计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(12):143−149 [doi: 10.15888/j.cnki.csa.006666]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

面向小运动目标的压缩域跟踪方法①

张鑫生, 刘　浩, 孙晓帆, 况奇刚, 吴乐明

(东华大学 信息科学与技术学院, 上海 201620)
通讯作者: 刘　浩, E-mail: liuhao@dhu.edu.cn

摘　要: 压缩域跟踪是直接从压缩码流中提取运动矢量和块编码模式来实现目标对象的跟踪. 针对现有压缩域跟踪方法对小运动目标跟踪性能较差的问题, 本文提出了一种面向小运动目标的压缩域跟踪算法. 在分析现有算法不足原因的基础上, 本文从起始帧掩模的获取、离群值边界的设置和预测跟踪小目标的边缘控制三个方面提升小目标跟踪的性能, 并通过数据驱动的方法寻找到块编码感知的系统参数优化. 所提算法在三个小目标视频序列上进行了测试, 实验结果表明, 与其它压缩域跟踪算法相比, 本文算法可以有效地提高小运动目标跟踪的准确率和 F 度量.

关键词: 压缩域; 目标跟踪; 块编码; 运动矢量; 小目标

引用格式: 张鑫生,刘浩,孙晓帆,况奇刚,吴乐明.面向小运动目标的压缩域跟踪方法.计算机系统应用,2018,27(12):143–149. http://www.c-s-a.org.cn/1003-3254/6666.html

Compressed-Domain Object Tracking for Small Moving Targets

ZHANG Xin-Sheng, LIU Hao, SUN Xiao-Fan, KUANG Qi-Gang, WU Le-Ming

(School of Information Science and Technology, Donghua University, Shanghai 201620, China)

Abstract: The compressed-domain object tracking approaches utilize the information that is directly extracted from the compressed bitstream, such as motion vector and block coding modes. Because the existing compressed-domain tracking methods have poor tracking performance for small moving targets, this study proposes a compressed-domain tracking algorithm for small moving targets. By analyzing the shortages of the existing algorithms, the performance of small target tracking is improved from the acquisition of initial frame mask, the setting of outlier boundary and the edge control of the predicating small target, and some system parameters of the block-coding system are optimized through data-driven methodology. Experiential results on three small-target video sequences show that compared with other object tracking methods, the proposed method can effectively improve the tracking performance for small moving targets in terms of accuracy and F-measure.

Key words: compressed domain; object tracking; block coding; motion vector; small target

1　序言

视频目标跟踪是许多机器视觉应用的重要组成部分, 例如运动识别、人机交互、自动监视和交通监控[1]. 目标跟踪问题可以定义为当图像视频中的物体在场景周围移动时, 如何估计物体轨迹; 相应的跟踪算法就是在某一视频的后续帧中预先为被跟踪对象分配一致的标签[2]. 在目标跟踪领域中, 小运动目标的跟踪是一个难点和重点, 尤其现在的视频分辨率越来越高, 跟踪目标在一帧中的占比越来越小, 多样化的背景和复杂的摄像机运动都增加了小目标跟踪的难度[3].

编码视频的目标跟踪算法根据其依赖信息可分为两大类: 像素域跟踪算法和压缩域跟踪算法[4]. 像素域

① 基金项目: 上海市自然科学基金 (18ZR1400300)
Foundation item: Natural Science Foundation of Shanghai (18ZR1400300)
收稿时间: 2018-04-26; 修改时间: 2018-05-17; 采用时间: 2018-06-05; csa 在线出版时间: 2018-12-03

跟踪算法的特点是跟踪的准确率高, 但是计算复杂度也高. 由于相当高的计算复杂度, 像素域跟踪算法在需要并行处理多个视频流的场景中并不适用. 另一方面, 压缩域算法处理的数据来自于压缩码流中已经编码的信息, 例如运动矢量、块编码模式或运动补偿的预测残差变换系数等. 压缩域跟踪算法相比于像素域跟踪算法, 由于避免了对视频的全部解码, 计算成本通常较低, 显著减少了数据处理量和存储需求. 然而, 由于压缩域跟踪算法没有利用全部的像素信息, 目标跟踪的准确率往往不及像素域跟踪算法.

