

自适应 HLBP 纹理特征的 Meanshift 目标跟踪算法

杜静雯¹ 黄山² 杨双祥¹

(四川大学电气信息学院 成都 610065)¹ (四川大学计算机学院 成都 610065)²

摘要 结合 Haar 型特性局部二元模式(HLBP)的图像纹理特征提取方法,提出一种新的目标跟踪算法,并将其运用到 Meanshift 框架中。将 Visual Studio 2010 和 opencv2.4.9 作为实验平台,将所提算法的实验结果与传统 Meanshift 跟踪算法、基于局部二元模式(LBP)纹理特征的 Meanshift 跟踪算法进行对比分析。实验结果表明,所提算法在背景复杂或背景简单的情况下都表现出了稳健而准确的跟踪特性,且在部分遮挡的情况下仍可以正确地跟踪目标。

关键词 局部二元模式, Haar 特征, Meanshift 跟踪算法, 部分遮挡

中图分类号 TP391 文献标识码 A

Meanshift Target Tracking Algorithm of Adaptive HLBP Texture Feature

DU Jing-wen¹ HUANG Shan² YANG Shuang-xiang¹

(School of Electrical Engineering and Information, Sichuan University, Chengdu 610065, China)¹

(College of Computer Science, Sichuan University, Chengdu 610065, China)²

Abstract In combination of the image texture feature extraction method, which is based on Haar local binary pattern (HLBP), a new target tracking algorithm was proposed, and applied to Meanshift tracking framework. Visual Studio 2010 and the opencv2.4.9 were the experimental platforms. We compared the results of the new algorithm with the results of other two kinds of algorithms, which are traditional Meanshift target tracking algorithm and the target tracking algorithm based on local binary pattern texture feature (LBP). Experimental results show that, in the case of simple or complicated background, the proposed tracking approach always shows steady and accurate tracking features, and in the event of partial occlusions, it can correctly track the target.

Keywords Local binary pattern, Haar feature, Meanshift tracking algorithm, Partial occlusions

1 引言

传统 Meanshift 理论很早就被运用于目标跟踪领域,由于它具有计算量小、容易实现、实时性好等优点,逐渐引起了人们的关注,对 Meanshift 目标跟踪算法的研究也逐渐成为热点。此算法最初是以颜色特征来表征目标,进而通过模板匹配进行跟踪,后来发现其应用范围有局限性,比如无法适用于背景与目标颜色相近的情况。为了寻求一种适用范围更加广泛的跟踪算法,人们提出了很多改进算法。文献[1]为了适应目标的尺度和旋转变化,把直方图分块理论以及目标旋转和缩放矩阵融入了 Meanshift 跟踪算法中,并取得了良好的效果。文献[2]在文献[1]的基础上进行了改进,提出了自适应分块方法,相应的权值根据巴士系数来确定,其不仅提高了目标形变的适应能力,同时具有一定的遮挡能力。文献[3]利用分块方法来抗遮挡,并且实现了核函数带宽的自适应,但耗时比较长。Meanshift^[4]跟踪算法的框架结合了三帧差分法与卡尔曼滤波,可以对目标进行实时准确的跟踪。针对目标特征描述的单一性,一系列改进算法被提出。文献[5]把颜色与边缘特征进行融合来共同描述目标,并且通过特征对目标和背景的区分能力来实现特征选择的自适应,提高了跟踪的鲁棒性。文献[6]融合了纹理特征,同时运用了分块的方法,

不仅使跟踪更为准确,而且使抗遮挡能力也有所提高;文献[7]提出了一种基于 Haar 特性的 LBP 纹理特征的提取方法,与 LBP 相比,它能更好地适应图像的放大与旋转,说明 HLBP 比 LBP 有更好的纹理表达能力。

在文献[7]的研究基础上,本文提出了一种基于 HLBP 纹理特征的 Meanshift 目标跟踪算法,实验结果表明,所提算法有较好的鲁棒性,而且跟踪准确性较高。

2 特征提取

2.1 LBP 纹理直方图

LBP 纹理特征^[8]在图像处理上反映了图像内部的结构信息,由于 LBP 对光照变化不敏感,很多学者对其进行了大量研究,并且把它运用到了 Meanshift 跟踪框架中,比如以纹理特征来代替颜色特征,将纹理和颜色特征相融合等,都是它在传统 Meanshift 跟踪算法中的具体应用,并且在跟踪性能方面取得了良好的效果,其中抗光照性能的增强尤为显著。

