计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(1):35−44 [doi: 10.15888/j.cnki.csa.006178]

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

室内单目机器人视觉目标发现与跟随①

骆　颇

(复旦大学 计算机科学技术学院, 上海 120013)

摘　要: 本文研究了室内单目机器人上的视觉目标人发现与跟随问题, 分为场景变化检测, 目标人检测, 目标人视觉追踪和目标人主动跟随几个部分, 主要研究了场景变化检测算法和目标人视觉追踪算法. 高速的场景变化检测算法通过对场景建模来分析该场景是否变化, 为目标检测部分提供潜在变化帧和潜在变化区域. 实验结果表明能够提高系统运行速度, 减少机器人运行时的卡顿. 视觉目标追踪算法结合表观模型和 SLAM 过程得到的地图点信息, 估计目标区域内属于背景的部分, 减少由于遮挡和目标尺度变化对于追踪算法的表观模型的影响, 实验结果相比于对比算法取得较大效果提升. 本文尝试使用近年来效果较好的深度神经网络来进行目标检测. 使用小型深度网络并加强对于室内场景下人的学习, 在运行速度和检测效果方面取得较好的平衡. 在视觉目标人的发现和跟踪的基础上, 我们实现了机器人的跟随. 由于单目视觉仅能够提供目标的方向信息, 所以机器人主动跟随的目标是保持目标人在成像平面的水平居中位置. 在目标无遮挡和部分遮挡的情况下, 机器人能够成功的跟随人.

关键词: SLAM; 视觉目标检测; 视觉目标追踪; 机器人跟随; 在线学习

引用格式: 骆颇.室内单目机器人视觉目标发现与跟随.计算机系统应用,2018,27(1):35–44. http://www.c-s-a.org.cn/1003-3254/6178.html

Visual Person Discovery and Following of Indoor Monocular Robot

LUO Po

(School of Science and Technology, Fudan University, Shanghai 120013, China)

Abstract: This study researches on the visual detection and following of object people by indoor monocular robots，which includes scene change detection algorithm, visual object people detection algorithm, visual object tracking algorithm, and robot following, with focuses on scene change detection algorithm and visual object tracking algorithm. A high speed scene change detection algorithm judges whether the scene changes by constructing scene models. If the scene changes, the algorithm outputs the change region, which is used by the visual object detection algorithm. The experiment shows this algorithm speeds up the system and alleviates the latency of robots. The visual object tracking algorithm combines the appearance model and map information obtained in SLAM process. The map information can judge which part of object bounding box is actually the background, which can reduce the effect of occlusion and object scale change on appearance model. This algorithm improves visual object tracking performance in the experiments. This paper applies the latest deep neural networks to do visual object people detection. We train a small deep neural network with enhancement on indoor people, which achieves a good balance between running speed and detection performance. Based on the visual detection and visual tracking of target, we accomplish robot following. Since monocular robots can only get the bearing information of target, the goal of robot following is keeping the target in the horizontally middle point of image plane. The robot can successfully follow human even if the person is partially occluded.

Key words: SLAM; visual object detection; visual object tracking; robot following; online learning

① 基金项目: 上海市科委基础研究领域项目 (14JC1402200); 上海市科委项目 (15511104303)
收稿时间: 2017-04-18; 修改时间: 2017-05-04; 采用时间: 2017-05-19; csa 在线出版时间: 2017-12-22

随着技术的发展, 机器人逐步从最初的军事、航天等领域逐步扩展到工业制造, 并向民用领域发展. 服务机器人是机器人家族中一个较为年轻的成员, 主要分为专业领域的服务机器人和个人服务机器人. 服务机器人大多可以移动. 在家用场景下存在着对目标进行发现和跟随的需要.

本文研究室内场景下低成本单目机器人上视觉目标人的发现和跟随. 相关的研究工作在进行目标人追踪时主要依赖人脸检测[1], 头肩检测[2]或者是目标人手持彩色板[3]的方式进行, 应用上存在较大局限. 本文针对整个人进行发现和追踪, 能够适应遮挡, 不需要人为发出指令. 本文主要研究场景变化检测算法和视觉目标追踪算法, 并介绍了目标人检测和主动跟随的实现方法.