基于马尔科夫模型来实现压缩域目标跟踪的算法近年来得到了广泛的研究[5]. Zeng 等人[6]提出的算法是最早使用马尔科夫随机场 (MRF) 进行目标跟踪的算法之一, 该方法通过最小化 MRF 能量, 将相似的运动矢量 (MV) 合并到运动对象中, 并定义了不同的 MV 类型. 方法[6]将跟踪问题转化为在已经分类的 MV 邻域上进行马尔可夫标记的过程, 在这个过程中该方法利用运动矢量的空间连续性和时间一致性来追踪移动目标. 但是, 文献[6]没有考虑潜在的摄相机运动, 因此, 它只适用于跟踪摄像机固定拍摄的物体运动[7]. Khatoonabadi 等人[8]提出了另外一种使用 MRF 模型进行目标跟踪的算法, 该方法不是先将运动矢量分成多个类型, 而是直接使用运动矢量的观测值来计算运动相关性. 此外, 文献[8]采用全局运动补偿来处理摄像机运动造成的影响, 并提出一种基于相邻块的运动矢量来为帧内编码块分配运动矢量的方法. Xu 等人[9]提出基于全局运动估计的视频显著图检测方法来对目标进行跟踪. 尽管文献[8]和[9]在一些传统视频序列下的跟踪性能良好, 但是针对小目标的跟踪, 它们的表现却不理想.

2 现有方法分析

基于 MRF 模型的压缩域目标跟踪算法对当前帧的预测会参考前一帧的运动目标预测结果, 导致前一帧的预测错误会影响到当前帧的预测. 由于小目标本身在一帧中所占的比例很小, 故而对预测错误的容忍度更小. 以 H.264 视频编码标准为例, 由于 H.264 采用的是块划分[10], 所以在得到起始帧的掩模时要进行 4×4 的块划分, 需要将起始帧的标准参考图像的高、宽分别缩小 4 倍, 然后用得到的起始帧掩模来计算下一帧的预测结果, 这个预测结果需要再将高、宽放大 4 倍来得到下一帧预测的像素域表示, 整个过程如图 1 所示, 图中括号的内容是不同阶段的图像尺寸.

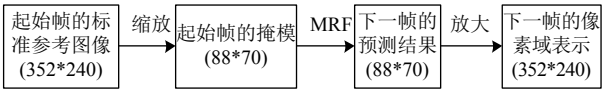


图 1 基于 MRF 的目标跟踪算法流程图

为了对目标跟踪算法进行比较, 通常使用准确率 (*Precision*)、召回率 (*Recall*) 和 F 度量 (*F-Measure*) 来衡量算法的客观性能[11]. 接下来分析起始帧的掩模特性. 假设将掩模放大得到的起始帧的像素域表示没有 *FN*, 则 $FP=\lambda\cdot TP$, 可得:

$$Precision=\frac{TP}{TP+FP}=\frac{TP}{TP+\lambda\cdot TP}=\frac{1}{1+\lambda} \quad (1)$$

图 2 给出了 *Precision* 和倍数λ的关系图, 从图中可以看出, 若 $Precision=0.8$, 则$\lambda=2$, $FP=0.25TP$; 若 $Precision=0.33$, 则$\lambda=2$, $FP=2TP$, 即λ 越大, *Precision* 就越小. 从大量实验中发现, 占比越大的目标, 相对应的λ越小; 占比越小的目标, 相对应的λ越大.

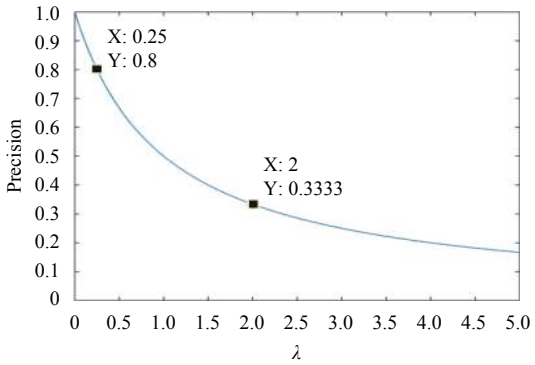
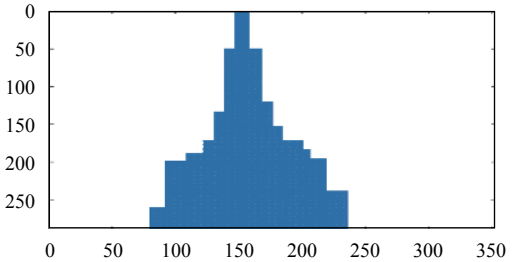


