

引用格式:景星烁,邹卫军,夏婷,等.融合HOG类特征的尺度自适应压缩跟踪算法[J].电光与控制,2018,25(8):54-59. JING X S, ZOU W J, XIA T, et al. An adaptive-scale compressive tracking algorithm fusing HOG-like features[J]. Electronics Optics & Control, 2018, 25(8):54-59.

融合HOG类特征的尺度自适应压缩跟踪算法

景星烁¹, 邹卫军¹, 夏婷¹, 李超¹, 许旭²

(1. 南京理工大学, 南京 210094; 2. 山东农业大学, 山东 泰安 271018)

摘要: 针对压缩跟踪算法中存在的特征单一、对光照变化敏感、不能适应目标尺度变化的问题,提出了一种融合HOG类特征的尺度自适应压缩跟踪算法。该算法在Haar-like特征基础上采用固定比值方式融合了HOG类特征,降低了算法对光照的敏感度,HOG类特征仍采用积分图算法进行加速计算。另外,在尺度估计方面,采用相关滤波尺度估计方法,找到使尺度滤波器响应最大的尺度作为新的目标尺度,及时调整跟踪窗的大小并更新特征提取模板,解决了尺度变化问题。实验结果表明:改进后的算法可以适应较大光照变化与尺度变化、平面内旋转等情况,跟踪精度和鲁棒性有明显提高,并且满足准实时性的要求。

关键词: 光照变化; HOG类特征; 尺度估计; 相关滤波

中图分类号: TP391.4 文献标志码: A doi:10.3969/j.issn.1671-637X.2018.08.011

An Adaptive-Scale Compressive Tracking Algorithm Fusing HOG-like Features

JING Xing-shuo¹, ZOU Wei-jun¹, XIA Ting¹, LI Chao¹, XU Xu²

(1. Nanjing University of Science and Technology, Nanjing 210094, China;

2. Shandong Agricultural University, Tai'an 271018, China)

Abstract: There are problems in traditional compressive tracking algorithms of few extracted features, high sensitivity to illumination changes and inadaptability to the changes in target scale. To solve these problems, an adaptive-scale compressive tracking algorithm fusing HOG-like features is proposed. On the basis of Haar-like features, the new algorithm uses the method of fixed ratio to fuse HOG-like features, so as to reduce its sensitivity to illumination. The integral graph algorithm is used to accelerate the calculation of HOG-like features. In addition, a scale estimation method based on correlation filter is proposed, so as to find out the scale to which the scale filter has the biggest response and then take it as the new target scale. The size of the tracking window is timely adjusted and the feature extraction template is updated. Therefore, the problem of scale change is solved. The experimental results show that the improved algorithm can adapt to marked illumination changes, scale changes and in-plane rotation. Its tracking accuracy and robustness are improved greatly, which satisfies the requirements of real-time tracking.

Key words: illumination change; HOG-like feature; scale estimation; correlation filter

0 引言

目标跟踪是在连续的图像序列中估计目标的位置,需要面临诸多困难,是一个具有挑战性的难题。最初的研究算法主要是基于生成模型来进行跟踪,如文献[1-2],但这些算法的鲁棒性不是太高,计算量很

大,不适合进行实时跟踪。随着目标跟踪算法的进一步发展,基于判别模型^[3]的跟踪成为研究热点。基于压缩感知理论^[4]的压缩跟踪算法^[5]就是基于判别模型,通过生成一个稀疏的投影矩阵,然后将压缩后的数据作为特征来进行分类,优点是简单、高效;但该算法仍存在一些缺陷,如对光照比较敏感,而且不能适应目标尺度的变化。随后,研究人员在压缩跟踪的基础上也做了一些改进。文献[6]以多个矩阵作为压缩感知中的投影矩阵,将压缩后的数据作为特征提取出跟踪所需的多种特征,使算法能够适应纹理和光照变化,但

