一种基于加权时空上下文的鲁棒视觉跟踪算法

徐建强 1,2 陆耀 1,2

摘 要 由于光照及外观变化、复杂背景、目标旋转与遮挡等因素的影响,给实现鲁棒的视觉跟踪带来困难.有效利用上下文 (Context) 中包含的有用信息有助于提升上述条件下视觉跟踪的鲁棒性.时空上下文 (Spatio-temporal context, STC) 算法 是新近提出的一种基于时空上下文的目标跟踪算法,它利用目标周围的稠密上下文信息,取得了良好的跟踪效果.STC 的不足 是其同等对待整个上下文区域,没有对上下文做进一步的区分,减弱了上下文的作用.本文采用动态分区处理思想,根据上下 文中不同区域与跟踪目标运动相似度大小,赋予不同权值,提出了基于加权时空上下文 (Weighted spatio-temporal context, WSTC) 的鲁棒视觉跟踪算法.最后在公共数据集上进行的对比实验表明,本文所提出的算法具有更好的跟踪效果和鲁棒性. 关键词 视觉跟踪,上下文,时空上下文,加权时空上下文

引用格式 徐建强, 陆耀. 一种基于加权时空上下文的鲁棒视觉跟踪算法. 自动化学报, 2015, **41**(11): 1901–1912 **DOI** 10.16383/j.aas.2015.c150073

Robust Visual Tracking via Weighted Spatio-temporal Context Learning

XU Jian-Qiang^{1, 2} LU Yao^{1, 2}

Abstract Implementing a robust visual tracker is a challenging task due to many disturbing factors such as illumination changes, appearance changes, rotation, partial or full occlusion, etc. The local context surrounding of the target could provide much effective information in getting a robust tracker. The spatio-temporal context (STC) learning algorithm proposed recently considers the information of the dense context around the target and has achieved a better performance. However, STC treats the whole region of the context equally, which weakens the effectiveness of the context information. In this paper, we propose a novel weighted spatio-temporal context (WSTC) learning algorithm. Our algorithm considers the surrounding context discriminatively and incorporates a weighted matrix by evaluating the motion consistencies of different regions with the tracking target. Extensive experimental results on public benchmark databases show that our algorithm outperforms the original STC algorithm and other state-of-the-art algorithms.

Key words Visual tracking, context, spatio-temporal context (STC), weighted spatio-temporal context (WSTC)

Citation Xu Jian-Qiang, Lu Yao. Robust visual tracking via weighted spatio-temporal context learning. Acta Automatica Sinica, 2015, 41(11): 1901–1912

视觉跟踪是计算机视觉领域的热门研究课题之一.目前,在受控环境下,视觉跟踪能够取得比较理想的跟踪效果,但在非受控环境下,实现鲁棒的视觉跟踪仍然存在着挑战.这主要来自于光照变化、外观变化、复杂背景、目标旋转以及目标被部分甚至完全遮挡等诸多因素.近年来研究者们经过不懈的努力,提出了大量优秀的视觉跟踪算法,如 MIL^[1]、IMIL^[2]、IVT^[3]、FCT^[4]、VTD^[5]、TLD^[6]及 Box-

PF^[7]、IA-MCMC^[8]、ANNF-SSAMC^[9] 算法等.

为了达到鲁棒跟踪的目的,一些研究者利用目标周围的上下文 (Context) 环境来辅助进行跟踪目标的定位,取得了良好的效果.一般来说,上下文环境可粗略分为两类:空间上下文环境及时间上下文环境.空间上下文环境包括目标本身及周围一定范围内的背景,如图1(a)所示,外层方框内的整个区域就是空间上下文环境,其中包括了内层方框内的跟踪目标.而时间上下文指跟踪视频中前后相关的一系列帧的集合及其它们之间的对应关系.

许多文献都探索了上下文环境在跟踪中的应用.例如,Grabner等^[10]在上下文中指定了一些关键点作为支撑点(Supporters),并用这些支撑点进行投票来共同决定跟踪目标的位置,即使目标已经离开可视区域,仍然可以通过这些支撑点定位到跟踪目标.Dinh等^[11]预先在上下文中提取了与目标相似的物体作为干扰目标 (Distracters),并用一系列 SURF (Speeded up robust features) 特征点来

收稿日期 2015-02-04 录用日期 2015-07-11

Manuscript received February 4, 2015; accepted July 11, 2015 国家自然科学基金 (61273273, 61271374), 高等学校博士学科点专项 科研基金 (20121101110034) 资助

Supported by National Natural Science Foundation of China (61273273, 61271374) and Research Fund for the Doctoral Program of Higher Education of China (20121101110034)

本文责任编委 徐昕

Recommended by Associate Editor XU Xin

北京理工大学计算机学院 北京 100081
 智能信息技术北京市 重点实验室 北京 100081

School of Computer Science, Beijing Institute of Technology, Beijing 100081
 Beijing Laboratory of Intelligent Information Technology, Beijing 100081

区别真正目标及干扰目标,实现了有相似目标存在 的情况下的有效跟踪. Wen 等^[12]分别对时间上下 文及空间上下文进行了建模,同时利用 SURF 特征 点信息建立了支撑域 (Supporting field), 从而实现 了复杂环境下的目标跟踪. Yang 等^[13] 采用数据挖 掘的方法,先在目标周围分割出一些区域作为辅助 对象 (Auxiliary objects), 然后对辅助对象及目标 本身进行合作跟踪 (Collaborative tracking), 达到 了较好的效果,上面的几种算法相对比较复杂,运 行效率不高, 达不到实时跟踪的要求, 而且由于提 取的关键点或者辅助对象从本质上来说都是稀疏 的,上下文环境中大量的剩余区域没有得到充分利 用,降低了上下文环境的使用效率.针对这些不足, Zhang 等^[14] 新近提出了一种非常高效的时空上下 文 (Spatio-temporal context, STC) 算法, 该算法 考虑了目标周围的稠密上下文环境(相对于上面稀 疏上下文环境而言),把整个上下文环境融入到跟踪 算法中,使上下文环境中的信息得到了充分利用,取 得了非常好的效果,其性能超过了大部分的当前主 流算法.



(a)

(b)

图 1 上下文环境及相应的权值矩阵 ((a) 内层方框为目标跟 踪结果, 其中心坐标为 x^* , 外层方框为当前上下文环境, 其中 包含目标本身, 用 $\Omega_c(x^*)$ 表示; (b) 本文算法针对当前上下 文生成的权值矩阵, 是算法执行过程中的中间结果) Fig. 1 Context and weighted matrix ((a) The object is inside the rectangle centering at x^* and the context $\Omega_c(x^*)$ is the region inside the rectangle which includes the object region; (b) The weighted matrix created by our WSTC tracker with respect to the context $\Omega_c(x^*)$)

STC 算法存在一个不足就是它同等对待目标周围的整个上下文区域,认为整个上下文区域对跟踪的贡献是一致的.其实在非受控环境下,一个完整的上下文环境可能包含了非常复杂的内容,有针对性地分析上下文中的不同区域,有助于进一步增强跟踪算法的鲁棒性.为了有效利用上下文中的有益信息,同时尽量排除干扰,本文采用了对上下文进行分区处理的思想.按照区域外观及其运动轨迹与目标

中心运动轨迹是否一致,本文把上下文大致分为三 类区域,即辅助区域、无关区域及干扰区域.辅助区 域指那些与跟踪目标运动轨迹一致或基本一致的区 域;无关区域指那些在跟踪过程中保持静止不变的 背景区域;干扰区域指那些与跟踪目标运动方向相 差很大,甚至是完全相反的区域.