场景变化检测算法分析可能出现人的图像帧和区域. 和此需要相关的主要是视频分析领域, 同时定位、建图和运动目标追踪 (Simultaneous Localization And Mapping and Moving Object Tracking, 即 SLAMMOT) 领域和多体运动恢复结构 (Multibody Structure From Motion) 领域. 视频分析的运动区域检测领域有大量的研究成果, 方法主要有帧间差分法[4], 光流法[5]和背景减除法[6]. 然而监控视频中的运动分析方法主要适用于摄像头固定的场景. 多体运动恢复结构[7]和 SLAMMOT[8]的研究主要利用投影几何约束关系, 结合光流或者是占用网格等方法来发现运动物体并且进行持续的追踪. 这两种方法应用于家庭场景的主要问题在于对家庭场景中常见的自运动物体如风扇、植物等比较敏感. 本文针对本研究场景提出了基于关键场景的超像素聚类的候选运动区域检测算法. 通过快速高效的场景变化检测, 为视觉目标人检测提供潜在变化帧和潜在变化区域, 提高系统运行速度, 减少机器人卡顿.

视觉目标追踪领域近年来取得了很多新的研究成果[9-11]. 但是现有的研究成果主要面向摄像头参数未知的场景, 仅利用 2 维图像信息来对目标进行建模和追踪, 并未考虑到图像序列中包含的场景结构信息. 且目前的追踪算法主要是通过检测进行追踪, 在模型更新的时候大多直接将当前帧的目标框内的图像认为是属于目标的, 未直接考虑遮挡、目标框内包含部分背景信息等问题. 针对以上问题, 本文研究结合表观模型与机器人在同时定位和建图时得到的场景信息, 减少由于遮挡, 目标区域包含背景信息等原因导致的漂移.

视觉目标人检测方面使用深度神经网络. 目标检测近二十年取得了很多的研究成果[12,13]. 尤其是在 2012 年以后, 深度学习在目标检测问题上取得了较大突破[14]. 近几年, 研究人员提出了大量的神经网络结构来解决目标检测问题[15-17]. 目前目标检测较好的神经网络需要使用计算显卡来进行运算, 而低成本机器人并不配备计算显卡, 且 CPU 的计算能力有限. 本文针对室内场景下人的检测训练一个小型深度网络, 在检测效果和运行速度方面取得一个较好的平衡.

1 系统概述

本系统总体分成三部分: 相关 Web 页面获取模块、Web 信息抽取模块、知识表示模块. 系统总体框图如图 1 所示.

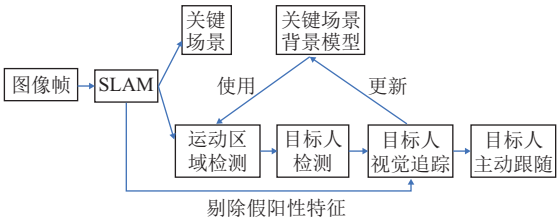


图 1 系统总体框图

在机器人采集到新的图像帧之后, 先进行 SLAM 过程. 本研究中 SLAM 模块使用 ORB_SLAM[18]. 待 SLAM 过程结束之后, 如果 SLAM 过程判断该位置是关键场景, 则建立关键场景背景模型. 在当前帧同时进行运动区域的检测. 如果当前帧存在显著的运动区域, 则目标人检测算法在当前帧的运动区域进行目标人检测. 如果在当前帧检测到目标人, 视觉追踪算法会持续追踪该目标人, 并且为机器人的主动跟随提供方向信息. 在追踪的过程中 SLAM 所获取的场景信息可以用来辅助目标追踪算法. 依据视觉目标追踪提供的目标方向信息, 控制机器人跟随目标.

2 各模块的算法设计与实现

2.1 场景变化检测

本文研究的重点在于在常见的家庭场景, 基于低成本的单目摄像机的机器人平台来较好地完成对目标, 主要是人, 进行跟随的任务. 在跟随任务中跟目标保持一定距离. 在单目 SLAM 能够较为稳定工作的假设下, 在常见家庭场景下进行运动区域检测的主要关注点在于有较高的运行速度, 对光照变化、自运动、震动等

具备良好的适应性, 能够减少对目标进行较为耗时的检测算法的调用.