传统 LBP 算子的计算方法如图 1 所示,其模板大小为 3×3 ,若要求出图像上某一点的 LBP 值,就要以这一点为 LBP 算子的模板中心,从左上角的像素开始,顺时针方向依次与中心像素值进行比较,大于中心像素值则对应位置赋予 1,反之则赋予 0,直到 8 个方向均比较完毕,所得的数据按先

后顺序排列形成二进制数,然后转化为十进制数,即为该点的LBP值。

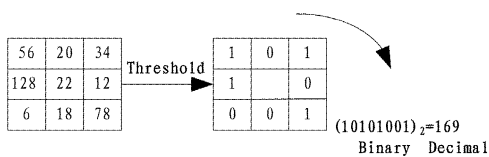


图1 LBP算子计算过程

通过以上方法提取的特征是用直方图来表征的,LBP纹理直方图如图2所示。

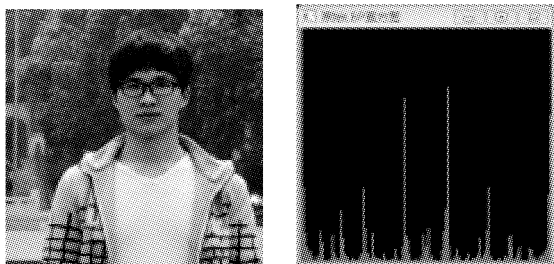


图2 示例图片及其LBP纹理直方图

2.2 HLBP直方图

2.2.1 Haar特征

Haar特征是由Viola等^[9]提出的,之后Lienhart R等对其模板进行了扩展。扩展后的特征模板如图3所示,由边缘特征、线特征环、中心环绕特征和对角线特征4部分组成。每个特征模板中白色部分像素和减去黑色部分像素和即为该模板特征值,该值是图像灰度变化的表征。

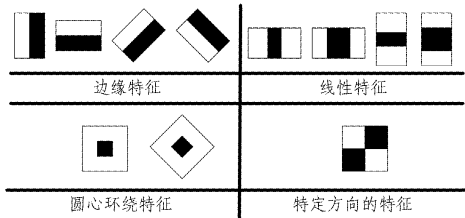


图3 HLBP特征模板

2.2.2 自适应阈值的HLBP特征

HLBP特征^[7]就是把Haar特征融入到LBP特征的计算中。Haar型特征计算简单且快速,能够很好地描述边缘信息,而且LBP特征恰好可以弥补Haar特征无法描述目标内部结构信息的局限性,因此HLBP特征充分结合了二者的优势,提高了目标特征描述的准确性。

HLBP特征是在8组Haar型特征编码模式下进行计算的,从图4可以看出,其共有8个方向,模板大小均为 5×5 ,若要求出待处理图像中某一点的HLBP值,就以该点为模板中心,并与其周围24个像素构成 5×5 模板。以图4所示进行计算,每个方向上深色部分对应的灰度值减去浅色部分相对应的灰度值,就是这个方向的灰度变化情况;8个方向计算完毕后,将所得值相加之和除以8(即均值),将此均值作为阈值,由此可以实现阈值自适应,而不必手动设置。从 135° 方向对应的计算值开始,依次与阈值做对比,按照LBP的计算方法,大于阈值的用二进制码1来表示,反之则用0来表示,最后得到一组二进制编码,再转化为十进制,即为模板中心点对应的实际位置的HLBP特征值。

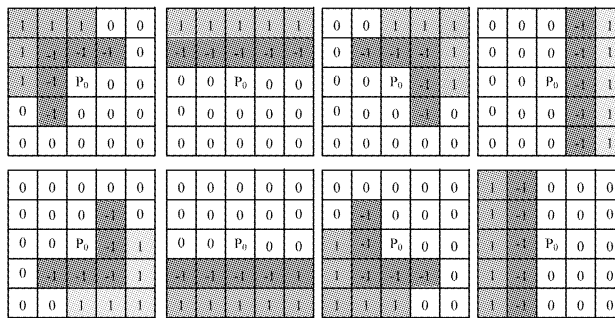


图4 HLBP的8种编码模式

所提取的特征通常通过直方图来表征,HLBP纹理直方图的示例如图5所示。相比于同一幅图片,HLBP纹理直方图有着更为明显的波峰与波谷,更加有利于模板匹配,鉴于此,本文将HLBP纹理特征运用到Meanshift目标跟踪算法中。

