2	2.53	1.40	1.50	2.17	1.61	1.79
Girl	19.34	3.27	3.08	11.38	83.44	4.97
Football	16.19	27.40	9.41	13.41	14.03	7.48
FaceOcc1	11.93	13.85	27.74	20.21	12.45	18.16
FaceOcc2	5.92	12.82	10.40	6.05	9.18	8.53
Suv	573.23	75.35	529.75	53.41	32.85	6.50
CarDark	3.23	18.32	1.01	1.21	1.30	0.95
Deer	4.96	17.50	7.12	7.74	13.16	5.68
Singer2	185.47	180.07	208.49	170.48	67.15	13.55
Skater2	16.78	68.32	250.50	40.57	105.90	20.71
Dudek	13.39	8.63	336.24	10.09	11.44	8.71
Subway	164.37	147.15	202.94	140.15	3.91	2.70
ACLE	75.30	48.82	144.65	34.45	25.81	6.40

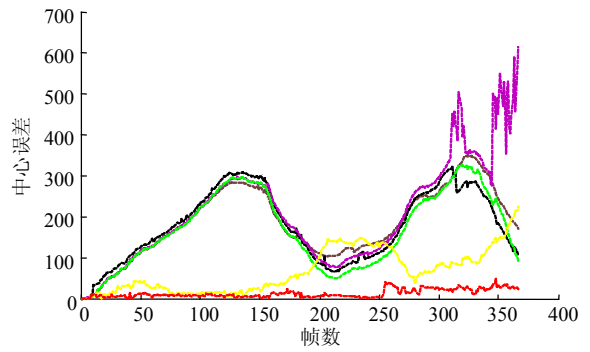
表4 基于平均成功率的各种不同算法性能的对比

Sequences	CSK	L1APG	MTT	SPT	SCM	本文算法
Walking2	40.81	97.78	72.32	100.00	100.00	100.00
Car4	28.22	27.47	19.73	100.00	100.00	100.00
Car2	100.00	100.00	100.00	99.23	100.00	100.00
Girl	41.60	95.20	99.40	27.00	33.40	91.60
Football	67.13	41.16	80.39	81.22	75.14	73.76
FaceOcc1	100.00	100.00	96.41	94.84	100.00	100.00
FaceOcc2	100.00	76.35	80.91	100.00	89.41	98.77
Suv	57.46	53.33	3.39	53.76	70.69	95.56
CarDark	99.24	67.43	100.00	100.00	98.73	97.96
Deer	100.00	88.73	94.37	94.37	88.73	100.00
Singer2	3.83	2.73	3.83	3.83	26.50	71.58
Skater2	77.01	23.22	4.83	28.28	19.77	59.54
Dudek	94.67	93.80	58.25	99.56	98.25	98.95
Subway	22.29	22.29	8.00	21.14	98.86	99.43
ASR	66.59	63.54	58.70	71.66	78.53	91.94

