基于 ROS 的移动机器人核相关目标跟踪方法的优化设计

陈丹, 姚伯羽, 吴欣

(西安理工大学自动化与信息工程学院 西安 710048)
(chdh@xaut.edu.cn)

摘要: 针对小型移动机器人跟踪人体运动目标时, 因光照变化或人体快速运动导致跟踪准确率降低甚至跟踪失败的问题, 提出人体足部运动模型引导下的核相关滤波跟踪算法(kernel correlation filter guided by motion model, MMKCF). 该算法通过建立足部运动模型预估视频跟踪中双脚位置, 获得核相关滤波跟踪算法的目标检测区域, 有效解决了跟踪框的漂移. 基于外场实际拍摄的 4 组人体行走视频进行了目标跟踪实验, 对比了 MMKCF, KCF 等 5 种算法的跟踪性能. 实验结果表明, 在光照变化以及目标快速运动时, MMKCF 算法的平均跟踪准确率约为 0.77, 远高于其他 4 种跟踪算法. 最后在机器人操作系统(robot operating system, ROS)下将 MMKCF 算法应用于 TurtleBot 机器人目标跟踪, 成功地完成了人体快速移动时的足部跟踪, 验证了所提算法具有较强的鲁棒性和实时性.

关键词: 移动机器人; 目标跟踪; 运动模型; 核相关滤波; 机器人操作系统

中图分类号: TP391.41 DOI: 10.3724/SP.J.1089.2020.18253

Optimal Design of Kernel Correlation Filtering Target Tracking Method for Mobile Robot Based on ROS

Chen Dan, Yao Boyu, and Wu Xin

(School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048)

Abstract: Aiming at the problem that the tracking precision of small mobile robot decreases or even fails due to the change of illumination or the fast motion of human body, a kernel correlation filter guided by motion model (MMKCF) is proposed. By building a feet motion model to predict the position of the feet in video tracking, the algorithm obtains the target detection area of the kernel correlation filter tracking algorithm, which effectively solves the drift of the tracking box. Based on four groups of human walking video, target tracking experiments are carried out, and tracking performance of five algorithms such as MMKCF and KCF are compared. The experimental results show that the average tracking precision of MMKCF algorithm is about 0.77 when the illumination changes and the target moves rapidly, which is much higher than the other four tracking algorithms. Finally, the MMKCF algorithm is applied to the target tracking of TurtleBot robot under the robot operating system (ROS), and the foot tracking of human body during fast motion is successfully completed, which proves that the proposed algorithm has strong robustness and real-time performance.

Key words: mobile robot; target tracking; motion model; kernel correlation filter; robot operating system (ROS)

在人工智能和机器人技术的推动下, 智能服务机器人越来越多地被应用在生产和生活中, 人机交互已成为机器人学的新兴领域之一^[1]. 智能机

器人能够通过自身携带的摄像头获取视频信息, 识别周围环境, 最后发现并找到目标. 相关滤波跟踪算法因为具有较高的准确率和优异的跟踪速度

收稿日期: 2020-02-28; 修回日期: 2020-05-13. 基金项目: 国家自然科学基金(61671375); 榆林市科技计划(2019-146); 西安理工大学研究生竞赛培育项目(252051834). 陈丹(1975—), 女, 博士, 副教授, 硕士生导师, 主要研究方向为现代信号处理; 姚伯羽(1994—), 男, 硕士研究生, 主要研究方向为计算机视觉; 吴欣(1995—), 女, 硕士研究生, 主要研究方向为机器人即时定位与地图构建.

成为计算机视觉领域的研究热点之一. Henriques 等^[2]于 2015 年提出核相关滤波(kernel correlation filter, KCF)跟踪算法, 对 CSK(circulant structure kernel)算法进行了改进, 采用方向梯度直方图(histogram of oriented gradients, HOG)特征代替了灰度特征, 针对 OTB(object tracking benchmark)测试集跟踪准确率为 0.74^[3]. 2016 年, Bertinetto 等^[4]提出 Staple 算法, 将 HOG 特征和颜色直方图 2 种特征加权融合, 对 OTB 测试集的跟踪准确率为 0.79, 但图像处理速度降低. 2017 年, Wang 等^[5]提出 DCFNet 算法, 是基于循环神经网络的相关滤波相算法, 使用 OTB 测试集时准确率达到 0.89. 2018 年, Bhat 等^[6]提出 UPDT(unveiling the power of deep tracking)算法, 采用自适应融合浅层特征和深层特征进行检测估计, 在 OTB 测试集上的跟踪准确率高于 0.91. 这些研究成果基本都是针对图像特征提取进行的算法改进.