图 2 给出了一个区域划分的示意图, 基中左图 展示了一个完整的上下文区域. 其中圆点代表像素, 点的灰度代表图像外观, 箭头表示运动方向, 中间的 虚线框代表跟踪目标. 右图为根据外观及运动划分 得到的结果. 其中 A、E 及 G 区域为辅助区域, B 和 F 区域为无关区域, C、D 及 H 区域为干扰区域. 需 要注意的是, 这三类区域并没有规则的形状, 相互之 间也没有明显的物理界限, 且各区域在跟踪过程中 是随时间动态变化甚至是相互转化的. 同时目标本 身所在区域也可能划分到不同区域中, 例如当发生 遮挡时, 目标所在区域中的被遮挡部分就变成了干 扰区域.





Fig. 2 Context and division result ((a) The dots are pixels with different grayscale and the arrows show the motion directions; (b) The result of division)

本文的基本思想就是针对上下文中不同区域内 的像素与跟踪目标之间运动轨迹的相似性,构建出 一个与上下文区域内像素一一对应的稠密权值矩阵, 与目标运动轨迹越相似的区域,其像素获得权值就 越高,反之就越低,最后再把这个权值矩阵整合进 STC 算法的跟踪框架中,从而达到了更加鲁棒的跟 踪性能.图1(b)展示了本文算法执行过程中实际得 到的一个权值矩阵.

本文剩余部分是按如下方式进行组织的:首先, 回顾了 STC 算法及其实现流程.其次,详细介绍 了本文提出的加权时空上下文 (Weighted spatiotemporal context, WSTC) 算法及实现过程.再次, 讨论了跟踪过程中的错误跟踪点消除问题.最后,本 文展示了大量对比实验结果并进行了分析讨论,并 得出了本文的最终结论.

1 基于 STC 的视觉跟踪算法

STC 是最近由 Zhang 等^[14] 提出的非受控环境 下的视觉跟踪算法, STC 算法把跟踪过程转化为一 系列寻找置信图 (Confidence map) 极值的过程.即 在第一帧, STC 算法先根据相关公式计算出一个置 信图, 然后把置信图极值所在位置当作跟踪目标的 中心.在当前帧中,目标中心位置用 x^* 来表示,则 可以定义出当前帧中上下文特征集合: $X^c = \{v(z) = (I(z), z) | z \in \Omega_c(x^*)\},$ 其中 z 代表位置坐标, I(z)表示图像中 z 处像素的灰度值, $\Omega_c(x^*)$ 是围绕目标 中心 x^* 产生的局部上下文区域 (如图 1 (a) 所示). 通过一些基本的概率公式, 就可以把置信图 c(x) 用 公式表示为

$$c(x) = \sum_{v(z) \in X^{c}} P(x, v(z)|o) = \sum_{v(z) \in X^{c}} P(x|v(z), o) P(v(z)|o) \quad (1)$$

其中, x 和 z 都是二维的位置坐标, o 代表跟踪对 象. P(x|v(z), o) 表示当跟踪对象存在且该位置具有 v(z) 的上下文特征时,该位置存在跟踪目标的概率. 不难看出,通过这个条件概率公式,就建立了跟踪目 标所在位置与其周围上下文环境特征之间的相互关 系. P(v(z)|o) 是上下文环境先验模型,其作用是对 上下文外观特征进行建模.在STC中, c(x) 由一个 已有的函数进行表示,是已知的,因此,该算法的主 要任务就转化为如何学习得到 P(x|v(z), o),即空间 上下文模型.

1.1 上下文先验概率模型

为了得到空间上下文模型 P(x|v(z),o),我们首 先要得到上下文先验概率模型 P(v(z)|o),在 STC 算法中,对 P(v(z)|o)进行了如下的建模:

$$P(v(z)|o) = I(z)\omega_{\sigma}(z - x^*)$$
(2)

式中的 ω_{σ} 是一个加权高斯函数,定义为

$$\omega_{\sigma}(d) = a \times \exp\left(-\frac{d^2}{\sigma^2}\right) \tag{3}$$

其中, I(z) 表示图像中 z 坐标处像素的灰度值, a 是 一个规则化的常数, σ 是一个尺度参数.

1.2 置信图函数

对于式 (1) 中的置信图函数 *c*(*x*), STC 直接用 一个已经存在的函数进行表示, 即

$$c(x) = b \times \exp\left(-\left|\frac{x - x^*}{\alpha}\right|^{\beta}\right) \tag{4}$$

其中, b 是规则化常数, α 是尺度参数, β 是形状参数. STC 算法指出, 当 $\beta > 1$ 时, 其图形比较平滑, 不利于跟踪过程中目标中心点的定位, 而当 $\beta < 1$ 时, 图形的尖锋又过于尖锐, 会造成跟踪器对噪声过于敏感. 因此 STC 算法采用 $\beta = 1$ 作为最终的参数.

1.3 空间上下文模型

在 STC 中, 条件概率函数 *P*(*x*|*v*(*z*), *o*) 就是要 求取的空间上下文模型, 其定义为

$$P(x|v(z), o) = h^{sc}(x - z)$$
(5)

将式 (2)~(5) 代入式 (1) 中, 可以得到

$$c(x) = \sum_{v(z)\in X^c} P(x|v(z), o)P(v(z)|o) =$$
$$\sum_{v(z)\in X^c} h^{sc}(x-z)I(z)\omega_{\sigma}(z-x^*) =$$
$$h^{sc}(x)\otimes (I(x)\omega_{\sigma}(x-x^*))$$
(6)

其中, ⊗ 表示卷积操作. 卷积操作计算量大, 速度较 慢. 但众所周知, 图像卷积之后再进行傅里叶变换 等价于先进行傅里叶变换, 然后再进行对应像素的 乘积. 而快速傅里叶变换 (Fast Fourier transformation, FFT) 的速度非常快, 可以节省大量的运行时 间. 因此 STC 引入了快速傅里叶变换操作来提高速 度. 对式 (6) 两边同时进行快速傅里叶变换, 可以得 到

$$F(c(x)) = F(h^{sc}(x)) \odot F(I(z)\omega_{\sigma}(z-x^*)) \quad (7)$$

其中, *F* 代表快速傅里叶变换, ⊙ 表示对应像素相 乘. 再对上述公式进行一个简单的变换, 即可得到空 间上下文模型的计算公式, 即

$$h^{sc}(x) = F^{-1}\left(\frac{F(c(x))}{F(I(z)\omega_{\sigma}(z-x^*))}\right) = F^{-1}\left(\frac{F\left(b \times \exp\left(-\left|\frac{x-x^*}{\alpha}\right|^{\beta}\right)\right)}{F\left(I(z)\omega_{\sigma}(z-x^*)\right)}\right)$$
(8)