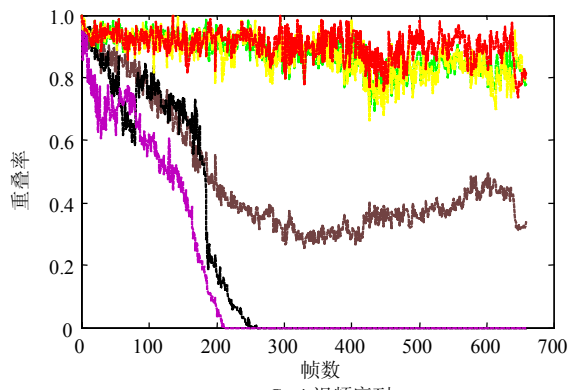




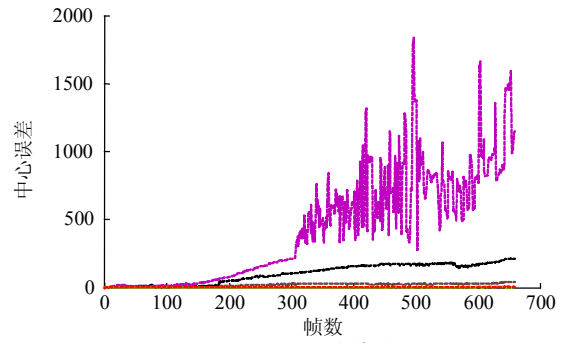
b. Singer2 视频序列



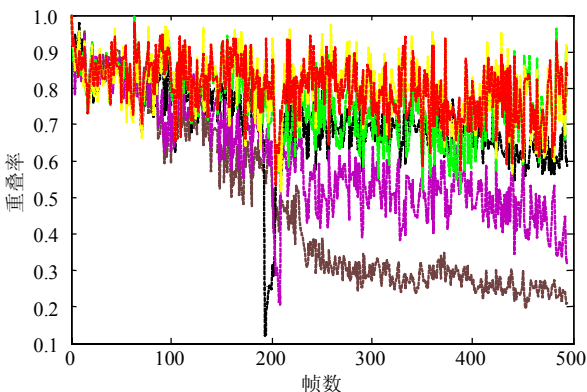
b. Singer2 视频序列



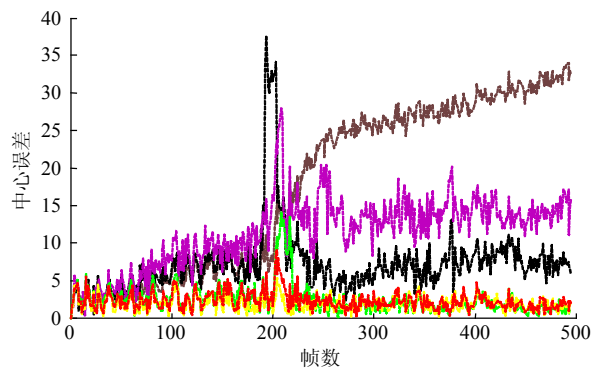
c. Car4 视频序列



c. Car4 视频序列



d. Walking2 视频序列



d. Walking2 视频序列

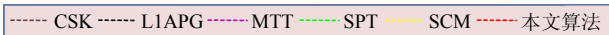
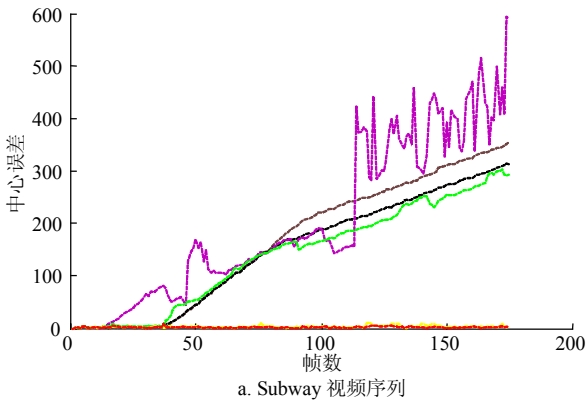


图1 各种不同追踪算法的平均重叠率



a. Subway 视频序列

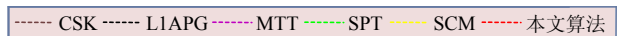
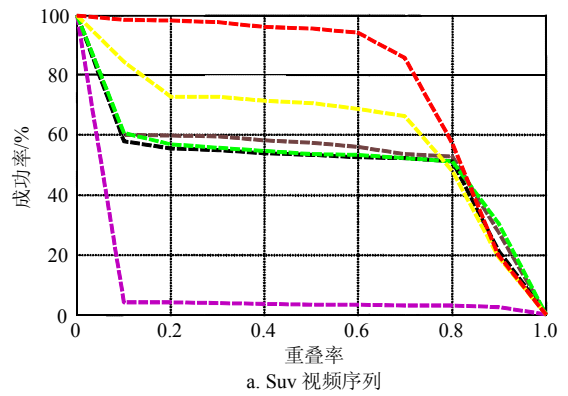
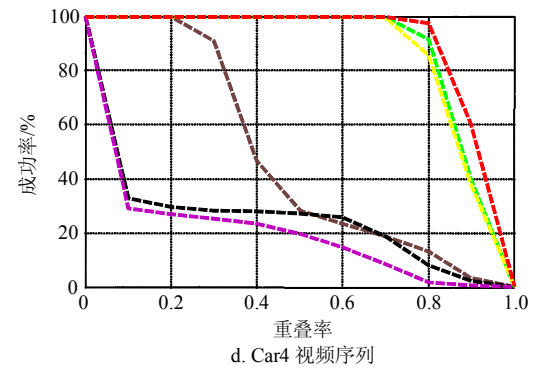
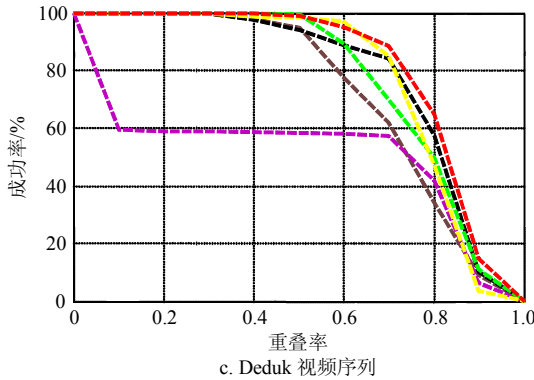
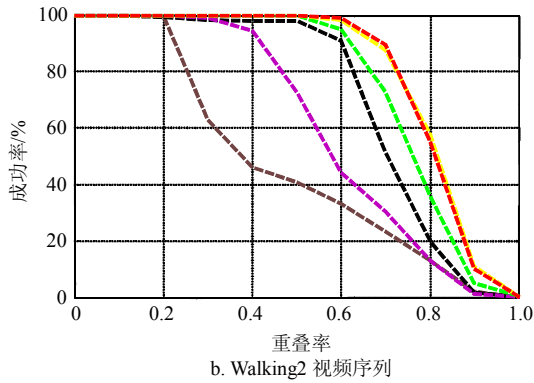


图2 各种不同追踪算法的平均中心误差



a. Suv 视频序列



CSK LIAPG MTT SPT SCM 本文算法

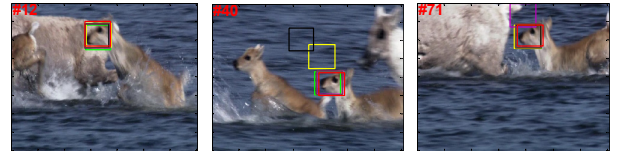
图3 各种不同追踪算法的AUC曲线

2.3 追踪结果分析

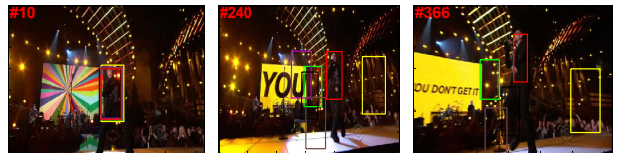
为了方便描述, 将本文提出的追踪算法简记为ASSAT 算法 (adaptive simultaneous sparse representation appearance tracking algorithm)。