基于应用场景的需要, 本文设计了基于关键场景的超像素聚类的候选运动区域检测算法. 关键场景的选取依据 SLAM 过程中所分析出来的关键帧位置. 采样关键帧前后位置及关键帧图像进行超像素分割, 并在 HSI 空间中对超像素进行聚类, 建立背景模型. 当机器人采集到新的图像帧时, 将机器人采集的图像进行超像素分割, 选取空间位置相邻的场景模型, 在 HSI 空间中相对于场景模型进行聚类, 依据与聚类中心和聚类半径之间的关系计算超像素的背景概率.

2.1.1 场景模型

为了构建关键场景的场景模型, 在 SLAM 过程得到的关键场景 (关键帧) 位置, 抽取临近的 p 帧图像, 序号记为 t. 使用 SLIC (Simple Linear Iterative Clustering)[19] 算法进行超像素分割, 在 HSI 空间中提取 HS 通道信息进行聚类. 算法步骤如下:

① 将第 t 帧图像进行超像素分割, 得到 N_t 个超像素. 每个超像素 $sp(t,r)(t=1,...,m,r=1,...,N_t)$ 由一个特征向量 f_t^r 来表示.

② 使用 meanshift 聚类算法对特征池 $F=\{f_t^r|t=1,...,m,r=1,...,N_t\}$ 进行聚类, 得到 n 个聚类. 每个聚类 $\text{clst}(i)(i=1,...,n)$ 由聚类中心 $f_c(i)$ 和聚类半径 $r_c(i)$ 表示.

2.1.2 模型使用

当新的图像帧到达的时候, 将新的图像在 RGB 空间中分割为 N_t 个超像素. 为了计算该帧每个像素属于前景的概率, 我们在 HSI 空间中评估每个超像素, 并且计算对应超像素对应于空间位置最为相近的场景变化的概率, 每个超像素的概率由它属于哪个聚类和在特征空间中与聚类中心之间的距离这两个因素决定.

第一个因素在于超像素相对于所属的聚类 $clst(i)$ 而言, 该超像素是否位于聚类半径 $r_c(i)$ 内. 第二个因素是一个权重因子, 这个因子考虑了距离的影响. 一个超像素的特征 f_t^r 在特征空间中距离对应的聚类中心 $f_c(i)$ 越远, 那么这个超像素属于该聚类的可能性越低, 每个超像素的置信度由以下公式度量:

$$C(r,i)=\begin{cases}\exp(-\lambda_d\times\frac{r_c(i)}{(f_t^r-f_c(i))^2}) & \text{if not in cluster radius}\\ -1*\exp(-\lambda_d\times\frac{(f_t^r-f_c(i))^2}{r_c(i)}) & \text{if in cluster radius}\end{cases}$$
$$\forall r=1,...,N_t,\forall i=1,...,n \tag{1}$$

其中参数 $r_c(i)$ 代表了 $clst(i)$ 在特征空间中的聚类半径, λ_d 是一个归一化项 (在实验中设置为 2). 综合考虑超像素 $sp(t, r)$ 所属于的聚类, 以及和对应的聚类中心之间的距离, 得出该超像素的目标置信度值 C_r^s.

2.1.3 候选区域生成

依据针孔摄像头模型, 计算 5 米处 1.5 米高的直立人在图像中的成像外接矩形面积. 当图像帧中存在大于该面积 1/3 且概率大于 0 的连通区域, 则认为该帧是潜在运动帧, 该区域周围一定范围的区域为潜在目标区域.

2.1.4 模型在线学习

当机器人重新进入相似位置和场景的时候, 在去除图片中人的信息之后, 将新抽取的 H 个图像, 加入训练图像集. 这个过程保留了过去在该场景下的多个图像信息. 每 K 次经过该场景时, 使用保存的信息更新一次表观模型. 具体更新算法同训练过程.

2.1.5 算法实验

本文在录制的 3 个室内场景视频中进行了实验. 图 2 为样例图. 图 2(c) 中的风扇处于打开并转动状态.