图 2 *Precision* 和倍数λ的关系图 (*Precision* 为小数形式)

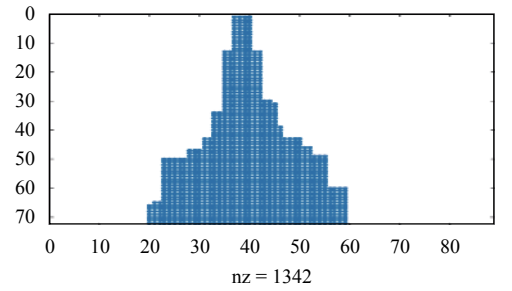
下面以 City 和 Ground 这两个视频序列为例进行压缩域性能分析. 在视频序列 City 的起始帧中, 如图 3(a) 所示, 标记为 1 的像素点有 20 299 个 (即 *TP* 值); 对此帧 4×4 块划分得到的掩模中, 如图 3(b) 所示, 标记为 1 的像素子块有 1342 个; 如果对此掩模按照 4×4 比例放大, 则标记为 1 的像素点有 21 472 个 (即 *TP* 值和 *FP* 值的和). 经计算, 此掩模的 *Precision* 为 94.5%, 利用这个掩模进行目标物体的跟踪, 即使像素点有几千个, *Precision* 至少可保持 80% 左右.

在视频序列 Ground 的起始帧中, 如图 4(a) 所示, 标记为 1 的像素点有 64 个 (即 *TP* 值), 对此帧进行 4×4 块划分得到的掩模如图 4(b) 所示, 因为图像编码

最小块是 4×4, 在 MRF 算法中标记为 1 的像素子块有 11 个. 如果对此掩模按照 4×4 比例放大, 则标记为 1 的像素点有 176 个 (即 *TP* 值和 *FP* 值的和). 此掩模的 *Precision* 为 36.3%, 用这个掩模进行目标物体的跟踪, 即使预测错误的像素点有几十个, *Precision* 也仅有 30% 左右. 图 4(c) 展示了 Ground 起始帧的标准参考图像 (黑色) 和由此帧的掩模放大后得到的图像 (灰色), 从中可以看出掩模放大后得到的图像相比标准参考图像而言, 边缘部分过多.



(a) 起始帧的标准参考图像



(b) 起始帧掩模放大后的像素域表示

图 3 City 序列起始帧变化图

3 所提算法

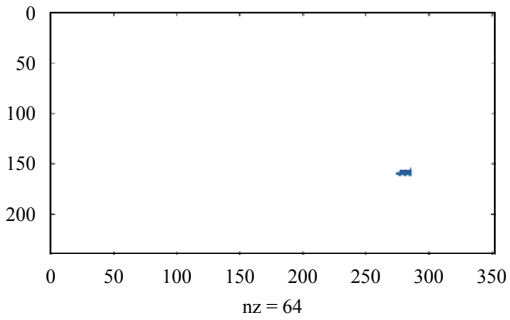
3.1 算法框架

基于以上的分析, 本文提出了一种面向小运动目标的压缩域跟踪算法, 算法流程图如图 5 所示.