1) 背景混杂和快速运动情形: 图4分别给出了快速运动和背景混杂的追踪效果, 视频Deer包含快速运动因素。从Singer2可看到大部分算法都无法有效追踪到结果, 在帧240时, 有很多算法的追踪框偏离目标, 漂移现象很严重, 如SCM, CSK, MTT, LIAPG, SPT等, 只有ASSAT算法可以有效追踪到目标。在帧366时(视频结束)本文提出的依然可以有效追踪到目标, 大大减弱了追踪漂移现象。在视频SUV中, 开始时所有追踪算法均能有效追踪到目标,

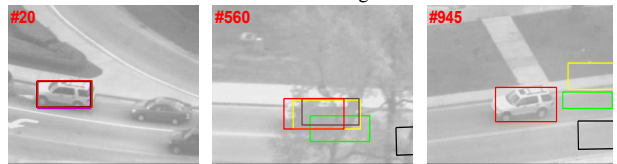
在帧560时, 只有SCM, CSK和ASSAT算法可以追踪到目标, 在第945帧时只有ASSAT可以有效追踪到目标。对于视频Deer而言, 可以看到除了MTT算法大部分算法均能有效追踪到目标, 另外可以看到SCM算法不太稳定, 因在帧40时SCM算法偏离了追踪目标, 出现跟踪漂移现象。ASSAT算法可以有效追踪背景混杂的视频序列是因为其采用了同时稀疏表示方法, 该方法通过模板基的选择来刻画粒子之间的关系, 对噪声污染较大的粒子给与了较低的稀疏权重, 因此具有很强的噪声抗干扰性。



a. 追踪结果Deer



b. 追踪结果Singer2

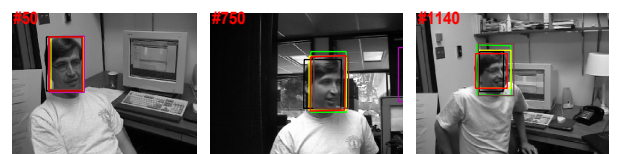


c. 追踪结果SUV

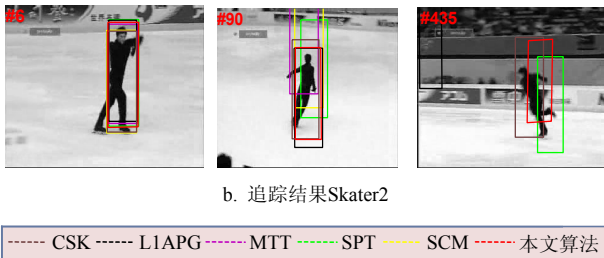
CSK LIAPG MTT SPT SCM 本文算法

图4 基于快速运动和背景混杂视频下的追踪效果

2) 姿势改变和光照变化情形: 图5和图6分别给出了姿势改变和光照变化的追踪效果。从图5中的Deduk视频可以看出所有的追踪算法都能有效追踪到目标(人脸), 但是ASSAT可以更加准确地追踪到人脸目标, 而对于Skater2视频, 大部分算法都无法准确跟踪到目标, 因为目标姿势改变的太过频繁且改变的幅度较大, 在第90帧时只有LIAPG, CSK和ASSAT算法可较为准确的追踪的目标, 但在帧435时只有ASSAT和CSK能够追踪到目标, 尽管追踪的不太精确。图6可以看到几乎所有算法都可以对光照变化产生一定的抗干扰性, 除了LIAPG和CSK算法有一些不太稳定。因此对于姿势改变不大的情况ASSAT算法还是可以有效的追踪到目标。