(a) BM1 (b) BM2 (c) BM3

图 2 场景变化检测样例图

表 1 为候选目标区域检测算法在测试数据上的表现结果. 表中计算时间减少时, 对比基准设定为每 5 帧执行 1 次检测算法. 测试中候选目标区域检测算法的运行速度为 39 fps. 目标检测耗时为 0.73 s/帧. 如果记候选目标区域检测算法每帧处理时间为 t_1, 记检测算法每帧处理时间为 t_2, 检测比为 p. 那么计算时间减少可以由以下公式计算得出:

$$1-\frac{t_1+p*t_2}{0.2*t_2} \tag{2}$$

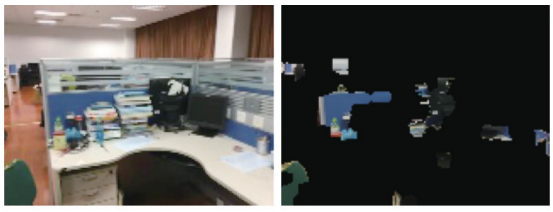
表 1 候选目标区域检测算法结果

表格	测试总帧	检测帧数	检测比 (%)	计算时间减少 (%)
BM1	4436	66	1.4	75.43
BM2	2847	131	4.6	59.43
BM3	1336	9	0.67	79.08

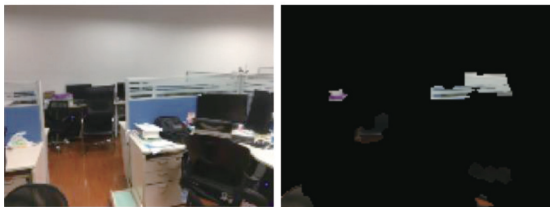
表 1 中计算时间减少一栏结果表明, 本文提出的基于关键场景超像素聚类的候选目标区域检测算法能够有效减少调用检测算法的次数, 降低了总体的计算

时间. 值得一提的是, 减少调用检测算法的次数不仅仅是降低总体的计算时间, 更重要的是使得机器人在运行的时候能够较少卡顿, 提高交互性.

我们分析了实验中误报的帧, 发现误报主要集中在以下两点. 第一点是如果相机对于场景遍历比较稀疏, 那么当相机以不同的位置或朝向经过类似场景的时候, 图像中所包含的场景区域不一样, 会有一些误报. 第二点是在光线充足的镜面反射区域, 视角的轻微差距便会导致图像有较大的区别, 导致误报较多. 图 3 是未遍历场景误报和镜面反射误报.



(a) 图像帧 (b) 误报区域分割



(c) 图像帧 (d) 误报区域分割

图 3 (a)(b) 为未遍历场景误报; (c)(d) 为镜面反射误报

从实验结果可以看到, 本文设计的方法能够显著提高运行速度. 本文的算法优点在于能够适应一定程度的光照变化, 对于家庭场景中常见的自运动物体具备良好的适应性. 本文设计的方法劣势在于忽略了物体在场景中的相对位置信息如人从沙发上站起来, 运动区域检测算法并不能够鲁棒地分析出该运动. 应该认识到的是, 这个劣势在本文的研究场景下并不会造成障碍. 如果机器人一直在伴随人, 那么人的图像信息会被过滤掉, 并不会进入背景模型.

2.2 视觉目标人检测

2.2.1 神经网络结构

目前目标检测较好的神经网络需要使用计算显卡来进行运算, 而低成本机器人并不配备计算显卡, 且 CPU 的计算能力有限. 在本文的实验平台上, 使用大型的深度网络 yolo, 检测一帧 640×480 像素的图片需要约 10 s. 而在家庭场景中进行目标人的发现并不需要支持 1000 类甚至更多类别的物体识别能力. 因而需要一个对人的检测效果较好且运算速度快的神经网络. 本文中的目标检测网络使用和 tiny-yolo 相同的网络结构. tiny-yolo 的创新之处是将检测和定位问题转换成一个回归问题, 只需要对图像进行一次处理就可以得到该图像中包含的所有目标的位置. tiny-yolo 包含 9 个卷基层, 其中前 4 个卷积层后面有一个 2×2 的最大值池化层. 网络结构如图 4 所示.