1.4 STC 算法视觉跟踪过程

STC 在得到了空间上下文模型之后,其后续的 跟踪任务其实就是一个简单检测问题.不失一般性, 假设第 t 帧中的空间上下文用 $h_t^{sc}(x)$ 来表示,则 t+ 1 帧中目标中心 x_{t+1}^* 可以通过求取第 t+1 帧的 置信图的极值所在位置得到,用公式表示为

$$x_{t+1}^* = \arg \max_{x \in \Omega_c(x_t^*)} c_{t+1}(x)$$
(9)

其中, $c_{t+1}(x)$ 为第 t+1 帧的置信图, 其计算式为

$$c_{t+1}(x) = F^{-1}\left(F(H_{t+1}^{sc}(x)) \odot F(I_{t+1}(x)\omega_{\sigma}(z-x_{t}^{*}))\right)$$
(10)

此处的 *H*^{sc}_{t+1}(*x*) 是时空上下文模型,由每帧中的空间上下文模型通过在线更新的方式得到,其公式为

$$H_{t+1}^{sc}(x) = (1-\rho)H_t^{sc}(x) + \rho h_t^{sc}(x)$$
(11)

其中, ρ 称为学习率. 通过上面一系列的公式, 就得 到了 STC 的完整的跟踪过程. 如果需要了解更多的 细节, 可直接参考文献 [14].

2 基于 WSTC 的新型视觉跟踪算法

针对 STC 算法的不足,本文提出了对上下文环 境进行分区处理的思想.对于跟踪来说,辅助区域相 对于其他两种区域更加重要,能够提供更多的有用 信息,因此应该对这些区域给予更多的关注.因此, 本文设计了一种算法来提取与上下文对应的权值矩 阵,使得辅助区域内的像素对应着较高权值,而其他 两个区域内像素对应着较低甚至是零权值.最后再 把该权值矩阵整合进 STC 算法框架,从而实现了基 于加权时空上下文 (WSTC)的跟踪算法.

2.1 加权空间上下文模型

在 STC 中, 需要学习的空间上下文模型即为式 (1) 中的条件概率 *P*(*x*|*v*(*z*),*o*). 而在本文中, 由于 考虑了目标周围上下文环境中不同区域对跟踪的贡 献权值, 因此把该条件概率 *P*(*x*|*v*(*z*),*o*) 重新定义 为

$$P(x|v(z), o) = h^w(x - z) \times W(z)$$
 (12)

其中, h^w(x - z) 即为带加权的空间上下文模型, W 为上下文对应的权值矩阵, W(z) 为 z 点位置的相 应权值. 后面会具体介绍如何获得该权值矩阵. 将式 (2)~(4) 和式 (12) 代入式 (1) 中, 可以得 到

$$c(x) = \sum_{v(z)\in X^c} P(x|v(z), o)P(v(z)|o) =$$
$$\sum_{v(z)\in X^c} h^w(x-z)W(z)I(z)\omega_\sigma(z-x^*) =$$
$$h^w(x)\otimes (W(x)I(x)\omega_\sigma(x-x^*))$$
(13)

其中, ⊗ 表示卷积操作. 对式 (13) 两边同时进行快速傅里叶变换, 可以得到

$$F(c(x)) = F(h^{w}(x)) \odot F(W(x)I(x)\omega_{\sigma}(x-x^{*}))$$
(14)

其中, *F* 代表快速傅里叶变换, ⊙ 表示对应像素相 乘. 再对上述公式进行一个简单的变换, 即可得到加 权的空间上下文模型的计算公式, 即

$$h^{w}(x) = F^{-1}\left(\frac{F(c(x))}{F(W(x)I(z)\omega_{\sigma}(z-x^{*}))}\right) = F^{-1}\left(\frac{F\left(b \times \exp\left(-\left|\frac{x-x^{*}}{\alpha}\right|^{\beta}\right)\right)}{F(W(x)I(z)\omega_{\sigma}(z-x^{*}))}\right)$$
(15)

其中, h^w(x) 为加权的空间上下文模型, 它表示了上下文环境中的每个像素与目标中心的空间关系, 同时也考虑了上下文环境的权值.

2.2 上下文权值矩阵 W

本文根据上下文环境中不同区域内像素的运动 轨迹与目标中心的运动轨迹之间的相似性来决定哪 些区域对跟踪目标的定位作用更大. 那些与目标运 动相似度比较高的辅助区域的作用通常比较大,需 要赋予相对较高的权值, 而那些无关区域和干扰区 域,几乎提供不了多少有用的信息,则赋予相对较 小的权值,最终形成一个与上下文环境大小一致的 权值矩阵.为了获得上下文环境中每个像素的运动 信息,本文采用光流跟踪算法来估计前后帧之间对 应点的位移,从而估算出其运动轨迹.目前主要有 两种类型光流跟踪算法,一种是稠密的光流跟踪算 法, 跟踪时同时考虑了跟踪图像内的所有像素, 得 到的是一种全局最优解,虽然效果很好,但运行速 度十分缓慢,通常跟踪一帧图像需要花费数秒的时 间[15]. 如果运用到跟踪中, 会极大地拖慢整个跟踪 算法的运行速度,无法达到实时跟踪的目的.另一 种是稀疏光流跟踪,由于只跟踪一定数量的离散点, 速度相对稠密光流来说要快得多,其中最著名的就 是 Lucas-Kanade 算法.在本文提出的 WSTC 算法 中,权值矩阵本质上来说是稠密的,但是使用稠密光 流跟踪算法,虽然可以直接得到各像素点的权值,但 由于速度太慢,无法达到实时跟踪.因此本文最终决 定采用 Lucas-Kanade 稀疏光流算法来确定部分离 散点权值,然后再通过插值的方法得到整个上下文 环境的权值矩阵.最终结果表明,采用这种方案既保 留了较高的跟踪速度,又达到了非常理想的效果.

2.2.1 选择适合的特征点

实施稀疏光流跟踪首先要选择一定数量的特征 点. 目前经常使用的特征点包括: Harris 角点、Shi-Tomasi 提出的 Goodfeaturestotrack 特征点、 SIFT 特征点、SURF 特征点以及随机像素作为 特征点等. Nourani-Vatani 等在其论文中对这些 特征点用于光流跟踪时的表现进行了完整地比较, 包括特征点的获取时间和获得数量,以及可跟踪性 等^[16]. 文章对比结果表明, 用随机选择像素作为特 征点在获取时间、获得数量方面占有很大的优势,但 其可跟踪性比较差,非常容易受到噪声等因素的干 扰, 跟丢的概率很高. 而 SIFT 和 SURF 在获得特 征点的同时,还可以得到特征点的一个描述向量,在 特征点表示上占据很大优势,但其处理时间比较长, 而且它们在可跟踪性方面的表现也比较一般.综合 来看, Harris 角点具有较快的处理时间, 同时获得特 征点的数量及可跟踪性都有所保证,因此本文采用 Harris 角点作为特征点来实施 Lucas-Kanade 稀疏 光流跟踪.