近年来, 将目标运动模型与跟踪算法相结合已成为视觉跟踪研究热点之一. 2015 年, Wu 等^[7]认为运动模型能使目标跟踪算法的性能得到提升. 2016 年, Wang 等^[8]针对视觉跟踪中速度和方向的突变运动, 提出了一种基于多运动模型的自校正算法, 该算法面对光照变化等问题时跟踪性能欠佳, 且跟踪速度较慢. 2017 年, Duan 等^[9]提出了一种具有推荐系统的逐帧检测跟踪方法, 该方法使用特征字典描述目标的外观模型, 但该方法计算复杂度高, 难以满足实时跟踪要求. 2018 年, Chen 等^[10]提出了一种手部跟踪方法, 建立了手部的运动模型并引入保角几何代数作为数学工具, 手部运动的自由度被赋予几何方式形成手掌和手指的不同平面, 最后采用粒子滤波算法对平面上每根手指被标注的彩色块进行跟踪.

本文是基于小型机器人进行的人体目标视频

跟踪算法研究. 小型机器人如果要完成人体上半身跟踪, 摄像头有 2 种安装方法, 一种是给机器人安装一个高位置的摄像头, 这种方法对于矮小型机器人不可行; 另一种是调高机器人摄像头相对于水平线的仰角, 但在这种情况下摄像头视野范围十分有限^[11]. 因此, 对于矮小型机器人, 本文将人体足部作为跟踪目标更为合适, 因为摄像头可以安装在一个较低的位置, 在视野更大的范围内通过足部跟踪人体目标. 本文首先建立了人体足部运动模型, 在运动模型引导下对核相关滤波跟踪算法进行了改进, 针对不同外场场景人体行走视频对足部跟踪进行了仿真实验, 根据人体位置采用自适应改变机器人线速度和角速度的方法. 最后在 ROS 系统下成功地应用于小型 TurtleBot 机器人, 实现了人体足部快速运动时的目标跟踪.

1 人体足部运动模型建立

在早期使用视觉传感器对人体腿部运动的研究中, 双脚中心点的垂直位移变化接近于低振幅的余弦波^[12]. 本文将人体行走时的足部运动分为 2 个部分: 局部运动和全局运动. 双脚之间的相对运动为局部运动, 双脚整体运动为全局运动.

1.1 局部运动模型

本文拍摄 3 组行人以不同速度行走的视频. 选取双脚脚后跟各自最低点来计算双脚之间的垂直位移, 视频中连续 100 帧图像进行计算后得到图 1 所示曲线.

由于人在行走过程中双脚之间的实际垂直位移的变化类似于余弦函数, 因此采用余弦函数建立足部局部运动模型.

在某一帧图像中, 取左脚位置坐标为 (x_1, y_1) ,

图 1 行走过程中双脚后跟之间的垂直距离

右脚位置为 (x_r, y_r) , 则双脚在垂直方向上的相对位移为 $d^t = y_r^t - y_l^t$. 其中, t 表示帧数. 采用幅度和相位调制余弦函数来对图中双脚之间的垂直位移进行建模, 其计算公式^[13]为

$$d^t = A^t \cos\left(\frac{2\pi}{T^t}t + \varphi^t\right) \quad (1)$$

其中, A^t 为余弦函数振幅; T^t 为周期; φ^t 为相位. A^t 与双脚的相对距离和高度有关, 在跟踪过程中该振幅是变化的. 通过检测的 d^t 最大值和最小值来不断更新 A^t . 余弦函数的周期是随步行速度而变化的, 周期可由数据 d^t 的 2 个连续过零点时间获得. 过零点的时间可表示为

$$t_{\text{zero}}^k = t - 1 + \frac{|d^{t-1}|}{|d^t| + |d^{t-1}|}, \quad d^t \cdot d^{t-1} \leq 0 \quad (2)$$

其中, t_{zero}^k 为第 k 个过零点, 周期为 $T^t = 2 \times (t_{\text{zero}}^k - t_{\text{zero}}^{k-1})$.