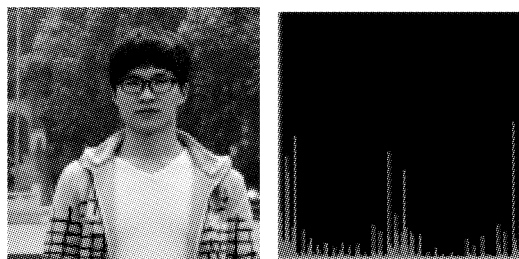


图5 示例图片及其HLBP纹理直方图

3 Meanshift跟踪算法

3.1 传统Meanshift跟踪算法的相关理论

传统Meanshift^[10]跟踪算法是通过特征模板匹配来进行跟踪的,初始目标需要在初始帧手动选择。

(1) 目标模型特征描述

$$\left\{ \begin{aligned} \sum_{u=1}^m q_u &= 1 \\ q_u &= C \sum_{i=1}^n k_E \left(\left\| \frac{x_0 - x_i}{h} \right\|^2 \right) \delta [b(x_i) - u] \\ C &= \frac{1}{\sum_{i=1}^n k_E \left(\left\| \frac{x_0 - x_i}{h} \right\|^2 \right)} \end{aligned} \right. \quad (1)$$

其中,初始帧中跟踪框内的位置信息用 $\{x_i\}$ 来表示, n 为框内总体像素数, x_0 为中心位置, u 为某个特征值($u=1, \dots, m$)。 $k_E(x)$ 为Epanechnikov函数, h 为带宽。 $b(x_i)$ 表示直方图的索引函数,其值为目标框中各个像素对应的RGB值。 δ 为Delta函数,用来判断 $b(x_i)$ 所对应的值是否属于特征值 u 。 C 是归一化系数。

(2) 候选模型特征描述

$$\left\{ \begin{aligned} \sum_{u=1}^m p_u &= 1 \\ p_u(y) &= C_h \sum_{i=1}^n k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \delta [b(x_i) - u] \\ C_h &= \frac{1}{\sum_{i=1}^n k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right)} \end{aligned} \right. \quad (2)$$

与目标模型描述方法相同,式(2)中 y 为候选模板的中心,当前帧中跟踪框里的位置信息用 $\{x_i\}_{i=1, \dots, n}$ 来表示, n 为候选模板中总体像素数, C_h 是归一化系数, $k(x)$ 为Epanechnikov函数。

(3) 相似性度量函数

采用巴氏相关系数^[7]来衡量两个模板之间的相似度,其定义如式(3)所示。

$$\rho(\hat{p}(y), \hat{q}) = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y) \hat{q}_u}} \quad (3)$$

其中, ρ 的值即为巴氏系数, 范围为 $0 \sim 1$, ρ 值越大则意味着两个模型越相似。

(4) 目标定位

$$\omega_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \delta[b(x_i) - u] \quad (4)$$

$$z = \sum_{i=1}^n \frac{x_i \omega_i g(\| \frac{y - x_i}{h} \|^2)}{\omega_i g(\| \frac{y - x_i}{h} \|^2)} \quad (5)$$

其中, z 为每次搜索得到的新的位置, ω_i 为权值, m 为特征值总数, $g(x) = -k'(x)$ 。在每帧的迭代过程中, 用 z 不断更新候选模板的中心位置 y , 直到巴氏系数取得最大值。

3.2 本文跟踪算法的流程

(1) 读取初始帧, 手动框出要跟踪的目标, 把彩色图转化为灰度图, 并计算其 HLBP 特征值, 根据式(1)计算得到初始目标特征模型 \hat{q} ;

(2) 读取下一帧, 同样在处理之前转化为灰度图, 以前一帧中跟踪框的中心位置来初始化当前帧跟踪框的中心 y_0 ;

(3) 根据式(2)计算 $\{\hat{p}_u(y_0)\}_{u=1,2,\dots,m}$;

(4) 根据式(3)计算巴氏相关系数 $\rho(\hat{p}(y_0), \hat{q})$, 并根据式(4)计算此时的权值 ω_i ;

(5) 根据式(5)计算出 Meanshift 算法搜索的新位置 z , 同时更新 $\hat{p}(z)$, 并计算新的巴氏相关系数 $\rho(\hat{p}(z), \hat{q})$;