收稿日期:2017-08-07

修回日期:2017-09-11

作者简介:景星烁(1994—),男,山东济宁人,硕士生,研究方向为视频图像处理及目标跟踪。

跟踪框大小固定,所以不能够适应目标尺度的变化。文献[7]通过结构约束性采样,获取不同位置不同尺度的扫描窗,离线计算不同尺度下的稀疏随机感知矩阵,并用于降维,解决了目标的基本姿态变化及尺度缩放的问题,但是没有解决光照变化、遮挡等问题。

本文综合考虑以上缺陷,提出了改进算法:分析了 HOG 类特征对光照的不敏感性,与 Haar-like 特征采用固定比值方式进行融合,并结合尺度估计相关滤波器^[8],寻找使滤波器得到最大响应的尺度,更新特征提取模板,从而实时地调整跟踪窗口进行精确跟踪。

1 CT 算法相关概念

压缩跟踪算法的主要思想就是利用符合压缩感知 RIP 条件的随机感知矩阵对高维图像特征进行降维,并且对降维后的特征采用简单的朴素贝叶斯分类器进行二分类,然后每帧通过在线学习更新分类器,实现目标跟踪。基于压缩感知理论^[4]的降维公式可以描述为: $V=RX$,其中, X 为高维特征信号(m 维), V 为低维特征信号(n 维), R 代表随机测量矩阵。其特征降维过程为

$$R^{n \times m} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \quad (1)$$

2 Haar-like 特征与 HOG 类特征的融合

2.1 Haar-like 特征算法

Haar-like 特征^[5]是通过统计多个矩形框内的像素和,然后由这几个矩形框内的像素和线性表示而成,此特征在图像内的不同区域形成像素灰度特征,在目标发生一定形变时有较好的鲁棒性,但是该特征由于引入的是颜色信息,所以对光照比较敏感,不能应对较大的光照变化情况。

每个样本^[5]的 Haar-like 特征算式为

$$f(i) = \sum_{k=1}^N w_k \times R_{\text{rect}}(r_k) \quad (2)$$

式中: $f(i)$ 为第 i 个特征; $R_{\text{rect}}(r_k)$ 为第 k 个矩形框内像素值的和; w_k 为第 k 个矩形框的权重; N 是矩形特征 $f(i)$ 中含有的矩形框总个数,其值在 2~4 之间。

2.2 HOG 类特征算法

2005 年,DALAL 提出 HOG(梯度向量直方图)^[9]特征,是指使用梯度的幅值和方向围绕兴趣点或图像的一个区域构建直方图。HOG 特征表征了图像矩形区域内的梯度强度和梯度方向的分布情况,由于它是在图像的局部单元上操作的,所以对图像的几何和光

学上的形变有着良好的不变性,弥补了 Haar-like 特征的不足。

计算 HOG 特征时首先要计算像素的梯度与梯度方向。图像各像素点 (x,y) 的梯度算式为

$$G_x(x,y) = i(x+1,y) - i(x-1,y) \quad (3)$$

$$G_y(x,y) = i(x,y+1) - i(x,y-1) \quad (4)$$

式中, $G_x(x,y), G_y(x,y), i(x,y)$ 分别代表输入图像中像素点 $A(x,y)$ 水平方向上的梯度、垂直方向上的梯度和该点像素值。所以,该像素点的梯度值与梯度方向分别为

$$G = \sqrt{G_x^2 + G_y^2} \quad (5)$$

$$\theta = \arctan(G_y/G_x) \quad (6)$$

但是,使用上式直接计算梯度,计算量十分巨大,不能满足实时性的要求,所以实际计算时采用中心向量算子^[9] $[1,0,-1]$ 来估计梯度。

为了与 Haar-like 特征进行融合,本文提出一种有别于传统 HOG 特征的 HOG 类特征表示方法:在采集得到正负样本之后,首先计算每个样本图像中所有像素点的梯度和梯度方向。其次,把梯度方向平均分成 B 个特征区间,按照每个像素点的梯度方向统计各像素点的梯度向量直方图。最终,在每个特征区间,都能得到一幅梯度幅值图像,进而得到 B 幅梯度幅值图像。根据式(2),对每个梯度幅值图像分别采用与 Haar-like 相同的处理方法,便可以得到上文所提到的 HOG 类特征。其特征数量为 Haar-like 特征数量的 B 倍。