T^t 仅在有过零点时更新. 同样在每个过零点需要更新相位

$$\varphi^t = \begin{cases} \frac{\pi}{2} - \frac{2\pi}{T^t}t, & d^{t-1} > 0 \text{ 且 } d^t < 0 \\ \frac{3\pi}{2} - \frac{2\pi}{T^t}t, & d^{t-1} \leq 0 \text{ 且 } d^t \geq 0 \end{cases} \quad (3)$$

一般假设人行走双脚之间水平位移变化很小, 当前帧中的水平位移可以估计为 $l_{\text{pred}}^t = l^{t-1}$.

1.2 全局运动模型

本文采用双脚之间中点位置的变化进行全局运动模型的建立. 一般情况下, 人体行走在连续帧上的变化不快, 所以假设连续 2 帧之间行走速度不变, 用 (x_m^t, y_m^t) 表示第 t 帧双脚之间的中点, 则第 t 帧中点的预测位置可由前面 2 帧中点位置获得:

$$\begin{pmatrix} x_{\text{mpred}}^t \\ y_{\text{mpred}}^t \end{pmatrix} = 2 \cdot \begin{pmatrix} x_m^{t-1} \\ y_m^{t-1} \end{pmatrix} - \begin{pmatrix} x_m^{t-2} \\ y_m^{t-2} \end{pmatrix}, \quad t \geq 2 \quad (4)$$

其中, 从目标人后方观测的左脚和右脚中点预测位置分别表示为

$$\begin{cases} \begin{pmatrix} x_{\text{lpred}}^t & y_{\text{lpred}}^t \end{pmatrix}^T = \begin{pmatrix} x_{\text{mpred}}^t - \frac{l^t}{2} & y_{\text{mpred}}^t + \frac{d^t}{2} \end{pmatrix}^T \\ \begin{pmatrix} x_{\text{rpred}}^t & y_{\text{rpred}}^t \end{pmatrix}^T = \begin{pmatrix} x_{\text{mpred}}^t + \frac{l^t}{2} & y_{\text{mpred}}^t - \frac{d^t}{2} \end{pmatrix}^T \end{cases} \quad (5)$$

2 KCF 算法

KCF 算法是在当前帧中对跟踪目标区域进行

循环移位, 以此构造大量的正负样本来训练分类器, 再运用分类器计算下一帧检测区域中不同子区域与跟踪目标的相似程度, 最后选取相似度最大的位置作为新的跟踪目标. 该算法利用了离散傅里叶变换降低分类器训练和检测过程中的运算复杂度. 算法分为分类器训练、模板匹配和模板更新 3 个步骤.

2.1 分类器训练

在 KCF 算法中, 分类器是通过目标周围大小为 $m \times n$ 的图像块 z 训练获得. 假设图像块 z 的目标函数为 $f(z) = w^T z$, 则分类器是通过最小正则化风险泛函

$$\min_w \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2 \quad (6)$$

进行训练. 其中, $i \in \{0, \dots, m-1\} \times \{0, \dots, n-1\}$; x_i 为利用循环移位获得的训练样本; y_i 为回归标签; λ 为正则化参数; w 为权重系数. 式(6)的解可以表示为

$$w = (X^T X + \lambda I)^{-1} X^T y \quad (7)$$

其中, 矩阵 X 由元素 x_i 组成; 向量 y 由 y_i 组成. 非线性分类器 $f(z)$ 将低维度输入 x_i 映射到高维空间 $\phi(x_i)$ 可以提高数据计算性能. 映射后式(7)的解可表示为输入的线性组合 $w = \sum_i \alpha_i \phi(x_i)$, $\phi(x_i)$ 为将训练样本 x_i 映射到高维特征空间的函数, 且

$$f(z) = w^T z = \sum_i \alpha_i \kappa(z, x_i) \quad (8)$$

其中, κ 表示核函数 $\kappa(x, x') = \phi^T(x) \phi(x')$.