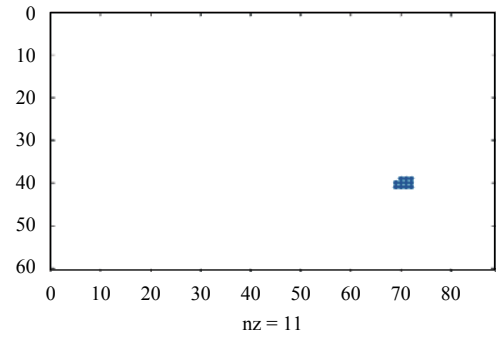
3.2 算法数学模型

本文所提小目标跟踪算法的原理是: 将视频的每一帧划分为多个子块, 每个子块标记为“0”或“1”, 其中“0”表示这个子块是跟踪目标的一部分, “1”则反之, 这样目标跟踪问题可转换为已知某一帧 $t-1$ 的标记 w^{t-1}和帧 t 的运动信息$K^t=(v^t,o^t)$的条件下, 预测出帧 t 的标记w^t, 其中运动信息K^t从压缩码流提取得来, v^t为子块的 MV, o^t为子块的编码模式和大小, $n=(x,y)$表示子块在这一帧的位置. 使后验概率$P=(w^t|w^{t-1},K^t)$最大化的w^t是最佳的标记, 根据贝叶斯理论, 运动目标的帧间似然性为$P(w^{t-1}|w^t,K^t)$, 运动目标的帧内似然性为$P(K^t|w^t)$, 运动目标的先验概率为$P(w^t)$, 那么后验概率$P=(w^t|w^{t-1},K^t)$可以表示为:

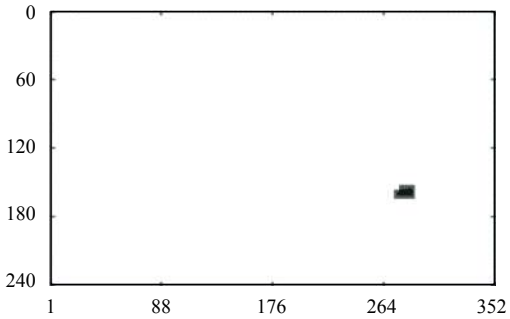
$$P=(w^t|w^{t-1},K^t)=\frac{P(w^t|w^t,K^t)\cdot P(K^t|w^t)\cdot P(w^t)}{P(w^{t-1},K^t)} \quad (2)$$



(a) 起始帧的标准参考图像



(b) 起始帧掩模放大后的像素域表示



(c) 起始帧开始和结束时的像素域表示的对比效果

图 4 Ground 序列起始帧变化图

从式 (2) 中可以看出, 分母的计算不需要w^t, 因此后验概率$P=(w^t|w^{t-1},K^t)$最大只需某个w^t使分子最大, 即:

$$w^t=\arg\max_{\psi\in\Omega}\left\{P(w^{t-1}|\psi,K^t)\cdot P(K^t|\psi)\cdot P(\psi)\right\} \quad (3)$$

其中, Ω 表示全部可能的标记w^t的集合. 式 (3) 等同于

$$w^t = \arg\min_{\psi\in\Omega}\{-\log P(w^{t-1}|\psi,K^t)-\log P(K^t|\psi)-\log P(\psi)\} \tag{4}$$

基于第 2 节块编码对精度性能影响的分析, 为了改善对小运动目标的跟踪效果, 本文引入了门限 α 和 β 以及系数 γ 将式 (4) 改写为:

$$w^t = \arg\min_{\psi\in\Omega}\{-\gamma\cdot\log P(\alpha\cdot w^{t-1}|\psi,K^t)- \log P(K^t|\psi)-\log P(\psi)\} \tag{5}$$

式 (5) 中, $P(K^t|\psi)$经计算可写为:

$$P(K^t|\psi)=\min\left\{\frac{d'(n)/\sigma_{d'}-2}{2},1\right\}/\rho \tag{6}$$

$$d'(\mathbf{n})=\begin{cases} d(\mathbf{n})d(\boldsymbol{n})\leqslant\beta\cdot\sigma_d \\ 0d(\boldsymbol{n})>\beta\cdot\sigma_d \end{cases} \tag{7}$$

$$d(\mathbf{n})=\left\|v'(\mathbf{n})-\hat{v}\right\|_2 \tag{8}$$

式 (6) 中, ρ 为全局运动补偿系数, $\sigma_{d'}$是$d'(\boldsymbol{n})$的标准差; 式 (7) 中σ_d是$d(\boldsymbol{n})$的标准差; 式 (8) 中$v'(\boldsymbol{n})$是运动矢量, $\hat{v}$是目标中心运动矢量.