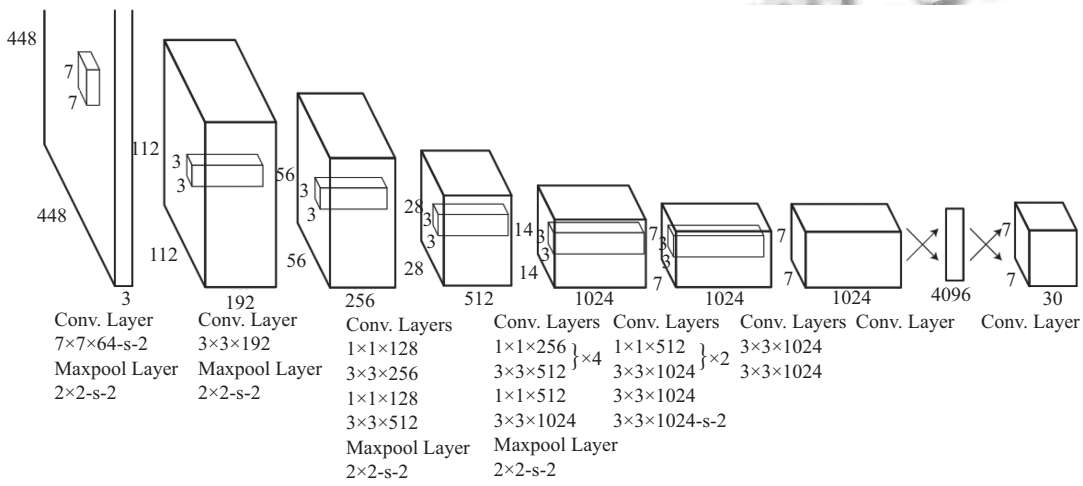


图 4 tiny-yolo 网络结构[17]

2.2.2 数据

本文中神经网络的训练使用 pascal voc(pascal visual object classes challenge)[20]数据集加上我们搜集人的图片进行训练, 其中 voc 数据共 16552 张, 我们搜

集的人的数据共 1897 张, 其中走廊场景 241 张, 室内场景 1656 张. voc 中的图像横向的尺寸大多在 500*375 左右, 纵向的尺寸大多在 375*500 左右. 我们搜集的数据尺寸为 460*640. 样例训练图片如图 5 所示.

(a) 样例 1 (b) 样例 2

(c) 样例 3 (d) 样例 4

图 5 训练数据样例. (a)(b) 来自 voc[20]; (c)(d) 是我们搜集的

测试数据使用我们标注的室内场景数据共 482 张, 所有测试数据中的人均未在训练数据中出现过. 样例图片如图 6 所示.

图 6 测试数据样例

2.2.3 算法实验

目标检测实验对比了 tiny-yolo, yolo 和我们的模型. 在 voc 数据上训练得到的 tiny-yolo 模型记为 tiny-yolo-voc. 我们在 voc 数据集和搜集的人数据上训练得到的模型记为 tiny-yolo-voc-lab. tiny-yolo-voc 和我们的模型使用同样的训练参数, 区别在于我们的模型加入了更多的人的图片. yolo 模型使用作者提供的预训练的模型. 评测指标为 AP (Average Precision), AP 是 PR(Precision Recall) 曲线下面的面积. 实验结果如表 2 所示.

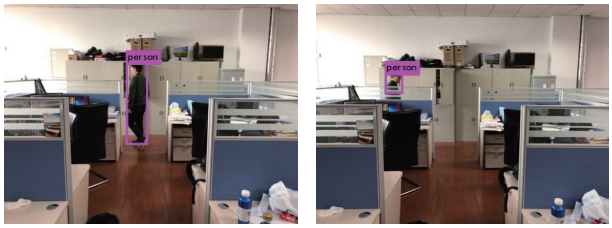
表 2 目标检测结果

模型	AP
tiny-yolo-voc	0.5136
tiny-yolo-voc-lab	0.6952
yolo	0.8137

检测结果样例如图 7.

(a) tiny-yolo-voc-lab

(b) tiny-yolo-voc



(c) yolo

图 7 目标检测结果样例

实验结果表明, 即使是网络规模较小的神经网络, 在训练集中包含更多室内场景下包含人的图片时, 能够取得较好的效果, 缩小和大型神经网络的差距.

2.3 视觉目标追踪

本模块算法详细流程如图 8 所示. 目标追踪的过程为在当前帧的前一帧的目标位置周围寻找目标. 当前帧的目标位置为目标概率最大的区域. 由视觉表观模型和地图点信息共同决定每个像素属于目标的概率或者叫置信度. 表观模型部分使用超像素追踪算法[21].