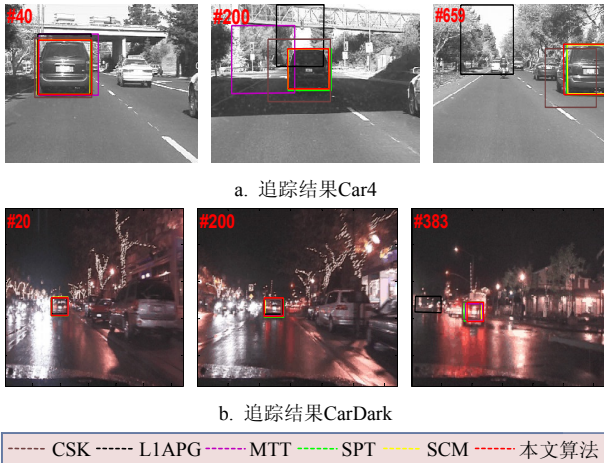


a. 追踪结果Dudek



b. 追踪结果Skater2

图5 基于姿势改变和旋转视频下的追踪效果

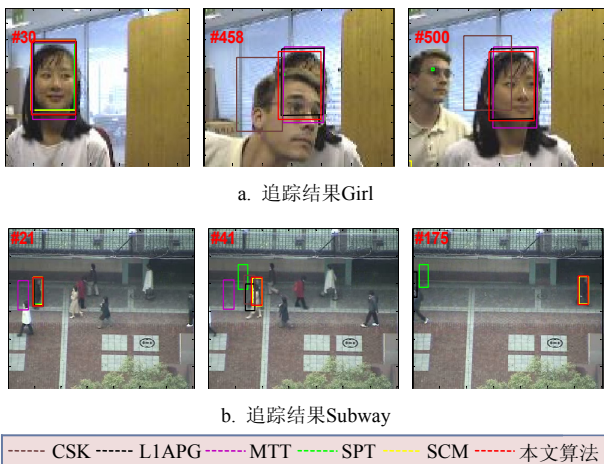


a. 追踪结果Car4

b. 追踪结果CarDark

图6 基于光照变化视频下的追踪效果

3) 目标遮挡情形: 图7展示了目标遮挡对跟踪性能的影响, 在Girl视频序列中可看到在帧458和帧500时, 大部分算法都出现了跟踪漂移现象, 不能有效追踪到目标, 除了ASSAT, LIAPG和MTT算法, 从Subway也可看到在帧41时, 只有SCM, LIAPG和ASSAT算法可有效追踪到目标, 而在帧175时, 除了SCM和ASSAT算法其他算法都出现了很大的跟踪漂移。通过比较可看出ASSAT算法能更好的解决目标遮挡问题, 这是因为该算法考虑了遮挡对跟踪的影响, 它就像一个滤波器在追踪之前对噪声进行了有效剔除, 去除了可能对追踪效果产生影响的不稳定的因素(遮挡、光照变化、背景混杂和姿势改变等)。



a. 追踪结果Girl

b. 追踪结果Subway

图7 基于目标遮挡视频下的追踪效果

3 模型分析

下面详细介绍模型中提到的拉普拉斯噪声和模板更新准则对追踪效果的具体影响。

传统的模板更新方法是直接通过追踪目标与模板的相似度进行更新, 若相似度大于给定的阈值, 则认为目标遭遇了较大的噪声污染, 因此需要将追踪目标替代原始权值较小的模板向量, 这样替换其实是比较粗糙的, 因为引入了较大的噪声误差, 这样就给下一帧目标的追踪造成了很多不确定性, 而本文提出的新的模板更新方法则削弱了噪声影响。具体表现如下:

1) 新的模板更新方法有效权衡原始模板向量和新的追踪目标之间的权重, 通过遗忘因子实现模板更新(具体见文献[4]);

2) 新的模板更新方法引入了K-means方法, 可以有效地降低冗余模板向量, 提高追踪的实时性, 通过式(18)可看到类中心的计算是通过加权平均得到的, 因此也可有效地减弱噪声。

下面给出具体实验分别比较模板更新和拉普拉斯对实验效果的影响。实验数据选择序列Skater2, Dudek, SUV, Walking2, Subway, Deer等。

表5 比较拉普拉斯对实验结果的影响

Sequences	MTT算法	ASSAT (仅拉普拉斯)	ASSAT(拉普拉斯 +模板更新)
Skater2	0.18	0.32	0.58
Dudek	0.49	0.79	0.81
SUV	0.03	0.81	0.79
Walking2	0.60	0.35	0.80
Subway	0.07	0.71	0.77

从表5可以看出除了Walking2序列, 加入拉普拉斯噪声后其追踪效果要优于MTT算法。但是原始模板更新的方法限制了它的追踪性能, 而提出的新模板更新方法促进了ASSAT算法的追踪性能。

表6 比较模板更新对实验结果的影响

Sequences	文献[6](IVT)	ASSAT (仅模板更新)	ASSAT(拉普拉斯 +模板更新)
Skater2	0.24	0.11	0.58
Dudek	0.80	0.80	0.81
SUV	0.14	0.32	0.79
Deer	0.26	0.75	0.75
Subway	0.16	0.16	0.77