由于每个 Haar-like 特征与 HOG 类特征提取时要统计矩形框内像素的和,计算量是十分庞大的,无法满足实际要求。为了减少计算量,本文仍采用积分图像^[5]进行加速运算。

2.3 特征融合

在实际提取特征时,需要使用上述所提到的随机测量矩阵的方法进行降维。本文采用 Haar-like 特征与 HOG 类特征固定比值融合的方法,提取图像更多有用的信息。利用随机测量矩阵将高维特征 X 投影到低维空间。

$$v_i = (y_{i1}, y_{i2}, \dots, y_{iB}, z_i) \quad (7)$$

$$V = (v_1, v_2, \dots, v_n) = RX \quad (8)$$

式中: $(y_{i1}, y_{i2}, \dots, y_{iB})$ 代表 HOG 类特征值, B 代表每个融合特征组中 HOG 类特征个数; z_i 代表 Haar-like 特征值; V 代表压缩之后的特征向量,即通过随机测量矩阵 R 得到的所有融合特征组。本文为了保持算法的实时性,在特征融合过程中采用的是固定比值融合,即参与训练的 Haar-like 特征数与 HOG 类特征数的比值设置为 1; B (B 的大小可以根据不同场景和目标特征进行

适当调整),然后使用融合特征组训练和更新分类器并进行分类。

2.4 朴素贝叶斯分类器的构建与更新

对于上述所计算的 Haar-like 与 HOG 类特征组,本文仍采用朴素贝叶斯分类器进行分类,假定 V 中的各元素是独立分布的,构建分类器为

$$H(v) = \ln \frac{\prod_{i=1}^n p(v_i | y = 1) p(y = 1)}{\prod_{i=1}^n p(v_i | y = 0) p(y = 0)} = \sum_{i=1}^n \ln \frac{p(v_i | y = 1)}{p(v_i | y = 0)} \quad (9)$$

式中: $y \in \{0, 1\}$ 代表样本标签, $y = 0$ 表示负样本, $y = 1$ 表示正样本,假设两个类的先验概率相等, $p(y = 1) = p(y = 0) = 0.5$ 。DIACONIS 和 FREEDMAN 证明了高维随机向量的随机投影几乎都是高斯分布的。因此,假定在分类器 $H(v)$ 中的条件概率 $p(v_i | y = 1)$ 和 $p(v_i | y = 0)$ 也属于高斯分布,并且可以用 4 个参数 $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ 来描述,则

$$\begin{cases} p(v_i | y = 1) \sim N(\mu_i^1, \sigma_i^1) \\ p(v_i | y = 0) \sim N(\mu_i^0, \sigma_i^0) \end{cases} \quad (10)$$

这 4 个参数会进行增量更新,即

$$\begin{cases} \mu_i^1 \leftarrow \lambda \mu_i^1 + (1 - \lambda) \mu^1 \\ \sigma_i^1 \leftarrow \sqrt{\lambda (\sigma_i^1)^2 + (1 - \lambda) (\sigma^1)^2 + \lambda (1 - \lambda) (\mu_i^1 - \mu^1)^2} \end{cases} \quad (11)$$

式中:学习因子 $\lambda > 0$;

$$\begin{cases} \sigma^1 = \sqrt{\frac{1}{n} \sum_{k=0|y=1}^{n-1} (v_i(k) - \mu^1)^2} \\ \mu^1 = \frac{1}{n} \sum_{k=0|y=1}^{n-1} v_i(k) \end{cases} \quad (12)$$