参考图 5, 在空域代价计算模块中, 要利用从压缩码流中提取的 MV 进行参数估计. 但是参数估计方法有一个明显的问题, 由于离群值 (噪声 MV 或不准确的 MV) 的存在, 导致样本方差的估计对离群值非常敏感. 对小目标而言, 这个问题更加突出, 即便是几个离群值也会造成很大的估计误差. 所以针对小目标的情况, 重新计算离群值边界门限 β 是很有必要的.

鉴于前面对起始帧掩模的分析, 可以知道基于 MRF 的算法中, 起始帧的掩模中过多的块被标记为 1. 因此, 本文算法引入了门限 α, 用来控制起始帧的掩模中标记为 1 的块的数量.

在时域代价计算模块中, 由于噪声 MV 或不准确的 MV 的干扰, 时域代价的计算也会受到影响, 特别是在目标的边界上, 而从前面对起始帧掩模的分析中, 可以看出小目标起始帧掩模中边界被错误标记为 1 的块的数量最多, 这个不利因素加剧了小目标边界计算时可能发生的错误. 故本文算法引入了系数 γ, 用以控制跟踪小目标的边缘计算.

3.3 块编码感知的系统参数优化

为了优化上述三个参数的设置, 以便得到更好地小目标跟踪效果, 我们进行了一系列的实验仿真. 通用的实验结果是在量化步长 (QP) 为 28 时针对小目标 Ground 视频序列的 *Precision* 指标获得的. 典型地, α 取值为 1/4、1/2、3/4 和 1; β 取值为 1.8、1.9、2.0 和 2.1; γ 取值为 4、4.4、4.8 和 5.2; 总共进行了 64 组参数测试. 我们将实验数据按照各个参数的不同取值, 以柱状堆叠的形式展现; 每个参数在四个取值的柱状图均由 16 个 *Precision* 值 (对应于其它两个参数的 16 种组合) 相加而得, 分别如图 6(a)、6(b) 和 6(c). 从图中可以看出, 当 α 设置为 1, β 设置为 1.8, γ 设置为 4.8 时, *Precision* 的通用效果最好.