由文献[14]可知, $\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda}$. 其中, $\hat{\alpha}$ 是 α 的离散傅里叶变换; $\hat{y}$ 为 y 的离散傅里叶变换; $\hat{k}^{xx}$ 表示核矩阵 K 第 1 行的离散傅里叶变换.

2.2 模板匹配

定义 K^z 表示测试样本和训练样本在核空间的核矩阵, 则测试样本的响应为

$$\hat{f}(z) = \hat{K}^{xz} \odot \hat{\alpha} \quad (9)$$

其中, $\hat{K}^{xz}$ 为核矩阵 K^z 的第 1 行; $\hat{\alpha}$ 为分类器系数; $\odot$ 为对应元素点乘. 模板匹配是为求取函数 $\hat{f}(z)$ 最大值对应的位置.

2.3 模板更新

将线性插值法引入到模板的更新中, 分类器

系数 $\hat{\alpha}$ 和目标观测模型 $\hat{x}$ 分别为

$$\hat{\alpha}^{t+1} = m\hat{\alpha}^{t-1} + (1-m)\hat{\alpha}^t \quad (10)$$

$$\hat{x}^{t+1} = m\hat{x}^{t-1} + (1-m)\hat{x}^t \quad (11)$$

其中, m 为学习率.

2.4 足部运动模型引导下的 KCF 算法

本文将人体足部运动模型引导下的 KCF 算法记为 MMKCF(kernel correlation filter guided by motion model)算法.

改进算法与原始算法的区别是原始 KCF 算法每一帧的检测区域位置由上一帧的目标位置决定, 而 MMKCF 算法的检测区域由运动模型估计而获得. 例如, 在对一组视频序列中足部进行跟踪时, 采用 KCF 算法在第 86 帧时跟踪目标框发生了漂移, 如图 2a 所示. 以该跟踪框中心扩大 250% 所得区域作为第 87 帧的目标检测区域(虚线框), 如图 2b 所示. 而 MMKCF 是由足部运动模型在对每一帧进行跟踪时获得目标检测区域, 如图 2b 所示实线框, 而不是通过上一帧目标位置获得. MMKCF 算法引入运动模型对检测区域位置进行预测, 从而使目标始终处于检测区域内. MMKCF 算法流程图如图 3 所示.

a. 第86帧目标框

b. 第87帧 KCF 和 MMKCF 检测区域

图 2 KCF 与 MMKCF 算法检测区域示意图

图 3 MMKCF 算法流程图

3 外场环境下实测视频跟踪实验

本文在西安理工大学校园拍摄了 4 组不同的行人步行测试视频. 采用 KCF 算法, MMKCF 算法, BACF(background-aware correlation filters)算法^[15], SAMF(scale adaptive kernel correlation filters with mutiple features)算法^[16], HCF(hierarchical convolutional features for visual tracking)算法^[17]5 种算法对 4 组视频进行足部跟踪实验, 在跟踪 100 帧情况下对比了跟踪准确率和跟踪误差, 并对 KCF 和 MMKCF 算法运行时间进行对比. 4 组视频分别命名为 Passageway, Shade, Garden 和 Zoom. 5 种算法中 BACF 算法采用一个掩码矩阵增大相关滤波器的搜索样本数量, 通过密集采样的方法得到真正的负样本. SAMF 算法将方向梯度直方图特征与颜色特征结合并加入了尺度滤波器. HCF 算法使用深度卷积特征代替方向梯度直方图特征.