式(11)可以由最大化似然估计^[5]得到。

3 自适应尺度相关滤波器的设计

3.1 滤波器的设计

针对目标运动过程中出现的尺度变化问题,本文借鉴多尺度跟踪方法^[10],通过学习一个独立的一维相关滤波器去检测目标尺度的变化,称之为尺度相关滤波器。尺度估计方法为:首先在检测出的目标周围采集一系列不同尺度的图像块 x_{si} ,提取 d 维 HOG 特征进行图像块表达,对应的高斯函数标签为 y_{si} ;利用这些图像块训练 RLS 分类器^[9],并在线性空间进行求解,获得一个一维的尺度相关滤波器;最后通过寻找这个滤波器的最大输出响应来完成尺度检测。

训练 RLS 分类器,得到尺度相关滤波器模板 B_{num} , B_{den} 的计算和更新公式分别为

$$B_{\text{num}(i)} = (1 - \eta_s) B_{\text{num}(i-1)} + \eta_s F(y_{si}) \odot F^*(x_{si}) \quad (13)$$

$$B_{\text{den}(i)} = (1 - \eta_s) B_{\text{num}(i-1)} + \eta_s \sum_{k=1}^d F(x_{si}^k) \odot F^*(x_{si}^k) \quad (14)$$

式中, η_s 是固定的参数。

尺度分类器的输出响应为

$$\hat{y}_s = F^{-1} \left[\frac{\sum_{k=1}^d B_{\text{num}}^k \odot F(z_s)}{B_{\text{den}} + \lambda} \right] \quad (15)$$

式中: F 为傅里叶变换; F^{-1} 为傅里叶逆变换; λ 为正则化参数; z_s 为窗口图像块;找到使 \hat{y}_s 最大的尺度即为新一帧的尺度。

随着目标尺度的变化,特征提取模板也要变化。本文采用比例缩放的方式更新模板,即通过缩放特征模板中小矩形框的大小来保证适应目标尺度的改变。

3.2 改进后的整体算法流程

改进后的算法流程如下:

- 1) 初始化有关参数^[5,10]并手动设定第一帧中被跟踪目标的中心位置及尺度;
- 2) 读取新帧图像,判断是否为初始帧,若是,转到 8);否则,转 3);
- 3) 在图像上一帧的目标中心位置的候选区域(距离半径 $r \leq \gamma$ ^[5])中提取候选样本,并计算 Haar-like 和 HOG 类特征组的特征值;
- 4) 使用朴素贝叶斯分类器从候选区域中找到使分类数最大的目标中心位置作为新位置;
- 5) 在新目标区域周围为尺度相关滤波器提取 33 个尺度样本^[10],并计算其特征值;
- 6) 计算尺度分类器的响应值,找到最大响应时的尺度;
- 7) 更新目标尺度,并更新特征模板;
- 8) 在图像目标中心的内邻域(距离半径 $r \leq \alpha$ ^[5])与环形域(距离半径 $\zeta \leq r \leq \beta$ ^[5])分别提取当前尺度的正样本和负样本,用稀疏测量矩阵在正负样本上提取 Haar-like 和 HOG 类特征组,并计算特征值;
- 9) 由 8) 得到的特征值更新朴素贝叶斯分类器参数;
- 10) 在目标区域周围为尺度相关滤波器提取 33 个尺度样本^[10],并计算样本特征值;
- 11) 更新尺度分类器的参数;
- 12) 判断是否为最后一帧,如果是,则算法终止;否则,转到 2)。

4 实验结果及分析

本文实验的计算机配置为:处理器为 Intel(R) Core(TM) i5-3230M CPU @ 2.60 GHz,内存为 8.00

GB,64 位 Windows8 操作系统。开发环境为 Matlab 与 C++ 的混合编译环境,版本为 Matlab 2014a。实验中所用到的 6 组视频序列来自公开的 OTB 数据集,每个视频序列的特点如表 1 所示。Struck 跟踪算法^[11]是基于结构输出预测的自适应目标跟踪框架,利用支持向量机(SVM)原理实现跟踪,其跟踪速度与精度都不错。实验中,对比了原始压缩跟踪算法和 Struck 跟踪算法,并在下文做出了具体分析。