(6) 如果 $\rho(\hat{p}(z), \hat{q}) < \rho(\hat{p}(y_0), \hat{q})$, 则使 $z = (y_0 + z)/2$ 。

(7) 如果 $\|z - y_0\| < \epsilon$, 则停止循环; 反之, 使 $y_0 = z$ 且返回步骤(3)继续循环。

(8) 当循环体最终停止迭代时, 此时的 z 即为目标在本帧中的位置, 然后返回步骤(2)。

4 实验结果与分析

4.1 实验一

该实验是为了验证算法在简单背景下的性能, 需要的测试视频的帧宽度为 352, 帧高度为 288, 帧速率为每秒 25 帧, 总帧数为 4750 帧, 跟踪目标为视频中草坪角正对且向 135° 行走的行人。实验运行环境为 Visual Studio 2010 和 opencv 2.4.9 版本, 使用的编程语言为 C/C++。在程序中添加了保存实验结果图的模块, 可以每 10 帧保存一次结果图。

图 6 为传统 Meanshift 目标跟踪结果图(为了便于描述, 以下统称为算法 1), 图 7 为基于 LBP 纹理特征的 Meanshift 目标跟踪结果图(以下统称为算法 2), 图 8 为本文所提算法的跟踪结果图, 图底所标数字为每幅图像所对应的帧数, 3 个算法都是从第 1 帧开始手动选择目标的。我们选择第 60 帧、第 120 帧、第 180 帧、第 240 帧、第 300 帧、第 360 帧、第 420 帧、第 480 帧的实验图来进行分析, 从图 6—图 8 中可以清楚地看出, 算法 1 在第 60, 120 帧可以稳定跟踪, 但之后出现跟踪不稳定的情况, 甚至在 360 帧之后就已跟丢目标。而在第

60, 120, 180 帧时算法 2 和本文所提算法均可以顺利跟踪, 从第 240 帧开始, 算法 2 开始出现跟不上甚至跟丢的情况, 而本文算法仍然可以稳健地跟踪目标, 从跟踪框与目标的位置可以看出, 目标基本处于跟踪框中心, 偏差较小。通过以上实验可以看出, 本文算法可以更好地跟踪目标, 而且稳健性更强。