从表6中可以看出仅使用模板更新的ASSAT方法和IVT方法的追踪效果差不多, 对于Skater2, Subway序列两种方法效果都不好, 原因是这两种序

列含有较大的遮挡, 对于仅考虑模板更新而没有考虑拉普拉斯噪声的ASSAT算法是无法有效追踪到目标的, IVT也是一样。但两者结合会取得更好的追踪

效果。事实上, 对于这种含有较大遮挡的情况, 若不考虑拉普拉斯噪声, 可归结到噪声因素影响了式(5)中解 x 的稀疏结构, 如图8所示。

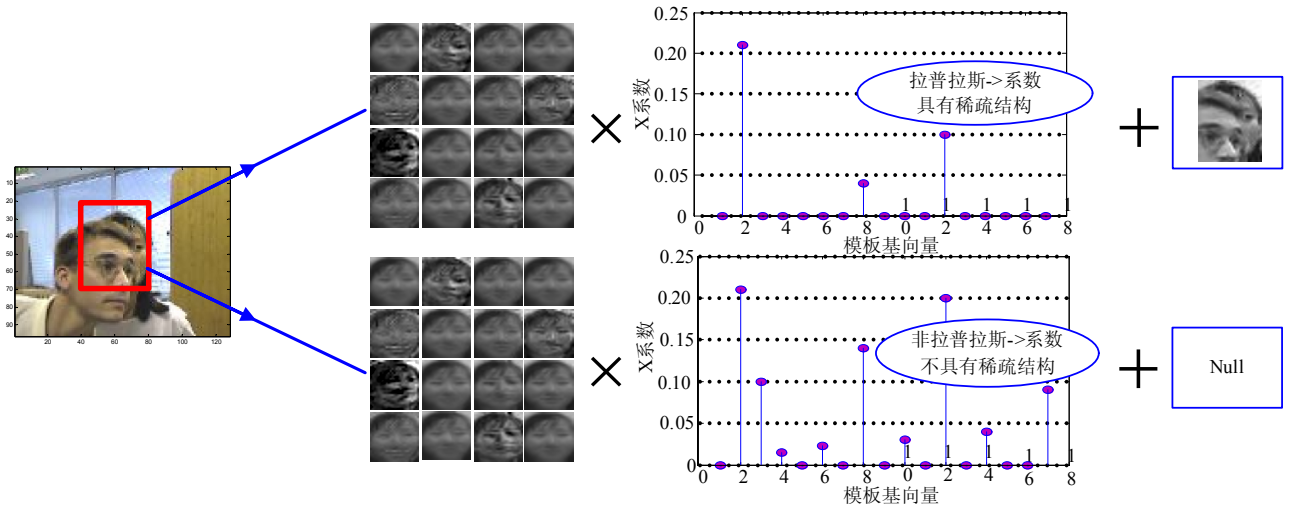


图8 拉普拉斯噪声对解 x 的影响

图8表示目标在遮挡情况不同的情况下噪声选择对解的影响。图8的第一行表示考虑拉普拉斯噪声时所得到的解, 可以看到解是稀疏的, 此时解是最优的; 图8的第二行表示未考虑拉普拉斯噪声时所得到的解, 此时解是稠密的, 非最优解, 因此保持解的稀疏结构直接影响算法的追踪性能。

4 模型的进一步改进

为了提高追踪系统的实时性, 下面考虑自适应 L_p 稀疏表示目标追踪算法, 该算法与前面提出的算法一样, 也考虑了噪声分布的影响, 且模板更新采用在线增量学习与K-means相结合的模板更新方法, 唯一不同的是, 模型中认为粒子之间的影响是相互独立的, 可以分别通过稀疏模型进行求解, 且使用了LASSO求解算法, 根据粒子滤波框架理论和最大后验概率求出最优追踪结果。数值试验结果表明该算法相对于目前存在的较好的追踪算法取得了更好的追踪效果, 表现为精确性更高、实时性更好。

4.1 自适应 L_p 模型的建立和求解

假设噪声分布服从高斯拉普拉斯分布: 噪声 n 服从高斯分布($n \sim N(0, \sigma_n^2)$), 噪声 s 服从拉普拉斯分布($s \sim N(0, \sigma_s)$), 候选目标 y 可以写成:

$$y = Tx + n + s \quad (19)$$

通常在噪声污染较小的情况下, 认为候选目标 y 可由模板 T 的少量基的线性组合来表示, 这就意味着求解系数 x 是稀疏的, 因此稀疏表示模型变为:

$$x_{\text{opt}} = \arg \min_x \|y - Tx - s\|_2^2 + \lambda_1 \|x\|_1 + \lambda_2 \|s\|_1 \quad (20)$$

同前面分析一样, 令:

$$\chi(s) = \begin{cases} 0 & \|s\|_2 \leq \tau \\ s & \|s\|_2 > \tau \end{cases}$$

式中, τ 是一个阈值, 当噪声能量大于 τ 时, 此时需要考虑拉普拉斯噪声的影响, 反之则忽略拉普拉斯噪声的影响。因此最终模型如下:

$$\begin{aligned} \min_{x,s} & \|y - Tx - \chi(s)\|_2^2 + \lambda_1 \|x\|_1 + \lambda_2 \|\chi(s)\|_1 \\ \text{s.t.} & x \geq 0 \end{aligned} \quad (21)$$

式(21)的求解可以采用交替迭代的方法, 每一次迭代都使用一次LASSO算法, 令:

$$L(x, s) = \|y - Tx - \chi(s)\|_2^2 + \lambda_1 \|x\|_1 + \lambda_2 \|\chi(s)\|_1$$

即:

$$x^{k+1} = \arg \min_x L(x, s^k)$$

$$s^{k+1} = \arg \min_{s^k} L(x^{k+1}, s^k) = \text{shrink}(Tx^{k+1} - y, 1/\lambda_2)$$

式中, shrink是一个紧缩算子。

4.2 自适应 L_p 模型的追踪算法

目标状态的更新和预测是目标追踪算法的核心问题。前面介绍了增量学习与K-means相结合的模板更新方法, 本节提出的算法依然采用这种模板更新机制。类似的给出了自适应 L_p 模型的追踪算法4。

算法4: 自适应 L_p 模型追踪算法

输入: 第 t 帧候选粒子 $Y_t = [y_t^1, y_t^2, \dots, y_t^m]$ 、最大迭代次数Loop、收敛误差tol、阈值区间 $[\alpha, \beta]$ 、正

则参数 λ_1 、 λ_2 。

1) 初始化 $i = 0$, $j \in [1, m]$, 拉普拉斯噪声 $s_i^j = 0$

2) 根据式(13)计算追踪目标 y_i 与模板均值的相似性, 记为 sim

3) 迭代

$$\begin{aligned} x_i^{i+1} &= \arg \min \|y_i^j - Tx_i^i - \chi(s_i^j)\|_2^2 + \\ &\quad \lambda_1 \|x_i^i\|_1 + \lambda_2 \|\chi(s_i^j)\|_1, x_i^i \geq 0 \\ \chi(s_i^{i+1}) &= \text{shrink}(Tx_i^{i+1} - y_i^j, 1/\lambda_2) \\ i &\leftarrow i + 1 \end{aligned}$$