表 1 各视频序列的特点

Table 1 Features of the video sequences

素材名称	特点
Dog1	尺度变化,平面内、外旋转
Dog2	尺度变化,平面外旋转,形变
CarDark	光照变化,背景相似干扰
Man	光照变化
Trellis	光照变化,尺度变化,背景相似干扰
Walking	尺度变化,人群干扰,部分遮挡

4.1 参数设置及评价指标

实验过程中,设置基本参数为:梯度区间 B 取 4, Haar-like 特征数仍然取原始压缩跟踪算法中的 50,学习率为 0.85, $\alpha = 4, \beta = 30, \zeta = 8, \gamma = 20$;有关尺度估计的参数设置见文献[10]。

中心距离精度 P_D 、重叠率 R_0 、中心位置误差 E_{cl} 是度量跟踪算法性能的重要指标。其中: E_{cl} 是跟踪目标位置与实际位置的欧氏距离,单位为像素; P_D 是 $E_{cl} < 20$ 的帧数占总帧数的比值; R_0 的定义为

$$R_0 = \frac{S(R_T \cap R_C)}{S(R_T \cup R_C)} \quad (16)$$

式中: R_T 为跟踪得到的目标框; R_C 为标定目标框; $S(\cdot)$ 表示矩形框 (\cdot) 的面积。当 R_0 大于 0.5 时,认为可以成功跟踪到目标。

4.2 定性结果比较

选取的 6 组视频序列分别用 Struck 算法、CT 算法及本文算法进行跟踪测试,跟踪效果对比如图 1 所示。

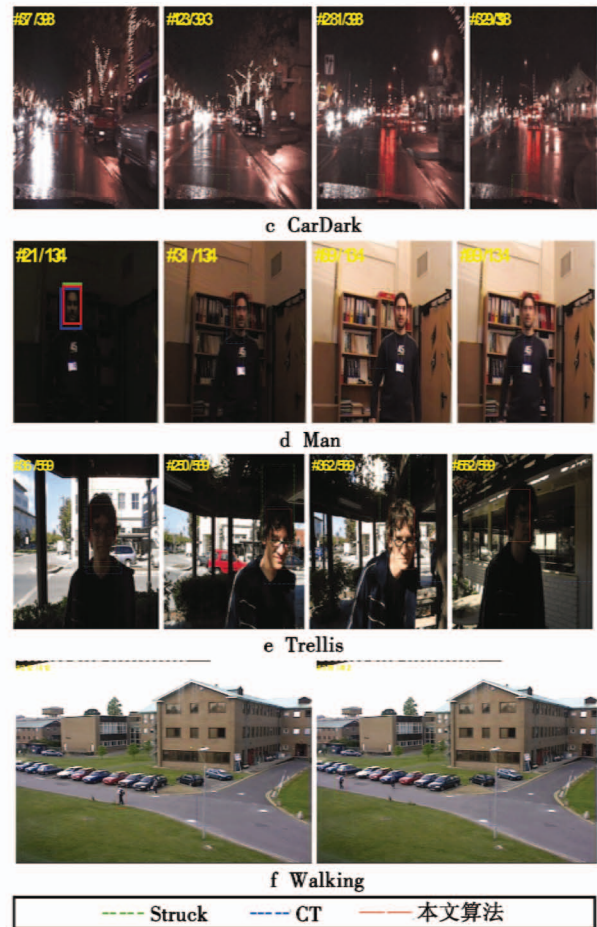
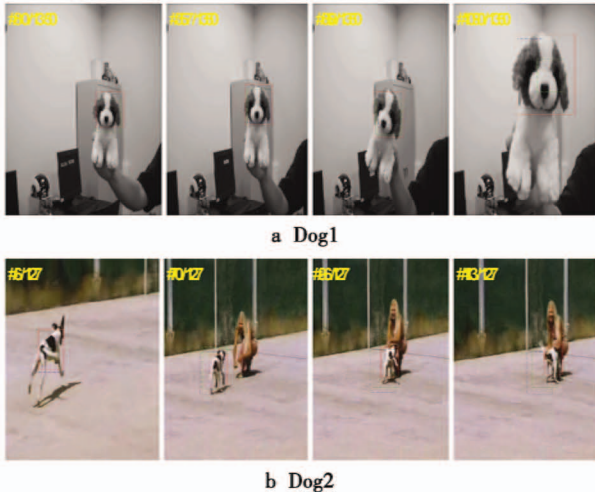