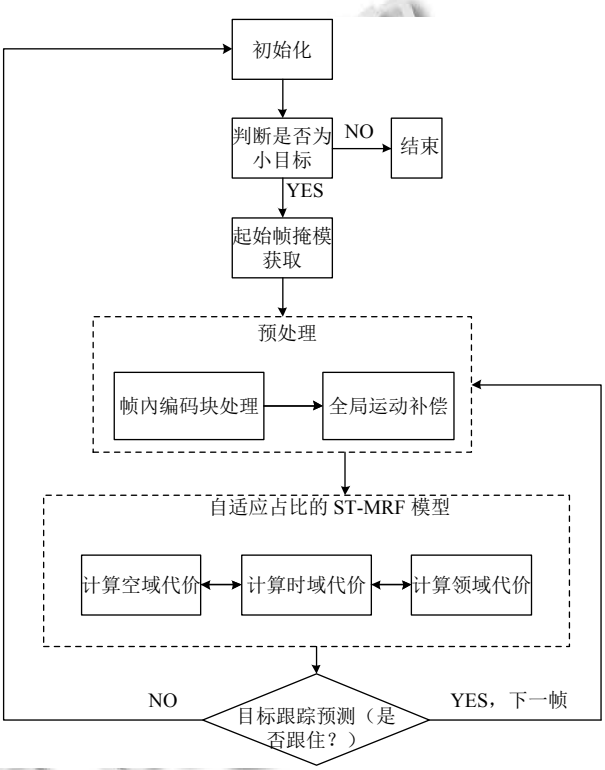


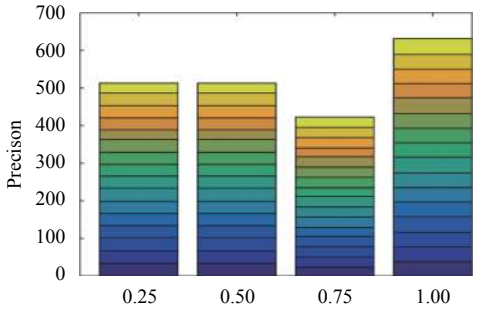
图 5 本文算法流程图

需要指出的是, 本文中此 3 个变量的选取是由 Ground 视频的跟踪结果来决定的, 实际场合中, 此 3 个变量依赖于提供的不同视频而变动. 由于不同视频的场景和特点未必与 Ground 视频相同, 可能导致在跟踪的过程中, 检测到某一帧的 *Precision* 为 0, 若出现这种情况则需要重新选取跟踪的小目标, 按照上述方式重新设置参数.

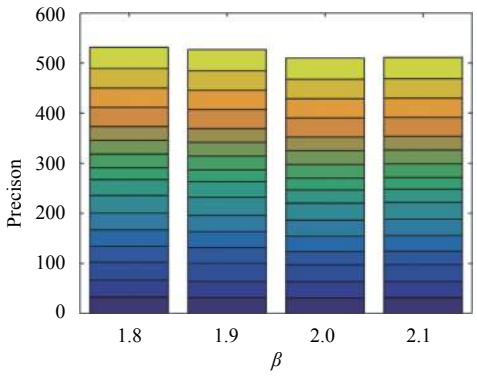
4 实验分析

针对标准视频序列中的小目标跟踪, 本节将所提算法与文献[8]和文献[9]的算法在 *Precision*、*Recall* 和 *F-Measure* 指标下的逐帧性能进行了比较. 从图 7 中可以看出, 就 *Precision* 和 *F-Measure* 而言, 尽管算

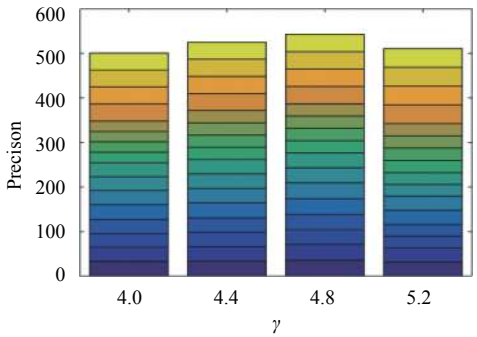
法[8]和[9]在开始的几帧表现较好, 但是在随后的跟踪中, 算法[8]和[9]的跟踪性能越来越差; 而本文算法在整体上的表现优于算法[8]和[9]. 对于 *Recall*, 本文算法在整体上的表现略低于算法[8], 这是因为对于给定的算法而言 *Precision* 和 *Recall* 是矛盾的, 而本文算法以小幅牺牲 *Recall* 来改进其它两大指标性能. 在表 1 中, 给出了本文算法、算法[8]和算法[9]在所有测试视频序列下的平均 *Precision*、*Recall* 和 *F-Measure*. 从表中可以看出, 对于全部的视频序列, 本文算法的 *Precision* 和 *F-Measure* 在三种算法中是最好的.



(a) α 门限

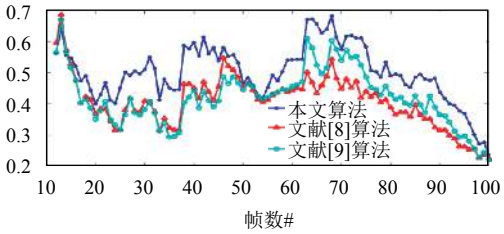


(b) β 门限

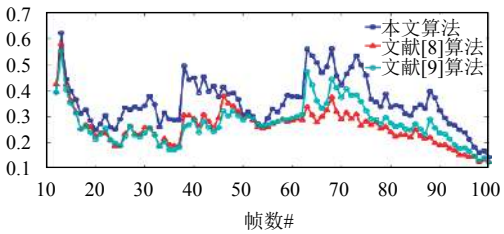


(c) γ 系数

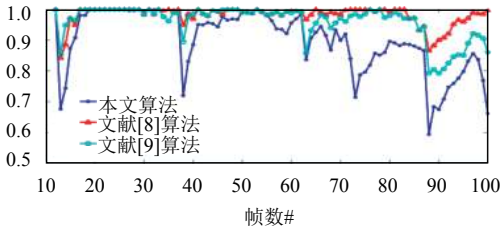
图 6 Ground 序列在 64 组实验参数下的 *Precision* 结果. *Precision* 为百分比形式.