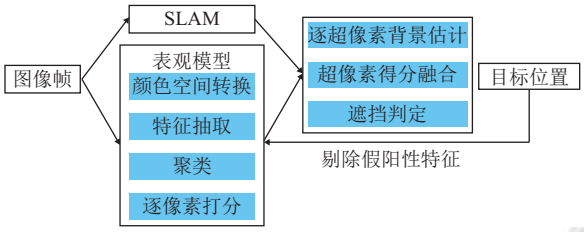


图 8 视觉目标追踪算法详细流程

2.3.1 表观模型

为了构建目标和背景的表观模型, 可以从 m 个训练帧中提取每个像素的标签信息. 对于第 t 帧中坐标位置为 (x, y) 的像素. 我们可以得到该像素的标签:

$$y_t(i,j)=\begin{cases}1 & \text{if } pixel(t,i,j) \text{ is in target}\\ -1 & \text{if } pixel(t,i,j) \text{ is not in target}\end{cases} \tag{3}$$

$y_t(i,j)$代表超像素 (t,i,j) 的标签, 假设目标可以由一组超像素来表示, 且这种表示并不会与目标和背景的边界差别过大, 关于目标和背景的先验信息可以由下式来表示:

$$y_t(r)=\begin{cases}1 & \text{if } sp(t,r) \text{ is in target}\\ -1 & \text{if } sp(t,r) \text{ is not in target}\end{cases} \tag{4}$$

在通常的追踪场景下, 这个信息难于获得. 在追踪开始之前从一组样本中推断先验信息是一种可行的方式. 以下方法可以用于从样本中推断超像素和目标之前的关系.

① 将第 t 帧中的目标周围区域进行超像素分割, 得到 N_t 个超像素. 每个超像素$sp(t,r)(t=1,...,m,r=1,...,N_t)$由一个特征向量$f_t^r$来表示.

②使用 meanshift 聚类算法对特征池$F=\{f_t^r|t=1,...,m,r=1,...,N_t\}$进行聚类, 得到 n 个聚类. 每个聚类$clst(i)(i=1,...,n)$由聚类中心$f_c(i)$和聚类半径$r_c(i)$表示.

③每个$clst(i)$对应于训练帧中的图像区域$S(i)$, 对每个$clst(i)$计算两个得分, $S^+(i)$, $S^-(i)$. 前一个分数表示聚类面积$S(i)$和目标区域的交集大小, 后面一个分数表示聚类面积$S(i)$在目标区域外的大小. $S^+(i)/S^-(i)$的值越大, 在训练帧中区域$S(i)$属于目标的可能性越大. 我们给每个聚类一个介于[1, –1]之间的打分来代表每个聚类的前景-背景置信度.

特征表示使用归一化的 HSI 颜色空间直方图. sp 代表 super pixel(超像素), clst 代表 cluster(聚类).

使用超像素的优点在于, 即使有少量的背景超像素出现在目标区域, 它们大部分也会被聚类到背景超像素所在的聚类, 且可以进行逐像素的前景估计. 使用超像素的劣势在于, 随着目标的运动, 目标的尺寸、形态的变化, 目标区域会被更多的背景超像素所占据. 因而模型在更新的过程中, 更多的背景超像素被当做目标, 模型逐渐的就会漂移. 本文结合 SLAM 过程所建立的地图信息来处理模型偏移问题.

2.3.2 追踪

2.3.2.1 表观模型打分

当新的图像帧到达的时候, 首先在前一帧的目标区域周围提取一个搜索区域, 并且分割为 N_t 个超像素. 为了计算该帧的置信度, 我们评估每个超像素, 并且计算对应的得分, 每个超像素的初始得分由它属于哪个聚类和在特征空间中与聚类中心之间的距离这两个因素决定. 第一个因素在于如果一个超像素属于聚类$clst(i)$, 那么$clst(i)$的前景置信度表明了该超像素属于前景的可能性. 第二个因素是一个权重因子, 这个因子考虑了距离的影响. 一个超像素的特征f_t^r在特征空间中距离对应的聚类中心$f_c(i)$越远, 那么这个超像素属于该聚类的可能性越低, 每个超像素的置信度由以下公式度量:

$$w(r,i)=\exp(-\lambda_d\times\frac{\|f_t^r-f_c(i)\|^2}{r_c(i)})\ \forall r=1,...,N_t,\forall i=1,...,n \tag{5}$$

$$C_r^s=w(r,i)\times C_i^c,\forall r=1,...,N_t \tag{6}$$

其中$w(r,i)$表示基于特征f_t^r(第 t 帧中第 r 个超像素$sp(t,r)$的特征) 和$f_c(i)(sp(t,r))$属于的聚类的特征中心) 的权重. 参数$r_c(i)$表示$clst(i)$在特征空间中的聚类半径, λ_d 是一个归一化项 (在试验中设置为 2). 综合考虑超像素$sp(t,r)$所属于的聚类, 以及和对应的聚类中心之间的聚类, 得出该超像素的置信度值C_r^s.