2.2.2 上下文权值矩阵的计算过程

利用光流跟踪获得上下文环境的权值矩阵的过 程如下:首先,我们在第t-1帧提取一定数量的 Harris角点,可以表示为一个集合: $C_{t-1} = \{x_{t-1}^{t}, x_{t-1}^{2}, \dots, x_{t-1}^{n}\},$ 其中 x_{t-1}^{i} 表示第i个Harris角点, n表示 Harris角点的数量,下标表示第t-1帧. 然 后通过Lucas-Kanade光流算法,可以得到第t帧 中相应的跟踪结果集合: $C_{t} = \{x_{t}^{1}, x_{t}^{2}, \dots, x_{t}^{n}\}$.由 于跟踪过程中会产生部分跟踪错误点,因此首先需 要去掉这些错误点,具体操作方法在第3节介绍.去 掉错误跟踪的点之后,即可计算出每个角点的位移 向量,表示为 $v_{t}^{i} = x_{t}^{i} - x_{t-1}^{i}$,通过该位移向量 v_{t}^{i} ,就 知道了该角点的运动方向与速度.同样,计算出目标 中心的运动方向与速度,用公式表示为: $v_{t}^{*} = x_{t}^{*} - x_{t-1}^{*}$,其中 x_{t}^{*} 表示第t帧中目标中心位置的坐标.

接下来需要确定每个 Harris 角点的运动轨迹与 目标中心的运动轨迹之间的相似性. 首先, 我们计算 目标中心位移向量 v_t^* 与每一个 Harris 角点位移向 量 v_t^i 之间的距离, 为了计算方便, 需要先把位移向 量转换成极坐标形式,即

$$p_t^i = (r_t^i, \theta_t^i) = \left(\sqrt{(\mathrm{d}x)^2 + (\mathrm{d}y)^2}, \ a \tan 2\frac{\mathrm{d}y}{\mathrm{d}x}\right)$$
(16)

式中, p_t^i 为位移向量 v_t^i 的极坐标表示, r_t^i 和 θ_t^i 分 别为位移向量的长度和角度. dx 和 dy 分别为位移 向量在 x 轴和 y 轴方向的分量. a tan 2 是反正切函 数,该公式计算得到的角度 $\theta \in (-\pi, +\pi]$. 之后,目 标中心与 Harris 角点之间的距离即可定义为目标中 心位移向量与 Harris 角点位移向量之间的差,其公 式表示为

$$d_t^i = \operatorname{dis}(p_t^*, p_t^i) = \sqrt{(r_t^*)^2 + (r_t^i)^2 - 2r_t^* r_t^i \cos(\theta_t^* - \theta_t^i)} \quad (17)$$

最后,我们再把每个角点与目标中心的距离转 换成一个权值的形式,公式为

$$W_t^i = \frac{1}{\sqrt{2\pi\gamma}} \exp\left(-\frac{(d_t^i)^2}{2\gamma^2}\right) \tag{18}$$

式中, W_t 为第 t 帧的上下文权值矩阵, W_t^i 为第 i 个 角点 (像素点) 对应的权值, d_t^i 为上面步骤中计算出 来的距离, 而 γ 为尺度参数. 从式 (18) 不难看出: 由于辅助区域内像素的运动轨迹与目标中心的运动 轨迹大致相似, 其相对距离 d_t^i 就比较小, 所以这些 像素就可以获到一个较高的权值. 相反, 位于其他 区域内的像素, 其获得的权值就小得多. 图 1 (b) 是 WSTC 算法执行过程中在图 1 (a) 所在帧中计算得 到的上下文权值矩阵, 灰度越浅表示权值越高, 可见 那些与目标运行轨迹相似的区域都获得了较高的权 值.

需要注意的一个细节是,由于在算法中提取的 Harris 角点是稀疏的,为了得到整个上下文权值矩 阵,本文采用了二维线性插值的方法,从而获得了上 下文区域内所有像素的稠密权值矩阵.

2.3 基于 WSTC 的视觉跟踪

经过前面的计算步骤得到第t帧的权值矩阵 W_t 后,我们可以参考 STC 框架,把跟踪问题转为一个 在当前帧中寻找置信图极大值的问题.具体过程如 下:在t+1帧中,通过式 (9)计算得到当前帧的置 信图的极大值,这样目标中心位置 x_{t+1}^* 也就可以确 定.不过在 WSTC 算法中,由于加入权值矩阵,因 此置信图的计算公式也会与式 (10)稍有差别,这里, 考虑了权值矩阵的置信图计算公式可以描述为

$$c_{t+1}(x) =$$

$$F^{-1}(F(H_{t+1}^{sc}(x)) \odot F(I_{t+1}(x)\omega_{\sigma}(x-x_{t}^{*})W_{t}^{stc}))$$
(19)

其中, W^{stc} 是由每一帧的权值矩阵 W_t 形成的时空 权值矩阵.为了充分利用时间上下文信息,每帧中得 到的权值矩阵都要持续不断地进行在线更新,从而 形成时空上下文权值矩阵,其更新规则为

$$W_t^{stc} = (1 - \eta) W_{t-1}^{stc} + \eta W_t \tag{20}$$

这里, η 是学习率参数, W_1^{stc} 是第一帧的权值矩阵, 我们把其直接定义为一个所有元素均相等的规则化 矩阵.

3 错误跟踪点的消除

本文利用稀疏光流跟踪来获得上下文权值矩阵, 但光流跟踪中难免会有部分特征点被跟踪到错误位 置,这会使得计算出来的速度矢量不准确,从而降低 了权值矩阵的准确性,最终会对整个跟踪算法的性 能造成影响.

在TLD (Tracking-learning-detection) 算法中, 其作者提出了一种基于 FB 误差的错误跟踪点消除 算法,能够较好地去除光流跟踪中的错误点^[17].FB 误差先进行正向跟踪 (Forward tracking), 再进行反 向跟踪 (Backward tracking), 然后计算从跟踪起点 到跟踪终点之间的欧氏距离,即FB误差.当该误差 超一定阈值时,则认为该点是跟踪错误的点.图3展 示了 FB 误差的基本原理, 其中左侧的图像是跟踪 中的第t帧,右侧图像是第t+1帧. 当从第t帧跟 踪到第t+1帧时,称之为正向跟踪,反之称为反向 跟踪. 从图 3 中可以看到, 1 号点的正向跟踪结果是 4号点,同时4号点的反向跟踪结果是1号点,两次 跟踪的轨迹是一致的,跟踪起点与终点都是1号点, 因此其 FB 误差为 0. 而 2 号的正向跟踪结果是 5 号点,但从5号点开始的反向跟踪结果却是3号点, 存在 FB 误差, 其值越大, 说明本次跟踪越不可靠.



图 3 FB 误差展示 (本图来自于文献 [17]) Fig. 3 The forward-backward error (This figure comes from [17].)