(a) F-Measure



(b) Precision



(c) Recall

图 7 三种算法的指标随 Sky 序列帧号的变化图

表 1 三种算法的平均 *Precision*、*Recall* 和 *F-Measure*

算法	指标	Ground(QP28)	Sky(QP28)	Seasky(QP24)	平均值
本文	*Precision*	39.1	35.9	25.9	33.6
	Recall	89.4	89.9	88.5	89.3
	F-Measure	53.3	50.2	38.6	47.4
文献[9]	*Precision*	30.8	27.5	20.3	26.2
	Recall	90.1	96.0	20.4	68.9
	F-Measure	44.4	42.2	20.0	35.5
文献[8]	*Precision*	29.5	26.0	18.3	24.6
	Recall	89.8	98.1	92.6	93.5
	F-Measure	42.7	40.6	28.8	37.4

表 2 给出了 QP 值从 0 变化到 50, 本文算法在不同视频测试序列下 *Precision*、*Recall* 和 *F-Measure* 的变化. 同时, 图 8 也描述了不同 QP 值下三个视频序列的平均 *Precision*、平均 *Recall* 和平均 *F-Measure*. 从这些结果可以看出, 在 QP 值等于 40 之前本文算法在小目标情况下的跟踪性能比较稳定; 在 QP 值大于 40 后, 各项指标开始下降, 这是由于在高压缩比时, 从压缩码流中提取的运动矢量的精度较低. 图 9(a) 和 9(b) 给出了不同 QP 值下本文算法与算法[8]和算法[9]对三个视频序列的平均 *Precision* 和平均 *F-Measure*, 实验结果

表明, 本文算法在常用 QP 值下的平均 *Precision* 表现最好, 同时由 *F-Measure* 结果可知, 本文算法的综合性能较好.

表 2 本文算法在不同 QP 下的跟踪性能

指标	QP	Ground	Sky	Seasky
Precision	0	71.6	33.0	19.9
	10	61.6	35.0	18.0
	20	40.8	35.9	16.4
	30	49.3	42.6	19.8
	40	30.8	43.5	23.9
	50	0.0	3.6	0.0
Recall	0	28	89.3	93.4
	10	76.5	87.0	85.9
	20	93.4	94.0	96.1
	30	83.0	88.3	63.2
	40	86.7	85.9	95.2
	50	0.0	2.6	0.0
F-Measure	0	34.9	47.2	31.8
	10	68.0	48.4	29.0
	20	55.3	50.4	27.3
	30	61.6	54.9	29.4
	40	43.9	55.3	38.0
	50	0.0	2.8	0.0

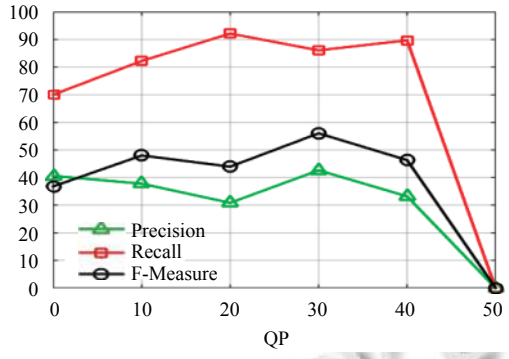
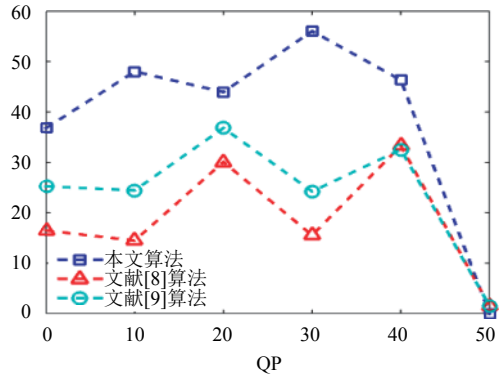