4) 迭代直到 $i > \text{Loop}$ 或者

$$|L(x_i^{i+1}, \chi(s_i^{i+1})) - L(x_i^i, \chi(s_i^i))| < \text{tol}$$

5) 获得最好追踪目标 $y_{\text{opt}} = \arg \max_{1 \leq j \leq m} p(y_i^j | q_i)$

6) 通过式(12)~式(15)更新目标模板 T , 并得到新模板 T^*

输出: 追踪目标 y_{opt} 和新的模板 T^*

4.3 实验结果与分析

选取20种不同的视频序列(<http://cvlab.hanyang.ac.kr/trackerbenchmark/datasets.html>), 实验中模板基的个数设定为15, 最大迭代次数和收敛误差分别为 $\text{Loop} = 5$, $\text{tol} = 0.001$, 阈值区间 $\alpha = 20$, $\beta = 28$ 。使用LASSO算法^[24]求解 L_1 模型, 在默认的情况下正则参数 $\lambda_1 = 0.2$, $\lambda_2 = 0.1$, 实际中通过交叉验证的方法可以设置参数 $\lambda_2 \in (0.0005, 0.5)$ 。

将算法4和目前9种比较好的算法进行比较, 除了前面的CSK算法^[8]、LIAPG算法^[14]、MTT算法^[17]、SPT算法^[23]和SCM算法^[21]、再增加DFT算法^[25]、ORIA算法^[26]、IVT算法^[4]、ASLSA算法^[7]。

表7和表8给出了在20种不同情境下的定量数据展示。表7中AOR表示平均重叠得分, Fps表示每秒追踪的帧数, Fps越大表示追踪的越快即说明该算法的实时性越好。本文提出的算法AOR=0.74和ACLE=6.5是最好的, 而Fps=9.53排名第三, 这表明算法4在保证追踪精度的同时实时性也较好。

表7 10种追踪算法的追踪性能的定量分析表

视 频	DFT	CSK	LIAPG	ORIA	MTT	IVT	ASLSA	SPT	SCM	本文算法
Walking2	0.41	0.47	0.71	0.49	0.60	0.63	0.82	0.75	0.80	0.82
Walking	0.57	0.55	0.71	0.14	0.58	0.72	0.76	0.74	0.70	0.76
Car4	0.24	0.48	0.22	0.21	0.17	0.91	0.89	0.88	0.87	0.89
Car2	0.16	0.69	0.88	0.74	0.89	0.91	0.88	0.87	0.87	0.92
Girl	0.29	0.38	0.70	0.48	0.71	0.40	0.67	0.27	0.25	0.68
Gil2	0.37	0.39	0.38	0.36	0.34	0.37	0.42	0.38	0.42	0.73
Football	0.62	0.56	0.41	0.70	0.65	0.52	0.53	0.60	0.57	0.63
Football1	0.90	0.58	0.16	0.22	0.68	0.67	0.61	0.68	0.47	0.70
FaceOcc1	0.69	0.80	0.78	0.62	0.66	0.74	0.31	0.73	0.80	0.75
FaceOcc2	0.77	0.78	0.66	0.69	0.70	0.75	0.81	0.80	0.75	0.77
Suv	0.08	0.52	0.49	0.65	0.03	0.14	0.83	0.51	0.62	0.85
Surfer	0.05	0.11	0.14	0.18	0.45	0.64	0.73	0.65	0.75	0.68
CarDark	0.38	0.75	0.55	0.42	0.83	0.83	0.81	0.86	0.81	0.80
Deer	0.26	0.76	0.68	0.12	0.74	0.26	0.68	0.72	0.66	0.75
Singr2	0.63	0.04	0.03	0.04	0.04	0.04	0.65	0.04	0.27	0.58
Skater2	0.07	0.59	0.28	0.13	0.18	0.24	0.49	0.36	0.25	0.50
Dudek	0.69	0.72	0.78	0.70	0.49	0.80	0.79	0.77	0.78	0.79
RedTeam	0.62	0.62	0.71	0.40	0.62	0.74	0.77	0.76	0.74	0.79
Subway	0.74	0.20	0.16	0.07	0.07	0.16	0.73	0.16	0.70	0.75
Car24	0.10	0.41	0.79	0.42	0.79	0.78	0.75	0.50	0.87	0.73
AOR	0.43	0.52	0.51	0.39	0.51	0.56	0.70	0.60	0.65	0.74
Fps	5.01	150.94	0.73	3.30	1.45	18.14	3.53	2.58	0.67	9.53

表8 10种追踪算法的ACLE比较

视 频	DFT	CSK	LIAPG	ORIA	MTT	IVT	ASLSA	SPT	SCM	本文算法
Walking2	28.8	17.8	7.1	9.57	11.3	2.9	2.1	2.4	1.9	2.4
Walking	5.9	7.2	3.5	212.4	9.5	2.2	2.1	1.7	2.2	2.9
Car4	61.9	19.1	102.3	123.6	425.6	1.6	2.0	3.0	2.9	2.2
Car2	87.7	2.5	1.4	3.2	1.5	1.4	1.2	2.2	1.6	1.5
Girl	24.0	19.3	3.3	12.7	3.1	17.2	4.6	11.4	83.4	4.6
Gil2	129.9	157.2	115.6	115.4	204.1	128.1	122.0	119.2	114.4	8.1
Football	9.29	16.2	27.4	6.4	9.4	15.2	15.8	13.4	14.0	7.4
Football1	1.52	8.1	48.5	24.1	5.6	5.7	6.9	5.5	14.0	4.7
FaceOcc1	23.6	11.9	13.9	25.5	27.7	18.0	81.3	20.2	12.5	18.4
FaceOcc2	7.9	5.9	12.8	12.0	10.4	8.7	5.6	6.1	9.2	7.9
Suv	111.4	573.2	75.4	38.0	529.8	92.3	4.3	53.4	32.9	2.9

(续表)