图 1 各视频序列跟踪结果对比

Fig. 1 The contrast of the tracking results of the video sequences

通过以上跟踪结果的定性对比,并且结合每个视频序列所具有的特点,可以分以下几种情况进行分析。

光照变化情况:图 1c、图 1d、图 1e 中所示的视频序列展示了光照变化对跟踪算法的影响。对于 CarDark 序列,在夜晚道路上有明显的光照变化,原始 CT 算法在开始时就已经跟踪失败,而 Struck 算法开始时跟踪良好,但后期目标框产生了少许偏移,本文算法整体跟踪都十分成功;Man 序列中黑暗背景下 CT 算法与本文算法跟踪很好,但发生光照剧烈变化时,CT 算法跟踪失败,本文算法仍然可以精确跟踪;Trellis 序列中的光照变化也十分明显,可以看到后期光照变化时 Struck 算法与 CT 算法直接跟丢目标,而本文算法十分稳定。可以看出,Struck 算法与 CT 算法对光照都十分敏感。

尺度变化情况:图 1a、图 1b、图 1e、图 1f 所示序列都有目标的尺度变化,明显看出,本文算法在加入了尺度自适应算法后效果比未加入尺度估计的 Struck 算法和 CT 算法效果更好。

旋转与形变问题:图 1a、图 1b 所示序列都涉及平面内旋转问题,由 Dog1 序列中第 650 帧可以看出,本文算法应对此类问题比其他两类算法更好,Dog2 序列

中的形变问题在第 70, 96, 113 帧可以明显对比, Struck 算法与 CT 算法跟踪的目标位置已经出现了偏移, 效果不如本文算法。图 1e 中涉及平面外旋转, 可以看出, 本文算法在应对平面外旋转的情况时鲁棒性有所欠缺。

背景相似干扰与部分遮挡问题: 图 1c、图 1e 所示序列中含有背景相似干扰, CarDark 序列中 Struck 算法与本文算法都可以成功跟踪, 而在 Trellis 序列中只有本文算法可以成功跟踪。对于 Walking 中的部分遮挡问题, 本文算法与 Struck 算法都能跟踪目标, 且效果较 Struck 算法好一些。

综合以上分析, 本文算法不仅能够大大降低对光照的敏感度, 而且还能适应目标尺度的变化, 另外, 对于一些平面内旋转形变及部分遮挡问题也能进行跟踪, 算法鲁棒性有了一定的提高。

4.3 定量分析结果

分别针对 Struck 算法、CT 算法及本文算法的目标跟踪结果定量计算出中心距离精度、重叠率以及中心位置误差 3 项评价指标, 如表 2 所示, 并绘出跟踪过程中的定位误差的实时曲线, 如图 2 所示。