图 8 本文算法在不同 QP 下的平均跟踪结果

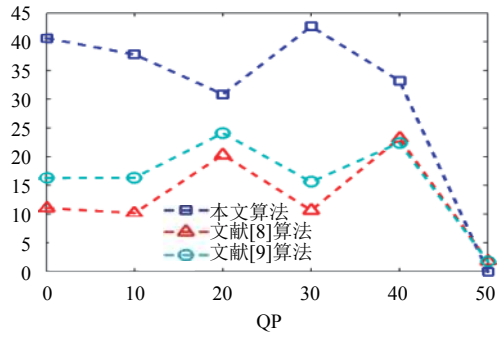
表 3 列出了本文算法对三个视频序列中每一帧的平均处理时间. 实验中使用的计算机的配置为 Intel(R) i5-4570 3.20 GHz CPU, 8 G 内存. 需要指出的是解码 mv 所需的时间没有包含在表 3 中, 这是因为解码 mv 的时间依赖于不同解码器的选择, 通常来说，解码所需的时间很少。从表 3 可以看出, Ground 视频序列每帧的平均处理时间为 82.8 ms; Sky 视频序列为 111.7 ms; Seasky 视频序列为 113 ms. 实验中使用了 MATLAB 实现本文算法, 若使用 C/C++实现，每帧的平均处理时间会大大减少.

表 3 不同视频序列每帧的平均处理时间 (单位: ms)

帧视频	Ground	Sky	Seasky
时间	82.8	111.7	113



(a) 平均 *F-Measure*



(b) 平均 *Precision*

图 9 三种算法在不同 QP 下的指标比较

5 结论

本文根据马尔科夫随机场理论, 提出了一种小运动目标的压缩域跟踪算法. 本文首先分析了在小运动目标的情况下现有压缩域跟踪算法效果较差的原因, 在此基础上寻找出可以优化的算法机制, 并结合数据驱动的方法论给出了块编码感知的参数设置. 实验结果表明, 相较于其它算法, 本文算法的小目标跟踪 *Precision* 和 *F-Measure* 均得到了显著地提高.

参考文献

1 Wu Y, Lim J, Yang MH. Online object tracking: A benchmark. Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR, USA. 2013. 2411–2418.

2 Yilmaz A, Javed O, Shah M. Object tracking: A survey. ACM Computing Surveys, 2006, 38(4): 13. [doi: 10.1145/1177352]

3 张微, 康宝生. 相关滤波目标跟踪进展综述. 中国图象图形学报, 2017, 22(8): 1017–1033.
4 Chen YM, Bajić IV. Compressed-domain moving region segmentation with pixel precision using motion integration. Proceedings of 2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. Victoria, BC, Canada. 2009. 442–447.
5 李旭超, 朱善安. 图像分割中的马尔可夫随机场方法综述. 中国图象图形学报, 2007, 12(5): 789–798. [doi: 10.3969/j.issn.1006-8961.2007.05.004]
6 Zeng W, Du J, Gao W, *et al*. Robust moving object segmentation on H.264/AVC compressed video using the block-based MRF model. Real-Time Imaging, 2005, 11(4): 290–299. [doi: 10.1016/j.rti.2005.04.008]
7 Babu RV, Tom M, Wadekar P. A survey on compressed domain video analysis techniques. Multimedia Tools and Applications, 2016, 75(2): 1043–1078. [doi: 10.1007/s11042-014-2345-z]
8 Khatoonabadi SH, Bajić IV. Video object tracking in the compressed domain using spatio-temporal Markov random fields. IEEE Transactions on Image Processing, 2013, 22(1): 300–313. [doi: 10.1109/TIP.2012.2214049]
9 Xu J, Tu Q, Li CW, *et al*. Video saliency map detection based on global motion estimation. Proceedings of 2015 IEEE International Conference on Multimedia & Expo Workshops. Turin, Italy. 2015. 1–6.
10 Richardson IEG. H.264 and MPEG-4 video compression: Video coding for next-generation multimedia. New York: John Wiley & Sons, 2004.
11 王闪, 吴秦. 基于马尔可夫随机场模型的运动对象分割算法. 传感器与微系统, 2016, 35(7): 113–115, 119.