视频描述如表 1 所示. 在 4 组视频中, 采用鞋上一点的灰度值对光照变化进行量化, 每迈一步

表 1 实际拍摄 4 种行人步行视频描述

视频名称	行人行走情况和背景情况
Passageway	室内阴天慢速度行走, 无明显光照变化
Shade	室外中等速度行走, 从树荫到阳光下, 有明显光照变化
Garden	室外中等步速行走, 有光照变化, 地板背景较复杂
Zoom	室外阴天快速行走, 无明显光照变化

所需帧数对步行速度进行量化. 视频 Passageway 中平均 26 帧/步, 灰度值的变化范围为 89~117; 视频 Shade 中平均 22 帧/步, 灰度值的变化范围为 54~154; 视频 Garden 中平均 30 帧/步, 灰度值的变化范围为 136~191; 视频 Zoom 中平均 14 帧/步, 灰度值的变化范围为 40~75.

3.1 跟踪准确率

准确率是指在视频序列中, 跟踪算法所估计的目标位置中心点与标准目标位置中心点的差值小于某一阈值所占视频帧总数的比例^[7]. 针对 4 组视频, 分别采用 KCF, MMKCF, BACF, SAMF 和 HCF 算法进行人体足部跟踪实验. 跟踪准确率曲线如图 4 所示, 其中横坐标为跟踪位置中心与实际标注的目标中心的误差阈值. 当位置误差阈值为 20 个像素时, 对于视频 Passageway, KCF 算法跟踪准确率为 0.86, MMKCF 算法跟踪准确率为 0.88, BACF 算法跟踪准确率为 0.92, SAMF 算法跟踪准确率为 0.93, HCF 算法跟踪准确率为 0.95. 对于视频 Shade, KCF 算法跟踪准确率为 0.39, MMKCF 算法跟踪准确率为 0.75, BACF 算法跟踪准确率为 0.71, SAMF 算法跟踪准确率为 0.58, HCF 算法跟踪准确率为 0.72. 对于视频 Garden, 由于地板上具有条纹导致背景较复杂,

KCF, BACF, SAMF 和 HCF 跟踪准确率分别为 0.14, 0.43, 0.29 和 0.58; 而 MMKCF 算法成功地跟踪目标, 准确率达到 0.72, 这主要是因为其采用了运动模型进行足部位置预估, 而减小了目标搜索框漂移的可能性. 对于视频 Zoom, 因人体步速较快, 双脚运动的位置超出了 KCF, BACF 和 SAMF 3 种算法的搜索区域而丢失目标, 跟踪准确率分别为 0.36, 0.29 和 0.44; HCF 算法和 MMKCF 算法成功地跟踪目标, 准确率分别为 0.69 和 0.71. 由此可知, 与其他 4 种算法相比, 在光照变化、较复杂背景和目标快速运动的情况下, 本文 MMKCF 算法跟踪准确率较高. 在上述 4 个视频中, KCF, BACF, SAMF 和 HCF 算法的平均准确率分别为 0.44, 0.59, 0.56 和 0.74, 而 MMKCF 算法平均准确率为 0.77, 高于其他 4 种算法. KCF, MMKCF, BACF 和 SAMF 4 种算法的平均跟踪速度均大于 30 FPS, 但是使用深度卷积特征的 HCF 算法的平均跟踪速度不足 1 FPS, 难以满足跟踪实时性的要求.

3.2 跟踪误差

跟踪误差定义为每帧跟踪框中心与真值框中心坐标之间的欧几里得度量. 图 5 所示为 4 组视频的 5 种算法跟踪误差曲线图. 因视频 Passageway

图 4 不同算法在 4 种视频下的跟踪准确率

图 5 不同算法在 4 种视频下的跟踪误差

中人体行走较慢, 5 种算法的跟踪误差均较小. 在视频 Shade 中, KCF 算法在第 15 帧出现跟踪框漂移导致在后续帧中丢失了目标; BACF 算法和 SAMF 算法在全部 100 帧中跟踪框都出现了较大的漂移, 但都始终成功地跟踪目标; HCF 算法也成功地跟踪目标, 在 5 种算法中 MMKCF 算法的跟踪误差最小. 在视频 Garden 中, KCF 算法在第 25 帧之后直接丢失了目标, BACF 算法在第 58 帧后丢失了目标, SAMF 算法和 HCF 算法在前 50 帧时跟踪误差较大, MMKCF 在误差波动较小的情况下始终成功地跟踪目标. 在视频 Zoom 中, 目标移动速度较快, KCF 算法在第 33 帧丢失了目标, BACF 算法在第 36 帧丢失目标, SAMF 算法在第 45 帧丢失目标, MMKCF 算法和 HCF 算法始终成功地跟踪目标, 但 MMKCF 算法跟踪误差波动小于 HCF 算法.