FB 误差易于计算、适用范围广,获得了很大的 成功.但 FB 误差判断方法是基于一个隐含假设的, 即正向跟踪和反向跟踪都是可靠的.在实际情况下, 这种可靠性是难以保证的.由于光照、遮挡、变形、 模糊等因素的存在,一个特征点可能被跟踪到一个 错误的特征点,而目这种错误既可能发生在正向跟 踪过程,也可能发生在反向跟踪过程,这时FB误差 判断方法就会出现问题. 如图 4 所示, 图中两个虚 线框分别代表第t-1帧和t帧,每帧中分别有一 个深色点及一个浅色点,相同灰度的点代表前后两 帧中实际对应的点,即深色点的正确跟踪结果是深 色点, 浅色点的正确跟踪结果是浅色点. FB 误差先 进行正向跟踪再进行反向跟踪总共可以产生4种轨 迹: 1) 如图 4(a) 所示, 从第 t-1 帧中的深色点开 始的正向跟踪与反向跟踪得到了相同的轨迹,此时 FB 误差为 0, 通过 FB 误差可判定本次跟踪是正确 的, 与实际情况相符. 2) 在图 4(b) 中, 正向跟踪从 t-1 帧中的深色点跟踪到了 t 帧中的深色点,得到 正确的跟踪结果.反向跟踪从 t 帧中的深色点跟踪 到了 t-1 帧中的浅色点, 跟踪发生错误. 此时就会 产生一个较大的 FB 误差, 如果该误差超过了规定 阈值, t 帧中的深色点就会被当作跟踪错误点而抛 弃,实际上这个点是不应该抛弃的.3)在图4(c)中, 正向跟踪从t-1帧中的深色点跟踪到了t帧中的 浅色点,发生跟踪错误,而反向跟踪从 t 帧中的浅色 点正确跟踪到了 t-1 帧中的对应浅色点. 此时得到 的 FB 误差较大,因此 t 帧中的浅色点就会被当作错 误跟踪点而抛弃,与实际情况相符,4)在图4(d)中, t-1 帧中的深色点被错误跟踪到 t 帧中的浅色点, 而反向跟踪又从 t 帧中的浅色点错误跟踪到了 t -1 帧中的深色点. 但此时的 FB 误差却是 0, 因此 t 帧中的浅色点被当作正确跟踪点保留了下来,实际 上该点是错误跟踪点.



Fig. 4 Forward-backward tracking trajectories

通过上面的分析不难看出,在正向及反向跟踪 产生的4种轨迹情况中,FB误差可以正确处理的有 两种情况、即图 4 (a) 和图 4 (c) 两种情形, 而在另外 两种情况下均出现了错误. 通过仔细分析各种跟踪 轨迹可以发现一个简单的规律, 即每当跟踪发生错 误时, 跟踪的起点到终点的路径长度都发生了较明 显的变化. 如在图 4 中, 从深色点到浅色点之间的距 离要明显比从深色点到深色点之间的距离要长 (根 据实际情况, 距离也可能变短). 此时错误路径长度 与正确路径长度之间就存在着一个差值, 本文称之 为行程误差 (Journey error, JE). 行程误差可做如 下形式化描述: 设 t - 1 帧中的特征点是 x_{t-1} , 其在 第 t 帧中正确对应的特征点为 x_t^* , 实际跟踪到的特 征点为 x_t , 则行程误差定义为

$$JE(x_t) = \operatorname{dis}(J(x_{t-1}, x_t) - J(x_{t-1}, x_t^*)) \quad (21)$$

其中, $J(x_{t-1}, x_t)$ 表示从点 x_{t-1} 到点 x_t 之间的行 程向量. dis(x, y) 表示 x = y 之间的向量距离. 而 确定 t 帧中 x_t 是否为正确跟踪点, 可通过下式进行 判断:

$$is_valid(x_t) = \begin{cases} 1, & JE(x_t) < T \\ 0, & JE(x_t) \ge T \end{cases}$$
(22)

is_valid(*x*)为1表示该点是正确跟踪点,为0表示 该点为错误跟踪点,应抛弃,*T*为行程误差阈值.

需要注意的一个关键问题就是 t 帧中的正确位 置 x_t 是未知的, 故无法得到行程误差的准确数值.

为了解决这个问题,考虑到光流跟踪是同时对 大量特征点同时进行跟踪,而发生跟踪错误的点总 是少量的,因此可以利用邻近特征点的均值来替代 单个特征点的真值,这样就得到了相对准确的行程 误值,从而顺利完成了错误点的筛选工作.不仅如 此,行程误差只需计算跟踪起点与终点之间的欧氏 距离即可,省去了反向跟踪的步骤,因此相对 FB 误 差而言,减少了一半的计算量,进一步提高了效率.

4 实验评估及结论

为了验证 WSTC 算法的有效性,我们在 12 个 基准评测视频上对 WSTC 算法进行了对比实验.这 些视频来自吴毅老师在 2013 年 CVPR 顶级国际会 议中发表的目标跟踪评测文章^[18] 中提供的基准评 测视频库,所有视频及相关的 Ground truth 文件 均在相关网站¹ 提供自由下载.这 12 个评测视频 总帧数为 7215 帧,基本涵盖了非受控环境下经常 遇到的各种挑战性因素,包括剧烈的光照变化、复 杂的背景、严重遮挡、非刚体变形、目标旋转以及 运动模糊等,具有广泛的代表性.为了避免不同人 标注的 Ground truth 不同,从而影响跟踪器性能 等问题.本文全部采用了吴毅老师 2013 年 CVPR 目标跟踪评测文章^[18] 中提供的 Ground truth 以 及初始框的位置与大小,并保持所有算法跟踪时的 初始条件一致. 除了原始的 STC 算法之外,本文 还另外挑选了 11 个公开源代码或二进制可运行代 码的当前主流跟踪算法一起参加测试,以保证实验 的信度与效度. 这些算法包括: MIL (Multiple instance learning)^[1]、CT (Compressive tracker)^[19]、 MTT (Multi-task tracker)^[20]、DFT (Distribution field tracker)^[21]、VTD (Visual tracking decomposition)^[5]、IVT (Incremental subspace tracker)^[3]、OAB (Online AdaBoost tracker)^[22]、 TLD (Tracking-learning-detection)^[6]、LOT (Local orderless tracker)^[23]、Frag (Fragment-based tracker)^[24] 和 L1APG^[25].

本文实验代码是在 STC 算法公开的 Matlab 源 代码的基础上改写而成的,主要增加了光流跟踪模 块及权值矩阵计算模块,相应增加的计算代价主要 来自于光流跟踪模块,但由于我们选用的是运行效 率比较高的 Lucas-Kanade 稀疏光流跟踪算法(该 光流跟踪算法也被用于 TLD 算法^[6]),因此基本保 证了 WSTC 算法的实时性.本文的实验机器配置 为: Intel i5 3.2 GHz 中央处理器、4 GB 内存,软件 环境为 Win7 + Matlab 2011b,实验测试中 WSTC 算法的平均运行速度约为 23 帧/秒.

4.1 实验参数

在实验过程中, WSTC 算法中的各个参数设置 如下:式(4)中的参数 $\alpha = 2.25$, $\beta = 1$.式(11)中 的参数 $\rho = 0.075$,这些参数值与原始STC 算法中 的参数相同.对于本文新出现的公式,式(18)中的 参数设置为 $\gamma = 2$,式(20)中的参数设置为 $\eta = 0.3$, 式(22)中的参数设置为T = 2.其他对比算法均使 用了其公开代码中设置好的默认参数.这些参数设 定之后,在整个实验过程中保持不变.