表 2 中心距离精度、重叠率及中心位置误差

Table 2 Distance precision, overlapping rate and center location error

视频素材	Struck 算法	CT 算法	本文算法	
中心距离精度/%	Dog1	62.37	88.22	95.85
	Dog2	70.87	78.74	98.43
	CarDark	100	0.508 9	100
	Man	23.13	30.60	100
	Trellis	50.62	14.94	80.67
重叠率/%	Walking	21.84	100	100
	Dog1	52.15	64.44	98.52
	Dog2	19.69	18.11	52.76
	CarDark	78.37	0.254 5	98.22
	Man	22.39	24.63	100
中心位置误差/像素	Trellis	51.85	14.76	72.58
	Walking	21.84	51.46	99.27
	Dog1	41.24	10.20	6.612
	Dog2	15.50	11.06	6.821
	CarDark	4.548	72.60	3.612
中心位置误差/像素	Man	38.21	25.20	2.872
	Trellis	44.77	61.24	12.40
	Walking	204.1	5.013	3.726

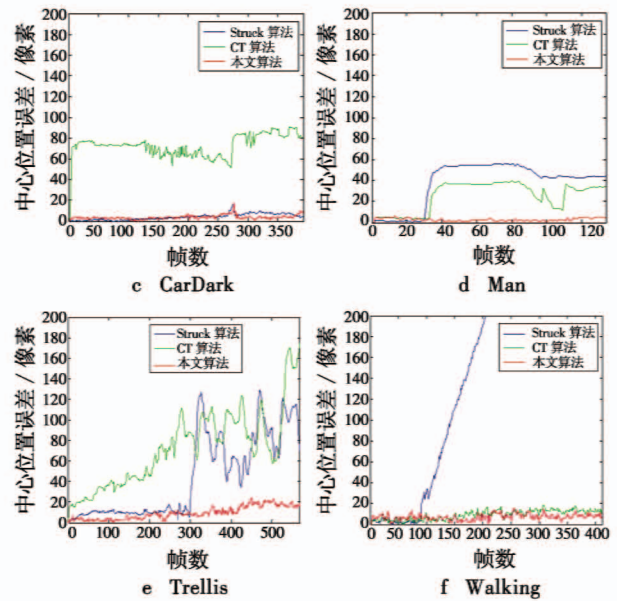
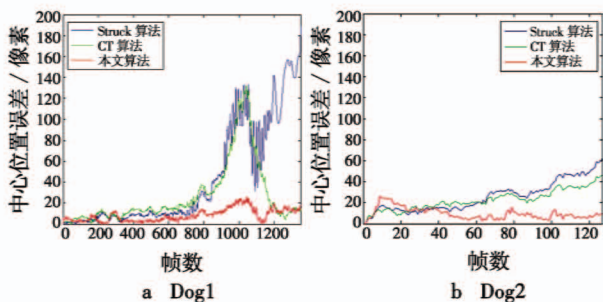


图 2 各视频序列定位误差对比

Fig. 2 The location errors of the video sequences

表 2 及图 2 都是对本文所进行对比的 3 种算法针对不同视频序列的定量分析结果。实验数据表明, 不论是从中心距离精度、中心位置误差还是重叠率方面来对比, 本文算法性能都更加优越。本文算法取得了最低的中心位置误差, 还取得了最高的中心距离精度和重叠率, 只有 Dog2 序列的跟踪重叠率较其他序列低一些。在实时性上, 以 Dog1 序列为例, 在上述计算机配置环境下, 图像像素大小为 240×320 , 初始跟踪框像素大小为 51×36 , 本文算法平均速度为 13.49 帧/s, 虽然相比原始压缩跟踪算法 64.4 帧/s 和 Struck 算法 20.2 帧/s 的速度有所降低, 但仍然满足准实时性要求。所以, 本文算法整体跟踪效果优于其他两种算法。

5 结论

本文以传统的压缩跟踪算法为基础, 提出了一种融合 HOG 类特征的尺度自适应压缩跟踪算法。由于融合了 HOG 类特征, 大大降低了算法对光照变化的敏感度, 而且加入了相关滤波尺度估计, 能够适应目标尺度的变化, 提高了跟踪算法的性能。实验表明, 本文算法在跟踪难点如光照变化、平面内旋转、形变、相似背景干扰及部分遮挡下跟踪性能表现良好, 具有较好的鲁棒性。算法虽然仍具有准实时性, 但在原始 CT 算法基础上降速很多, 所以在本文算法基础上继续提升速度是下一步的研究计划。另外, 本文对于两类特征的融合采用了固定比值的融合方式, 在保证算法实时性条件下两种特征的自适应融合也将是本文未来的研究方向。