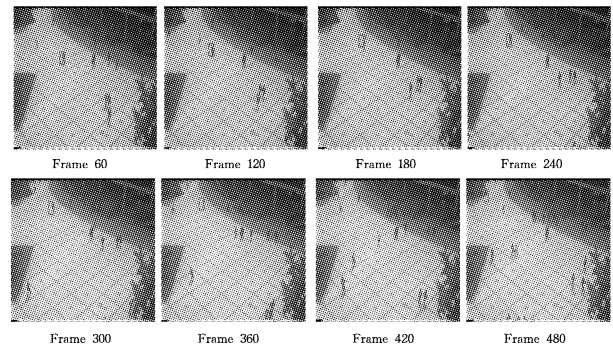


图 6 传统 Meanshift 目标跟踪结果图

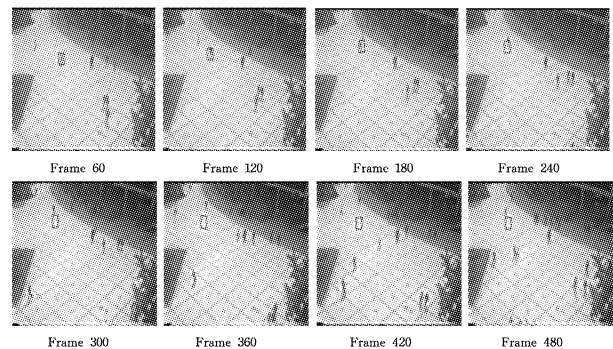


图 7 基于 LBP 纹理特征的 Meanshift 目标跟踪结果图

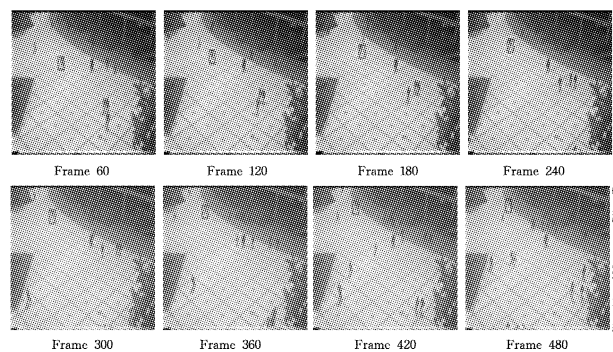


图 8 基于 HLBP 纹理特征的 Meanshift 目标跟踪结果图

为了进一步证明本文算法的跟踪性能, 对实验结果进行定量分析。记录两个算法在不同帧中跟踪框的中心位置与目标的实际位置, 使用欧氏距离^[11]来计算两个位置之间的距离, 欧氏距离越小, 说明跟踪误差越小, 以此来衡量算法跟踪性能的好坏。图 9 给出了 3 种算法的欧氏距离的曲线图。

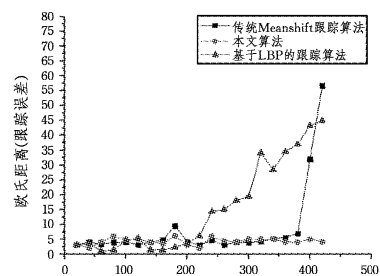


图 9 3 个算法的欧氏距离的曲线图

从图中可以看出,本文算法的欧氏距离比较小且比较稳定,说明跟踪误差较小,跟踪更加准确。

4.2 实验二

该实验是为了验证算法在背景复杂情况下的性能,需要的测试视频的帧宽度为 720,帧高度为 576,帧速率为每秒 5 帧,总帧数为 1200 帧,跟踪目标为视频中火车旁边穿黑色风衣并且向右行走的女人。实验运行环境和编程语言均与实验一相同,由于视频帧数较少,保存方式改为逐帧保存。

图 10 为传统 Meanshift 目标跟踪结果图,图 11 为基于 LBP 纹理特征的 Meanshift 目标跟踪结果图,图 12 为本文所提算法的跟踪结果图,图底所标数字为每幅图像所对应的帧数,两个实验都是从第 110 帧开始手动选择目标的。