4.2 实验结果

本文使用了当前比较流行的三种评测标准来量 化评价本文提出的 WSTC 算法的性能,这三个评 测标准分别是:中心位置误差 (Centre location error)、跟踪重叠率 (Tracking overlap ratio) 和跟踪 成功率 (Tracking success rate).中心位置误差是指 跟踪得到的目标中心位置与 Ground truth 标定的 目标中心位置之间的欧氏距离,表1给出了13个跟 踪器在12个评测视频中的中心位置误差,以及每个 跟踪器在全部视频中的平均跟踪误差.另外两个评 测标准是跟踪重叠率及跟踪成功率,跟踪重叠率定 义为 $R = Area(B_t \cap B_g)/Area(B_t \cup B_g), 其中 B_t$

¹http://www.visual-tracking.net

是跟踪器跟踪得到的目标框, B_g 是事先标注好的 Ground truth 目标框. $\cup 和 \cap 分别表示两个目标框$ 的并集与交集, *Area*(·) 表示区域的面积. 通常, 当 某一帧中根据跟踪结果得到的重叠率 *R* 超过 0.5 时, 则认为此帧的跟踪是成功的. 而视频中跟踪成功的 帧的数量与总帧数的比值, 即为跟踪成功率. 在表 2 和表 3 中, 分别列出了利用跟踪重叠率及跟踪成功 率标准计算得出的量化跟踪结果. 从表 1、表 2 和 表 3 中的实验统计结果可以看到, 在全部 12 个视频 中, 本文提出的 WSTC 算法在三种测量标准下均优 于原始的 STC 算法. 不仅如此, 与其他 11 个当前 主流的跟踪算法相比, 本文的 WSTC 算法也占有较 大优势, 在大部分实验视频中都取得了相对最好成 绩, 而且在三种测量标准下, WSTC 算法在所有视 频上的平均成绩均是最优秀的. 这表明本文提出的 WSTC 算法是合理有效的, 最终使 WSTC 算法达

表 1 跟踪结果的平均中心误差 (像素) (最优结果用粗体显示) Table 1 Average center location error (in pixels) (The best results are shown in bold fonts.)

Sequence	CT	MTT	DFT	VTD	MIL	OAB	TLD	IVT	LOT	Frag	L1APG	STC	WSTC
fish	10.7	45.5	8.8	16.8	24.1	87.0	6.5	5.7	33.6	21.6	29.4	6.4	4.5
coke	40.5	30.0	70.7	68.6	46.7	35.9	25.1	83.0	69.4	124	50.4	17.7	15.6
david	10.5	33.1	42.9	11.6	16.9	21.7	5.1	4.8	23.9	82.1	14.0	8.2	3.9
david2	76.7	1.7	17.3	2.9	10.9	33.8	5.0	1.2	4.1	56.9	1.4	4.3	3.9
girl	18.9	4.3	24.0	8.6	13.7	3.7	9.8	22.5	22.8	20.7	2.8	11.8	6.5
boy	9.0	12.8	106	7.6	12.8	2.9	4.5	91.3	66.0	40.5	7.0	21.9	9.5
mountainBike	214	7.1	155	9.8	73.0	12.0	216	7.7	24.9	206	8.3	7.2	7.1
jumping	47.7	84.6	67.1	41.4	10.0	45.3	5.9	61.6	5.6	5.6	83.7	93.6	3.9
shaking	80.0	97.9	26.3	9.0	24.0	191	37.1	85.7	82.6	192	109	17.0	6.4
sylvester	8.6	7.6	44.9	19.6	15.2	14.8	7.3	34.2	11.3	15.0	26.2	11.0	6.8
woman	114	137	8.5	118	125	31.4	139	176	117	111	128	9.4	10.2
mhyang	13.3	3.1	9.1	4.4	20.4	7.4	9.5	1.9	113	12.5	3.2	6.1	4.1
Average	53.7	38.7	48.4	26.6	32.8	40.6	39.3	48.0	47.9	74.2	38.8	17.9	6.9

表 2 跟踪结果的重合率 (最优结果用粗体显示)

Table 2 Tracking overlap ratio (The best results are shown in bold fonts.)

Sequence	CT	MTT	DFT	VTD	MIL	OAB	TLD	IVT	LOT	Frag	L1APG	STC	WSTC
fish	0.72	0.16	0.76	0.56	0.45	0.04	0.81	0.77	0.23	0.55	0.34	0.78	0.83
coke	0.23	0.44	0.11	0.14	0.20	0.33	0.40	0.12	0.12	0.04	0.17	0.50	0.60
david	0.50	0.29	0.30	0.56	0.43	0.39	0.72	0.64	0.26	0.17	0.54	0.52	0.54
david2	0.00	0.86	0.54	0.69	0.45	0.32	0.69	0.70	0.60	0.24	0.86	0.71	0.74
girl	0.31	0.66	0.28	0.55	0.40	0.72	0.57	0.17	0.42	0.45	0.73	0.54	0.62
boy	0.59	0.50	0.40	0.63	0.49	0.79	0.66	0.26	0.53	0.39	0.73	0.54	0.67
mountainBike	0.14	0.74	0.30	0.70	0.45	0.63	0.20	0.73	0.58	0.13	0.74	0.71	0.72
jumping	0.04	0.10	0.11	0.12	0.52	0.07	0.66	0.12	0.58	0.67	0.15	0.05	0.73
shaking	0.10	0.04	0.64	0.71	0.43	0.01	0.39	0.03	0.13	0.08	0.08	0.57	0.74
sylvester	0.67	0.65	0.38	0.62	0.53	0.56	0.67	0.52	0.57	0.58	0.40	0.61	0.71
woman	0.13	0.17	0.76	0.14	0.16	0.48	0.13	0.15	0.09	0.15	0.16	0.66	0.69
mhyang	0.60	0.85	0.71	0.73	0.51	0.75	0.63	0.80	0.22	0.65	0.83	0.76	0.79
Average	0.34	0.46	0.44	0.51	0.42	0.42	0.55	0.42	0.36	0.34	0.48	0.58	0.70

表 3 跟踪结果的成功率 (最优结果用粗体显示)