视 频	DFT	CSK	L1APG	ORIA	MTT	IVT	ASLSA	SPT	SCM	本文算法
Surfer	115.7	37.5	29.7	74.4	10.4	3.5	3.2	4.9	3.2	4.6
CarDark	58.9	3.2	18.3	26.2	1.0	1.9	1.5	1.2	1.3	1.2
Deer	98.8	5.0	17.5	87.3	7.1	121.0	20.4	7.7	13.2	6.5
Singer2	21.9	185.5	180.1	185.8	208.5	173.1	10.9	170.5	67.2	17.1
Skater2	125.0	16.8	68.3	80.1	250.5	77.4	25.0	40.6	105.9	18.9
Dudek	18.7	13.4	8.6	15.2	336.2	9.4	11.6	10.1	11.4	10.3
RedTeam	2.9	2.6	3.2	7.8	2.9	2.6	2.6	2.4	2.5	2.3
Subway	3.3	164.4	147.2	162.8	202.9	147.3	3.7	140.2	3.9	2.8
Car24	160.5	8.1	2.3	1.5	1.6	1.2	2.4	59.4	1.8	2.9
ACLE	54.9	63.7	44.3	61.2	113.0	41.5	16.4	33.8	25.0	6.5

图9展示了目标遭受快速运动、尺度变化和姿势旋转等噪声影响的追踪实验结果。对于视频Football, 很明显本文算法可成功追踪到目标, 其他算法都丢失了目标(见帧318和帧362); 对于视频Skater2, 大部分算法都不能准确追踪到目标, 在帧10时, 每一种追踪算法都可以捕捉到目标, 但在帧200时, 因为目标形态改变太多, 此时已经没有算法可以准确捕捉到目标了, 在帧435时, 只有本文的算法、CSK和ASLSA算法可以捕捉到目标的一部分; 对于视频Deer和Surfer, 可很清楚地看到本文的算法, SCM, SPT, ASLSA和L1APG算法可有效捕捉到目标。其原因归结于本文的算法综合考虑了噪声的

影响, 且将噪声部分进行很好地剔除, 使得目标可以被模板稀疏表示, 因而追踪成功。

在实验中所有的参数都通过交叉验证的方法得到, 在追踪过程中参数 λ_1 不灵敏, 一般可根据经验设置在区间0.05~0.5, 本文设置为0.1。难点是参数 λ_2 的调整, 因参数 λ_2 直接影响拉普拉斯噪声能量的大小, 一般相对来说, λ_2 越大拉普拉斯噪声能量越小。本文只给出了参数 λ_2 的经验调整范围, 当目标遭受较大的噪声污染时, 如完全遮挡, 此时 λ_2 通常取较小的值一般在0.000 5~0.01之间, 反之参数 λ_2 取值在0.01~0.5之间。