3.3 KCF 与 MMKCF 算法运行时间

检测区域大小对跟踪算法实现的实时性存在一定影响, 为探究检测区域大小对算法成功跟踪时间的影响. 本文在成功跟踪的情况下, 对 2 种算法需要的检测区域大小以及算法运行时间进行对比. 结果如表 2 所示, 其中检测区域扩展系数表示后续帧的检测区域相对于第 1 帧检测区域扩大所

需要相乘的系数. 在成功地跟踪 17 帧时, KCF 算法要将目标区域乘以扩展系数为 1.6 后作为搜索区域才可以成功跟踪, 而 MMKCF 算法只需乘以扩展系数为 1.5 的搜索区域, 此时对比 2 种算法跟踪 17 帧的耗时, KCF 算法耗时比 MMKCF 算法长. 在成功地跟踪第 18、第 58、第 59 和第 100 帧时, KCF 算法所需的搜索区域面积增大且均大于 MMKCF 算法, 因此 KCF 算法运行时间比 MMKCF 算法长, 而成功跟踪 100 帧时 MMKCF 算法运行时间比 KCF 大约减少 2 s. 由此可知, 在成功跟踪相同帧数时, KCF 需要扩大目标检测区域实现成功跟踪, MMKCF 算法因为引入运动模型进行目标位置预

表 2 2 种算法运行时间

成功跟踪 帧数	KCF ^[2]		MMKCF	
	扩展系数	时间/s	扩展系数	时间/s
12	1.5	0.210 6	1.5	0.212 6
17	1.6	0.324 0	1.5	0.289 1
18	1.9	0.401 5	1.5	0.311 1
58	2.1	1.596 5	1.5	0.949 3
59	2.4	2.825 0	1.5	0.980 5
100	2.7	3.809 3	1.5	1.610 8

估计, 不需要扩大检测区域, 提高了运动目标跟踪的实时性。

3.4 跟踪实验结果

针对有光照和快速运动跟踪问题, MMKCF 算法较 KCF 算法具有更高的鲁棒性。本文选取中等步速且存在光照变化的视频 Shade 和具有快速行走的视频 Zoom 分别进行 2 种算法跟踪实验结果对比分析, 跟踪对比结果以视频中第 10, 第 40, 第 70 和第 100 帧为例, 如图 6 所示。

a. KCF 算法^[2]跟踪视频 Shade

b. MMKCF 算法跟踪视频 Shade

c. KCF 算法^[2]跟踪视频 Zoom

d. MMKCF 算法跟踪视频 Zoom

图 6 不同算法对 2 种视频跟踪结果

在对视频 Shade 进行跟踪时, 在走出阴影时, KCF 算法由于光照变化以及抬脚遮挡对鞋子的 HOG 特征造成了一定的影响, 在第 40 帧时, 导致右脚的跟踪框发生了一定程度的漂移, 跟踪目标转变为右脚的左半部分。MMKCF 算法在第 40 帧时, 跟踪框有较小的漂移, 但是始终成功地跟踪双脚。对于视频 Zoom, KCF 算法在第 32~33 帧时, 左脚抬起迈步, 目标在垂直方向上移动 72 个像素, 目标 45% 的区域超出了检测区域, 在第 33 帧时完全丢失左脚目标; 同样在第 46~47 帧时, 右脚的部分区域也超出了搜索区域, 导致跟踪失败。而对于 MMKCF 算法, 采用运动模型首先预估出下一帧的检测区域位置, 再完成模板匹配, 因此目标始终在检测区域内, 最后成功地跟踪目标。

4 TurtleBot 机器人目标跟踪

机器人在跟随目标时多采用恒定的线速度与角速度, 但是人体运动时不能保证是恒定速度, 如果机器人速度设定太低有可能会跟不上人体目标的运动, 同样速度设定较高又会导致电机频繁地启停。因此, 本文根据人体位置采用自适应改变线速度和角速度的方法。