Sequence	CT	MTT	DFT	VTD	MIL	OAB	TLD	IVT	LOT	Frag	L1APG	STC	WSTC
fish	0.89	0.04	0.86	0.64	0.39	0.04	0.96	1.00	0.11	0.55	0.04	1.00	1.00
coke	0.09	0.62	0.09	0.14	0.12	0.17	0.29	0.13	0.09	0.03	0.20	0.47	0.81
david	0.43	0.29	0.23	0.68	0.23	0.15	0.97	0.80	0.15	0.12	0.69	0.56	0.63
david2	0.00	1.00	0.54	0.99	0.32	0.25	0.95	0.93	0.77	0.30	1.00	0.96	0.99
girl	0.18	0.93	0.25	0.65	0.29	0.94	0.76	0.19	0.55	0.54	0.97	0.69	0.80
boy	0.69	0.49	0.48	0.79	0.39	0.99	0.94	0.33	0.65	0.46	0.90	0.65	0.79
mountainBike	0.17	0.96	0.35	1.00	0.57	0.91	0.26	0.98	0.69	0.14	0.92	0.96	0.96
jumping	0.01	0.09	0.12	0.11	0.48	0.05	0.85	0.10	0.78	0.85	0.12	0.05	0.99
shaking	0.04	0.01	0.82	0.94	0.23	0.01	0.40	0.01	0.08	0.07	0.04	0.53	0.98
sylvester	0.83	0.82	0.41	0.80	0.55	0.68	0.93	0.68	0.68	0.68	0.43	0.69	0.84
woman	0.16	0.20	0.93	0.18	0.19	0.61	0.17	0.18	0.08	0.18	0.20	0.86	0.93
mhyang	0.73	1.00	0.78	0.95	0.39	0.96	0.89	1.00	0.26	0.72	0.97	0.97	1.00
Average	0.35	0.54	0.49	0.66	0.34	0.48	0.70	0.53	0.41	0.39	0.54	0.70	0.89





到甚至超过了当前主流的跟踪算法.在随后的图 5、 图 6 及图 7 中,给出了实验过程中部分视觉跟踪结 果的屏幕截图.

4.2.1 光照变化及复杂背景

在参加实验的评测视频中, shaking 视频存在剧

烈的光照变化,视频背景也相当杂乱,这给跟踪算法 带来了相当大的挑战.从图 5 第 1 行的视频截图中 可以看到,仅仅到了 #10 帧,OAB 算法就已经完全 漂离了目标.而在后面的 #62 帧、#140 帧、#270 帧及 #365 帧里,绝大多数跟踪算法都发生了严重 的漂移.在此视频中,超过 50% 跟踪成功率的算法



图 6 部分视觉跟踪结果截图 Fig.6 The tracking results on some benchmark video sequences



图 7 部分视觉跟踪结果截图

Fig. 7 The tracking results on some benchmark video sequences

只有 WSTC、DFT、STC 及 VTD 四种. WSTC 算 法在该视频中表现得最好,取得了98%的跟踪成功 率. 这是由于 WSTC 是在频率域对目标进行建模, 因此受光照的影响比较小,而 STC 算法虽然也是在 频率域进行建模,但其他受到目标周围较杂乱的背 景影响较大, 而 WSTC 通过加权矩阵, 自动选择了 目标周围的有效上下文区域,从而获得了比 STC 更 好的跟踪性能. 在图 5 第 2 行所示的 sylvester 视频 中包含了光照变化、复杂背景以及旋转等多种因素, 但在该视频中,大部跟踪算法都得到了较好的跟踪 结果 (参见 #467 帧、#547 帧及 #616 帧). 13 个算 法中,有10个算法的跟踪成功率大于或等于68%, 这可能与该视频中目标运动范围小且背景变化不大 有关. 本文的 WSTC 算法在该视频上取得了 93% 的跟踪成功率,在全部算法中排名第二.在 fish (参 见 #22 帧、#326 帧及 #428 帧) 及 mhyang (参见 #701 帧、#830 帧及 #849 帧) 视频中, 也包含了 一定的光照变化及复杂背景等因素,但 WSTC 算法 在两个视频上均取得了最佳成绩,两个视频都达到 了100%的跟踪成功率.

4.2.2 快速运动及运动模糊

图 6 中的第1 行给出了 jumping 视频的实验截 屏, jumping 视频包含了快速运动及较大程度的运 动模糊.此视频的跟踪目标是人脸,目标本身比较 小,再加上做大范围的运动及运动模糊,给跟踪带来 了较大的困难,因此大部分算法在此视频上的表现 都非常差,从#102帧、#176帧、#250帧及#311 帧可以明显看出,除TLD、LOT、Frag及WSTC之 外,其他跟踪器均完全脱离了目标.而 WSTC 算法 因为考虑了目标周围的上下文环境,可以借助于人 的身体部分来定位目标,因此取得了良好的跟踪效 果. boy 视频中也包含了目标的快速运动、运动模糊 及旋转等困难因素,从#248 帧、#442 帧及#484 帧帧可以看到 IVT、LOT、Frag、STC 等算法已经 完全跟丢, 而本文的 WSTC 算法一直能较好地跟 踪到目标. 图 6 中的 david (参见 #496 帧、#600 帧及 #643 帧) 及 david2 (参见 #299 帧、#366 帧 及 #466 帧) 视频, 也分别包含了运动模糊及快速 运动等因素,同时还包括了许多其他挑战性因素,如 旋转、遮挡及变形等. 而在这两个视频中, 本文的 WSTC 算法都表现的不错, 尤其在 david2 视频中, 取得了99%的跟踪成功率.

4.2.3 目标遮挡及目标旋转

在非受控环境下,经常存在目标被遮挡的情形. 在图 7 的 coke 视频中,当人拿着可乐瓶在一簇绿 叶后面不断移动时,可乐瓶经常被绿叶遮挡.例如 在 #75 帧、#166 帧及 #188 帧中均有不同程度的 遮挡, 而在 #255 帧中, 目标几乎被完全遮挡住了. 从这些截屏中可以看到,此情形下许多跟踪器都丢 失了目标,能较好跟踪目标的只有 MTT、STC 及 WSTC 三种算法. 而这三种算法中, 又以 WSTC 的跟踪效果最好,达到了81%的成功率.这是由 于当遮挡发生时,由于遮挡区域与目标本身运动 轨迹不一致,因此遮挡区域对应的权值非常小,而 其他区域对应的权值相对较大,相当于进行目标 跟踪时更多地参考了非遮挡区域的信息,从而增加 了对遮挡的鲁棒性. 在 girl 视频中, 既存在着遮 挡,也存在着平面旋转 (In-plane rotation) 及离面 旋转 (Out-of-plane rotation). 从 #90 帧、#309 帧、#331 帧及 #436 帧可以看到,虽然该视频发生 了多次遮挡及各种类型的旋转, WSTC 全部能较好 地跟踪到目标,并最终取得了80%的跟踪成功率. 而 CT、DFT、MIL、IVT、LOT 及 Frag 等算法的 跟踪成功率均未达到 60%. 图 7 第 3 行及第 4 行 展示了 woman 及 mountainBike 视频,其分别包括 了遮挡及目标旋转等困难因素. 在 woman 视频的 #123 帧、#142 帧及 #368 帧中可以看到, 虽然发 生了遮挡, 但并未影响 WSTC 算法的跟踪效果, 最 终 WSTC 算法以 93% 的跟踪成功率位居榜首,其 他两个跟踪效果较好的算法分别为 DFT 及 STC. mountainBike 视频中主要是目标平面旋转,在#78 帧、#114 帧、#166 帧及 #199 帧中可以看到,本 文的 WSTC 算法均取得了较好的跟踪效果.