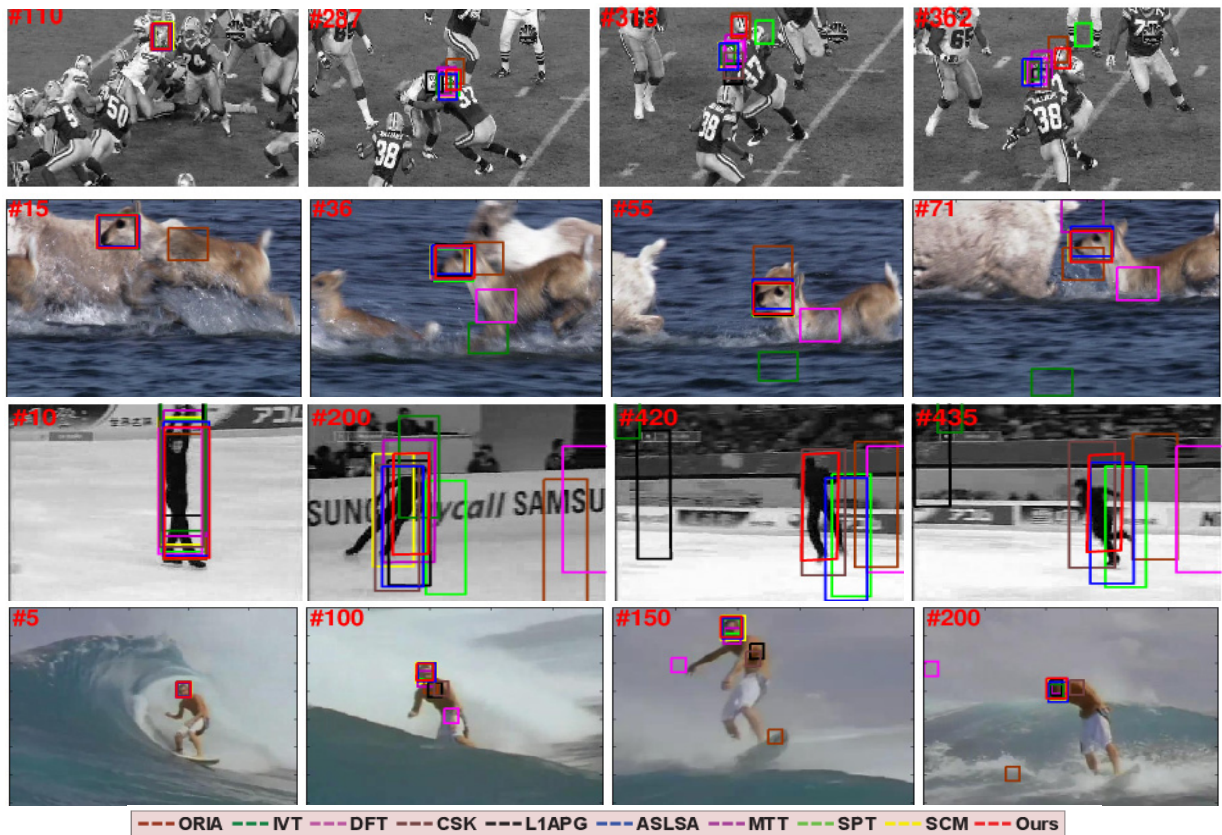


图9 几种追踪算法在快速运动和姿势旋转视频中的追踪效果

5 结束语

本文提出了同时稀疏表示的自适应追踪算法,该方法考虑了拉普拉斯噪声的影响,并根据噪声的能量大小自适应的选择稀疏模型。另外,模型使用了2种稀疏表示:同时稀疏($L_{p,q}$)法,综合考虑了粒子之间的关系,使用ADMM方法对模型进行求解;对 L_p 模型,认为粒子之间的影响是相互独立的,可以分别通过LASSO算法分别求解;不同情境下的大量实验表明,这两种方法均得到了稳定的结果。

参 考 文 献

- [1] LI A, LIN M, WU Y, et al. NUS-PRO: a new visual tracking challenge[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 38(2): 335-349.
- [2] YANG H, SHAO L, ZHENG F, et al. Recent advances and trends in visual tracking: a review[J]. Neurocomputing, 2011, 74(18): 3823-3831.
- [3] ADAM A, RIVLIN E, SHIMSHONI I. Robust fragments-based tracking using the integral histogram[J]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, 1: 798-805.
- [4] ROSS D A, LIM J, LIN R S, et al. Incremental learning for robust visual tracking[J]. International Journal of Computer Vision, 2008, 77(1-3): 125-141.
- [5] LIWICKI S, ZAFEIRIOU S, TZIMIROPOULOS G, et al. Efficient online subspace learning with an indefinite kernel for visual tracking and recognition[J]. IEEE Transactions on Neural Networks and Learning Systems, 2012, 23(10): 1624-1636.
- [6] BABENKO B, YANG M H, BELONGIE S. Robust object tracking with online multiple instance learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1619-1632.
- [7] YANG F, LU H, YANG M H. Robust superpixel tracking[J]. IEEE Transactions on Image Processing, 2014, 23(4): 1639-1651.
- [8] HENRIQUES J F, CASEIRO R, MARTINS P, et al. Exploiting the circulant structure of tracking by detection with kernels[C]//Computer Vision-ECCV. [S.l.]: Springer, 2012, 7575: 702-715.
- [9] WRIGHT J, YANG A Y, GANESH A, et al. Robust face recognition via sparse representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(2): 210-227.
- [10] 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.
- [11] SUN B, LIU Z, SUN Y, et al. Multiple objects tracking and identification based on sparse representation in surveillance video[C]//IEEE International Conference on Multimedia Big Data. Beijing: IEEE Computer Society, 2015.
- [12] WANG D, LU H, BO C. Online visual tracking via two view sparse representation[J]. IEEE Signal Processing Letters, 2014, 21(9): 1031-1034.
- [13] LI Y, HE Z, YI S, et al. The robust patches-based tracking method via sparse representation[C]//International Conference on Security, Pattern Analysis, and Cybernetics. Wuhan: IEEE, 2014.
- [14] DUAN X, LIU J, TANG X. Visual tracking via weighted sparse representation[C]//International Conference on Intelligent Computing and Internet of Things. Harbin: IEEE, 2015.
- [15] BAO C, WU Y, LING H, et al. Real time robust L1 tracker using accelerated proximal gradient approach[C]//IEEE Conference on Computer Vision and Pattern Recognition. Providence: IEEE, 2012.
- [16] LU X, YAO H, SUN X, et al. Real-time visual tracking using L2 norm regularization based collaborative representation[C]//IEEE International Conference on Image Processing. Melbourne: IEEE, 2013.
- [17] AHUJA N. Robust visual tracking via multi-task sparse learning[C]//Computer Vision and Pattern Recognition. Providence: IEEE, 2012.
- [18] YAN Q, LI L, WANG C, et al. Kernel sparse representation for object tracking[C]//International Conference on Multimedia Information Networking and Security. Beijing: IET, 2013.
- [19] WANG L, YAN H, LV K, et al. Visual tracking via kernel sparse representation with multikernel fusion[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2014, 24(7): 1132-1141.
- [20] WU Y, LIM J, YANG M H. Online object tracking: a benchmark[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Portland: IEEE, 2013.
- [21] ZHONG W, LU H, YANG M H. Robust object tracking via sparse collaborative appearance model[J]. IEEE Transactions on Image Processing, 2014, 23(5): 2356-2368.
- [22] MEI X, LING H. Robust visual tracking using L_1 minimization[C]//IEEE 12th International Conference on Computer Vision. Kyoto: DBLP, 2009.
- [23] WANG D, LU H, YANG M H. Online object tracking with sparse prototypes[J]. IEEE Transactions on Image Processing, 2013, 22(1): 314-325.
- [24] MAIRAL J, BACH F, PONCE J, et al. Online dictionary learning for sparse coding[C]//International Conference on Machine Learning. Quebec: DBLP, 2009.
- [25] LEARNEDMILLER E, SEVILLALARA L. Distribution fields for tracking[C]//Computer Vision and Pattern Recognition. Providence: IEEE, 2012.
- [26] HE J, ZHANG D, BALZANO L, et al. Iterative Grassmannian optimization for robust image alignment[J]. Image and Vision Computing, 2014, 32(10): 800-813.