4.1 机器人线速度

设其中一只脚的 MMKCF 跟踪框宽度为 a , 高度为 b , 则在某一时刻跟踪框左上角顶点的坐标为 (x, y) 。在每一只脚的跟踪框内选取 4 点, 其坐标分别为 $(x+a/3, y+b/3)$, $(x+a/3, y+2b/3)$, $(x+2a/3, y+b/3)$, $(x+2a/3, y+2b/3)$, 双脚共取 8 个位置点。采用机器人的 Kinect 摄像头计算出这 8 个位置点的深度信息再求算术平均值, 作为目标人体到机器人的距离。机器人会根据预先设定好的最大最小跟踪距离阈值判断该距离下目标是否在范围内, 如果超出或者小于跟踪距离的范围, 则机器人都会保持静止。当目标出现在跟踪距离范围内时, 线速度计算公式为 $V = cS + u$ 。其中, c 为线速度系数, $c = (V_{\max} - V_{\min}) / (S_{\max} - S_{\min})$; $V_{\max}$ 为最大线速度; $V_{\min}$ 为最小线速度; $S_{\max}$ 为最大跟踪距离; $S_{\min}$ 为最小跟踪距离; u 为速度补偿参数, $u = V_{\min} - cS_{\max}$ 。

4.2 机器人角速度

设 MMKCF 跟踪框长度为 a , 高度为 b 。选双脚目标框间中点位置作为角速度计算依据。实验中视频窗口大小为 640×480 , Kinect 摄像头的水平

视场角为 57° ^[11], 将边长 640 像素的窗口分为 57 个子区间, 再将整个子区间分为 3 段, 其中第 1~28 子区间作为第 1 段, 第 29 子区间作为第 2 段, 30~57 子区间作为第 3 段. 假设左脚跟踪框中心 (x_{lm}, y_{lm}) 为 $(x_l + a_l / 2, y_l + b_l / 2)$, 右脚跟踪框中心 (x_{rm}, y_{rm}) 为 $(x_r + a_r / 2, y_r + b_r / 2)$, 则这 2 个中心连线中点水平方向坐标为

$$x_m = x_{lm} + (x_{rm} + a_r / 2) / 2 - (x_{lm} + a_l / 2) / 2.$$

当中点 x_m 位于第 1 段区间时, 角速度 ω 为

$$\omega = \omega_{\max} - \left\lceil \frac{57x_m}{640} \right\rceil \cdot \frac{\omega_{\max}}{28}.$$

其中, $\omega_{\max}$ 为最大角速度; $\lceil \cdot \rceil$ 表示向下取整.

当中点 x_m 位于第 2 段区间时, 目标位于图像中间区域, 角速度为 0; 当中点 x_m 位于第 3 段区间

时, 角速度为 $\omega = \left\lceil \frac{57x_m}{640} \right\rceil \cdot \frac{\omega_{\max}}{28} - \frac{29}{28} \omega_{\max}.$

4.3 ROS 下机器人跟踪试验

本文采用 TurtleBot 机器人和 ROS(robot operating system)^[18]完成实验设计. 机器人上搭载了微软 Kinect V1 视觉传感器和处理设备间数据通信的 2.4 GHz Intel Core i5 CPU 笔记本. 安装 Ubuntu 16.04 版本的操作系统和 ROS Kinetic 版本. 根据 RGB 相机传输的实时视频进行目标跟踪, 通过深度相机判断双脚与机器人之间的距离以及 MMKCF 跟踪算法所预估双脚的位置, 得到机器人移动所需的线速度与角速度. 在室内走廊进行了真实场景下的人体目标跟踪实验, 行人以较快步行行走; 图 7 所示为机器人视角的 MMKCF 算法跟踪示意图, 图 8 所示为机器人跟踪示意图.