5 结论

本文提出了一个基于加权时空上下文的鲁棒视 觉跟踪算法. 该算法对目标周围上下文区域内的信 息进行了深层次处理,根据不同区域的运动特征给 出不同的权值,以实现更鲁棒的跟踪.本文主要思 路是按照上下文中每个区域与目标本身之间的运动 相似性,把整上下文区域分为三类,即辅助区域、无 关区域和干扰区域,然后根据不同区域内像素与目 标本身的运动相似程度生成整个上下文的权值矩阵, 并参照 STC 跟踪框架, 重新定义了加权时空上下 文模型,从而形成了本文的 WSTC 算法. 该算法 自动确定上下文环境中不同区域的权值,能够充分 利用辅助区域信息提升跟踪性能,同时压制了其他 区域带来的负面影响,达到了更鲁棒的视觉跟踪目 的. 文章最后大量的实验结果表明, 本文的 WSTC 算法,性能更加优秀,超过了原始STC算法及其他 一些当前主流算法, 验证了本文思想的有效性. 另 外,在算法的实现过程中,本文仔细分析了FB误 差,并针对其不足,提出了一个基于行程误差的简 单有效的跟踪错误点的消除方法,亦取得了良好效 果.

References

- 1 Babenko B, Yang M H, Belongie S. Robust object tracking with online multiple instance learning. *IEEE Transactions* on Pattern Analysis and Machine Intelligence, 2011, **33**(8): 1619–1632
- 2 Wang Li-Jia, Jia Song-Min, Li Xiu-Zhi, Wang Shuang. Person following for mobile robot using improved multiple instance learning. Acta Automatica Sinica, 2014, 40(12): 2916 –2925
 (Tight With A FEW TRANSFORM TRANSFORM CONTRACTOR AND ADDRESS OF A DRESS OF

(王丽佳, 贾松敏, 李秀智, 王爽. 基于改进在线多示例学习算法的机器人目标跟踪. 自动化学报, 2014, 40(12): 2916-2925)

- 3 Ross D A, Lim J, Lin R S, Yang M H. Incremental learning for robust visual tracking. International Journal of Computer Vision, 2008, 77(1-3): 125-141
- 4 Zhang K H, Zhang L, Yang M H. Fast compressive tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(10): 2002-2015
- 5 Kwon J, Lee K M. Visual tracking decomposition. In: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). San Francisco, CA, USA: IEEE, 2010. 1269–1276
- 6 Kalal Z, Mikolajczyk K, Matas J. Tracking-learningdetection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409–1422
- 7 Li Zhen-Xing, Liu Jin-Mang, Li Song, Bai Dong-Ying, Ni Peng. Group targets tracking algorithm based on box particle filter. Acta Automatica Sinica, 2015, 41(4): 785-798 (李振兴, 刘进忙, 李松, 白东颖, 倪鹏. 基于箱式粒子滤波的群目标 跟踪算法. 自动化学报, 2015, 41(4): 785-798)
- 8 Zhou X Z, Lu Y, Lu J W, Zhou J. Abrupt motion tracking via intensively adaptive Markov chain Monte Carlo sampling. *IEEE Transactions on Image Processing*, 2012, **21**(2): 789–801
- 9 Zhou T F, Lu Y, Di H J. Nearest neighbor field driven stochastic sampling for abrupt motion tracking. In: Proceedings of the 2014 International Conference on Multimedia and Expo (ICME). Chengdu China: IEEE, 2014. 1–6
- 10 Grabner H, Matas J, Van Gool L, Cattin P. Tracking the invisible: learning where the object might be. In: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). San Francisco, CA, USA: IEEE, 2010. 1285–1292
- 11 Dinh T B, Vo N, Medioni G. Context tracker: exploring supporters and distracters in unconstrained environments. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Colorado Springs, CO, USA: IEEE, 2011. 1177–1184
- 12 Wen L Y, Cai Z W, Zhen L, Dong Y, Li S Z. Online spatiotemporal structural context learning for visual tracking. In: Proceedings of the 2012 European Conference on Computer Vision (ECCV). Florence, Italy: Springer, 2012. 716–729
- 13 Yang M, Wu Y, Hua G. Context-aware visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, **31**(7): 1195-1209
- 14 Zhang K H, Zhang L, Liu Q S, Zhang D, Yang M H. Fast visual tracking via dense spatio-temporal context learning. In: Proceedings of the 2014 European Conference on Computer Vision (ECCV). Czech Republic: Springer, 2014. 127–141
- 15 Sundaram N, Brox T, Keutzer K. Dense point trajectories by GPU-accelerated large displacement optical flow. In: Proceedings of the 2010 European Conference on Computer Vision (ECCV). Florence, Italy: Springer, 2010. 438-451
- 16 Nourani-Vatani N, Borges P V K, Roberts J M. A study of feature extraction algorithms for optical flow tracking. In: Proceedings of the 2012 Australasian Conference on Robotics and Automation. Victoria University of Wellington, New Zealand, 2012.

- 17 Kalal Z, Mikolajczyk K, Matas J. Forward-backward error: automatic detection of tracking failures. In: Proceedings of the 2012 International Conference on Pattern Recognition (ICPR). Istanbul Turkey: IEEE, 2010. 2756–2759
- 18 Wu Y, Lim J, Yang M H. Online object tracking: a benchmark. In: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Portland, OR, USA: IEEE, 2013. 2411–2418
- 19 Zhang K H, Zhang L, Yang M H. Real-time compressive tracking. In: Proceedings of the 2012 European Conference on Computer Vision (ECCV). Florence, Italy: Springer, 2012. 864–877
- 20 Zhang T X, Ghanem B, Liu S, Ahuja N. Robust visual tracking via multi-task sparse learning. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, RI, USA: IEEE, 2012. 2042-2049
- 21 Laura S L, Erik L M. Distribution fields for tracking. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, RI, USA: IEEE, 2012. 1910–1917
- 22 Grabner H, Grabner M, Bischof H. Real-time tracking via on-line boosting. In: Proceedings of the 2006 British Machine Vision Conference. 2006, 47–56
- 23 Oron S, Bar-Hillel A, Levi D, Avidan S. Locally orderless tracking. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, RI, USA: IEEE, 2012. 1940–1947
- 24 Adam A, Rivlin E, Shimshoni I. Robust fragments-based tracking using the integral histogram. In: Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2006. 798-805
- 25 Bao C L, Wu Y, Ling H B, Ji H. Real time robust L_1 tracker using accelerated proximal gradient approach. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, RI, USA: IEEE, 2012. 1830–1837



徐建强 北京理工大学计算机学院博士 研究生.主要研究方向为目标跟踪,计算 机视觉,模式识别.

E-mail: xujq@bit.edu.cn

(XU Jian-Qiang Ph. D. candidate at the School of Computer Science, Beijing Institute of Technology. His research interest covers object tracking,

computer vision, and pattern recognition.)



陆 耀 北京理工大学计算机学院教授. 主要研究方向为神经网络,图像和信号 处理,模式识别.本文通信作者.

E-mail: vis_ly@bit.edu.cn

(**LU Yao** Professor at the School of Computer Science, Beijing Institute of Technology. His research interest covers neural network, image and signal pro-

cessing, and pattern recognition. Corresponding author of this paper.)