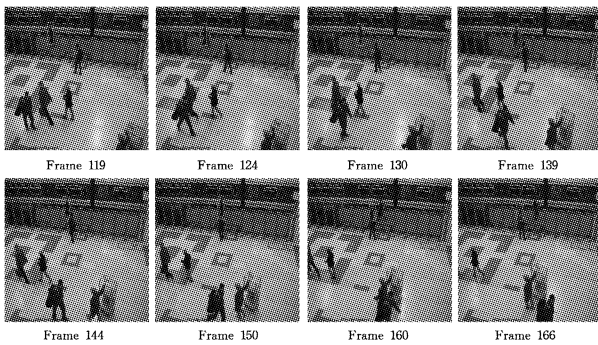


图 10 传统 Meanshift 目标跟踪结果图

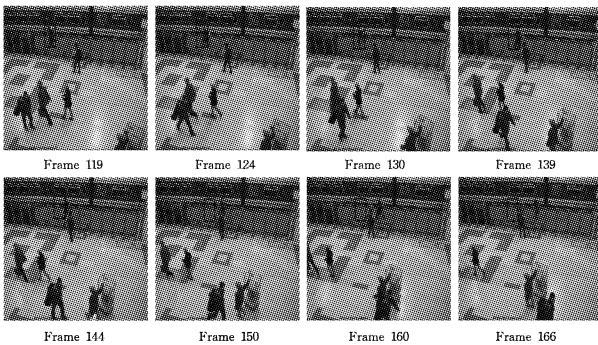


图 11 基于 LBP 纹理特征的 MeanShift 目标跟踪结果图

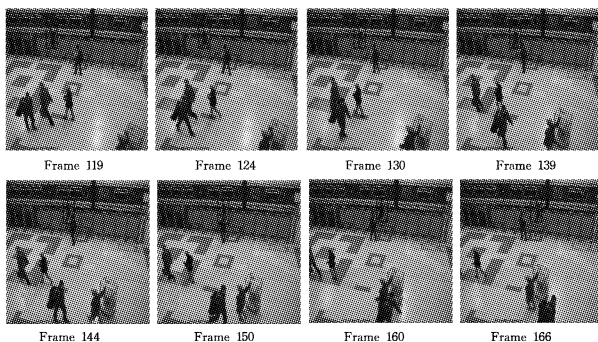


图 12 基于 HLBP 纹理特征的 MeanShift 目标跟踪结果图

我们选取第 119 帧、第 124 帧、第 130 帧、第 139 帧、第 144 帧、第 150 帧、第 160 帧、第 166 帧的实验图来进行分析,从图 10—图 12 中可以清楚地看出,在第 119,124,130,139 帧时 3 种算法均可以顺利跟踪,从第 144 帧开始,算法 2 出现跟不上甚至跟丢的情况,而本文算法和算法 1 仍然可以稳健地跟踪目标,可以注意到,在第 139 帧时,一名背书包的男子靠近目标周围,并且在第 150 帧时遮挡了目标的下半身,算法 1 在遮挡后跟错了目标,但是本文算法依然可以准确跟踪目标,不被干扰;从跟踪框与目标的位置来看,本文算法的实验结果

图中,目标基本处于跟踪框中心,偏差不是太大。由此可以说,在背景复杂的情况下,本文算法可以更好地跟踪目标,而且稳健性更强,具有一定的抗遮挡能力。

与实验一相同,对实验二的结果进行定量分析。记录两个算法在不同帧中跟踪框的中心位置与目标的实际位置,使用欧氏距离来计算两个位置之间的距离。图 13 给出了 3 种算法的欧氏距离的曲线图,从图中可以看出,本文算法的跟踪误差整体较小,而且上下浮动范围不是很大,相对于基于 LBP 算法和传统 MeanShift 目标跟踪算法的欧氏距离,本文算法的欧氏距离比较稳定,可以更直观地看出本文算法的跟踪更为准确且更为稳健。

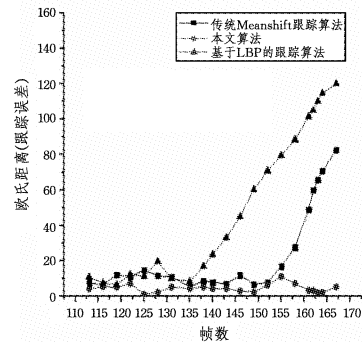


图 13 3 个算法的欧氏距离的曲线图

结束语 文中提出了一种基于 HLBP 纹理特征的 MeanShift 跟踪算法,在 LBP 算子中融入了 Haar 型特征,不仅包含内部的纹理信息,还包含了外部的结构信息,结合了二者的优势,使所跟踪目标的特征描述更为细致。实验结果表明,本文算法在背景简单和背景复杂的情况下,仍可以实现准确而稳健的跟踪,而且在出现部分遮挡的情况下,也可以稳定跟踪。今后可以在尺度方面做一些改进,使其可以随目标自适应地变化,进一步完善该算法。

参考文献

- [1] 胡钢,杨静宇. 基于分块颜色直方图的 MeanShift 跟踪算法[J]. 系统仿真学报,2009,30(8):66-70.
- [2] 杜凯,巨永锋,靳引利,等. 自适应分块颜色直方图的 MeanShift 跟踪算法[J]. 武汉理工大学学报,2012,34(6):141-143.
- [3] 徐海明,黄山,李云彤. 基于改进的 MeanShift 鲁棒跟踪算法[J]. 计算机工程与科学,2015,37(6):1161-1167.
- [4] 张英,车进,牟晓凯,等. 改进的 MeanShift 运动目标跟踪算法[J]. 电视技术,2016,40(10):97-100.
- [5] 史宝明,贺元香,邢玉娟. 融合边缘特征的 MeanShift 跟踪算法[J]. 兰州文理学院学报,2015,29(6):49-52.
- [6] 李冠彬,吴贺丰. 基于颜色纹理直方图的带权分块均值漂移目标跟踪算法[J]. 计算机辅助设计和图形学学报,2011,22(8):1232-1235.
- [7] 周书仁,殷建平. 基于 Haar 特性的 LBP 纹理特征[J]. 软件学报,2013,24(8):1909-1926.
- [8] 袁瑜健. 基于纹理特征的目标识别与跟踪技术研究[D]. 北京:北京理工大学,2016.
- [9] OJALA T, PIETIKAINEN M, HARWOOD D. A comparative study of texture measures with classification based on featured distributions[J]. Pattern Recognition,1996,29(1):51-59.
- [10] 李洋. 基于 MeanShift 的运动目标跟踪算法研究[D]. 沈阳:沈阳理工大学,2015.
- [11] 姜明新. 智能视频监控中的目标跟踪技术研究[D]. 大连:大连理工大学,2013.