图 7 机器人视角的 MMKCF 算法跟踪示意图

5 结 语

本文针对小型低高度机器人在人体跟踪时存在的问题, 采用了足部跟踪的方法. MMKCF 算法有效提升了目标在快速运动以及光照变化情况下

图 8 机器人跟踪示意图

的跟踪效果. 在有光照及快速运动情况下的视频仿真实验结果可知, MMKCF 算法与原始 KCF 算法、BACF 算法、SAMF 算法和 HCF 算法相比, 跟踪误差下降, 跟踪准确率得到提高. 此外, 运用基于 ROS 的移动机器人, 对人体双脚进行了现实场景下的跟踪试验, 在光照变化、较复杂背景和快速运动情况下验证了足部运动模型应用于人体目标跟踪具有优良的鲁棒性和实时性.

参考文献(References):

- [1] Willeke T, Kunz C, Nourbakhsh I. The history of the mobot museum robot series: an evolutionary study[C] //Proceedings of the 14th International Florida Artificial Intelligence Research Society Conference. New York: ACM Press, 2001: 514-518
- [2] Henriques J F, Caseiro R, Martins P, *et al.* High-speed tracking with kernelized correlation filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596
- [3] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C] //Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2005: 886-893
- [4] Bertinetto L, Valmadre J, Golodetz S, *et al.* Staple: complementary learners for real-time tracking[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

- Los Alamitos: IEEE Computer Society Press, 2016: 1401-1409
- [5] Wang Q, Gao J, Xing J L. DCFNet: discriminant correlation filters network for visual tracking[OL]. [2020-02-28]. <https://arxiv.org/abs/1704.04057>
- [6] Bhat G, Johnander J, Danelljan M, *et al.* Unveiling the power of deep tracking[C] //Proceedings of European Conference on Computer Vision. Heidelberg: Springer, 2018: 493-509
- [7] Wu Y, Lim J, Yang M H. Object tracking benchmark[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848
- [8] Wang X P, Zhao Q J, Tan H K. Self-tuning visual tracking based on multiple motion models[C] //Proceedings of the 35th Chinese Control Conference. Los Alamitos: IEEE Computer Society Press, 2016: 3990-3994
- [9] Duan R, Fu C H, Kayacan E. Tracking-recommendation-detection: a novel online target modeling for visual tracking[J]. Engineering Applications of Artificial Intelligence, 2017, 64: 128-139
- [10] Chen P, Wang P J, Wang J D, *et al.* Design and motion tracking of a strip glove based on machine vision[J]. Neurocomputing, 2018, 288: 20-29
- [11] Cheng Xi. Pedestrian following method for indoor mobile robot based on depth image[D]. Changsha: Hunan University, 2018(in Chinese)
(程熙. 基于深度图像的室内移动机器人行人跟随[D]. 长沙: 湖南大学, 2018)
- [12] Kajita S, Kanehiro F, Kaneko K, *et al.* The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation[C] //Proceedings of the IEEE International Conference on Intelligent Robots and Systems. Los Alamitos: IEEE Computer Society Press, 2001: 239-246
- [13] Li Y, Ding S H, Zhai Q, *et al.* Human feet tracking guided by locomotion model[C] //Proceedings of the IEEE International Conference on Robotics & Automation. Los Alamitos: IEEE Computer Society Press, 2015: 2424-2429
- [14] Henriques J F, Caseiro R, Martins P, *et al.* Exploiting the circulant structure of tracking-by-detection with kernels[C] //Proceedings of the 12th European Conference on Computer Vision. Heidelberg: Springer, 2012: 702-715
- [15] Galoogahi H K, Fagg A, Lucey S. Learning background-aware correlation filters for visual tracking[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 1144-1152
- [16] Li Y, Zhu J K. A scale adaptive kernel correlation filter tracker with feature integration[C] //Proceedings of European Conference on Computer Vision. Heidelberg: Springer, 2015: 254-265
- [17] Ma C, Huang J B, Yang X K, *et al.* Hierarchical convolutional features for visual tracking[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2015: 3074-3082
- [18] Hu Chunxu. Development practice of ROS robot[M]. Beijing: China Machine Press, 2018: 11-19(in Chinese)
(胡春旭. ROS 机器人开发实践[M]. 北京: 机械工业出版社, 2018: 11-19)