参考文献

- [1] NUMMIARO K, KOLLER-MEIER E, VAN GOOL L. An

- adaptive color-based particle filter[J]. *Image and Vision Computing*, 2003, 21(1):99-110.
- [2] CHENG Y Z. Mean shift, mode seeking, and clustering [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995, 17(8):790-799.
- [3] LEI Y, DING X Q, WANG S J. Visual tracker using sequential Bayesian learning; discriminative, generative, and hybrid[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008, 38(6):1578-1591.
- [4] DONOHO D L. Compressed sensing [J]. *IEEE Transactions on Information Theory*, 2006, 52(4):1289-1306.
- [5] ZHANG K H, ZHANG L, YANG M H. Real-time compressive tracking [C]//*Proceedings of the 12th European Conference on Computer Vision*, 2012:864-877.
- [6] 朱秋平, 颜佳, 张虎, 等. 基于压缩感知的多特征实时跟踪[J]. *光学精密工程*, 2013, 21(2):437-444.
- [7] 张路平, 韩建涛, 李飏, 等. 尺度自适应特征压缩跟踪[J]. *国防科技大学学报*, 2013, 35(5):146-151.
- [8] 沈秋, 严小乐, 刘霖枫, 等. 基于自适应特征选择的多尺度相关滤波跟踪[J]. *光学学报*, 2017(5):166-175.
- [9] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection [C]//*IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005:886-893.
- [10] DANELLJAN M, HÄGER G, KHAN F, et al. Accurate scale estimation for robust visual tracking [C]//*Proceedings of the British Machine Vision Conference (BMVC)*, 2014:1-11.
- [11] HARE S, GOLODETZ S, SAFFARI A, et al. Struck; structured output tracking with kernels [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 38(10):2096-2109.

(上接第42页)

参考文献

- [1] 刘李旭, 戴鑫志, 侯振宇, 等. 基于光栅投影法的3D扫描技术研究[J]. *电光与控制*, 2017, 24(6):101-105.
- [2] 汤一平, 宗明理, 吴立娟. 基于双色反射模型的彩色结构光颜色识别的研究[J]. *计算机学报*, 2013, 36(9):1908-1916.
- [3] CHEN T B, GOESELE M, SEIDEL H P. Mesostructure from specularly [C]//*IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006:1825-1832.
- [4] KLINKER G J, SHAFER S A, KANADE T. A physical approach to color image understanding [J]. *International Journal of Computer Vision*, 1990, 4(1):7-38.
- [5] TAN R T, IKEUCHI K. Separating reflection components of textured surfaces using a single image [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, 27(2):178-193.
- [6] YANG Q X, WANG S N, AHUJA N. Real-time specular highlight removal using bilateral filtering [C]//*Computer Vision ECCV*, 2010:87-100.
- [7] YOON K J, CHOI Y, KWEON I S. Fast separation of reflection components using a specularly-invariant image representation [C]//*IEEE International Conference on Image Processing*, 2006:973-976.
- [8] SHEN H L, ZHENG Z H. Real-time highlight removal using intensity ratio [J]. *Applied Optics*, 2013, 52(19):4483-4493.
- [9] KIM H, JIN H, HADAP S, et al. Specular reflection separation using dark channel prior [C]//*IEEE Conference on Computer Vision and Pattern Recognition*, 2013:1460-1467.
- [10] SHAFER S A. Using color to separate reflection components [J]. *Color Research & Application*, 1985, 10(4):210-218.
- [11] ZHAO Y Q, PENG Q N, XUE J Z, et al. Specular reflection removal using local structural similarity and chromaticity consistency [C]//*IEEE International Conference on Image Processing*, 2015:3397-3401.