对于整个图像帧, 通过以下步骤得到每个像素的置信值. 对于搜索区域内每个属于超像素$sp(t,r)$的像素打分为C_r^s, 对于搜索领域之外的像素打分为–1.

2.3.2.2 地图点打分

依据表观模型对新的图像帧中的目标领域进行打

分之后, 依据 SLAM 所建模的地图点信息, 对于地图点所在的超像素判断是否属于背景, 进而对置信度打分进行调整.

在 SLAM 过程中计算得到的地图点有两个重要的信息: 一是共见次数; 二是地图点的位置.

共见次数就是某一个地图点在多少个图像帧中被发现到, 即地图点在该帧图像中的投影点和多少帧中的投影点可以关联上. 目标表面会存在边界等能够提取出角点的位置, 且符合在不同图像帧之间的几何约束, 但是运动的目标表面无法存在持续而稳定的符合极点几何的特征点. 使用简单的可见次数阈值就可以过滤掉大部分错误匹配的目标表面角点. 地图点周围的超像素块属于背景的可能性随着地图点共见次数升高而降低. 我们用以下公式来计算地图点所在超像素块的的置信度.

$$\begin{aligned} &C_r^m = -2*(sigmoid(n(r,k)-10)-0.5) \\ &\forall r = 1,...,N_t, Vk = 1,...,M_t \\ &\text{if } sp(t,r) \text{ is neighbour of map point } k \end{aligned} \tag{7}$$

$n(t,k)$代表图像帧 t 中地图点 k 的共见次数. 模型设定共见次数大于等于 10 次的地图点位置的超像素为背景.

2.3.2.3 打分融合

表观模型的得分和地图点的得分通过求均值的方式进行融合, 融合的位置仅限地图点周围的超像素, 没有地图点的超像素的打分仅由表观模型决定.

$$C_r = \begin{cases} (C_r^m + C_r^s)/2 & \text{if } C_r^m \text{ exist} \\ C_r^s & \text{if } C_r^m \text{ not exists} \end{cases} \quad \forall r = 1,...,N_t \tag{8}$$

2.3.2.4 遮挡判定

当概率最大的目标候选区域的平均置信度低于阈值且置信度较低的区域伴随大量可靠地图点, 即可判定目标被遮挡. 具体的遮挡程度以及目标可见部分的位置和大小使用类似于 camshift[22]中所使用的质心法来估算. 计算步骤如下:

① 以超像素为单位进行高斯模糊.

② 使用 meanshift 寻找概率密度最高的区域.

③ 在 meanshift 算法收敛之后, 使用公式$s = 2\times\sqrt{M_{00}/256}$来获取目标可见区域.

④ 继续步骤 2 和 3 直到收敛.

⑤ 如果步骤 3 得到的 s 低于当前目标尺寸一定阈值, 则判定目标遮挡

如果判断目标被遮挡, 那么该帧的目标图像信息不会用来更新表观模型.

2.3.3 表观模型在线更新

表观模型在线学习使用滑动窗口的学习模式. 在追踪过程中存储 H 个图像帧构成的序列, 每隔 U 个图像帧, 放入一个新的图像帧进入该序列, 并且删除序列中最老的帧. 这个过程保留了过去 $H*U$ 个图像帧的一个记录. 对于这个序列中的每个帧, 保留它的目标状态和超像素分割的结果. 位于目标区域外或者是地图点判断为属于背景的的超像素作为负样本, 位于目标区域内且未被地图点信息判断为属于背景的超像素作为正样本. 每 W 帧使用保存的信息更新一次表观模型. 具体更新算法同训练过程.

2.3.4 实验

本文主要的研究目的是帮助室内机器人进行目标的主动跟随, 确定机器人路径规划的目标, 机器人路径规划的目标由视觉目标追踪算法提供. 由于无法构造完全一样的场景和目标移动过程来对比多个视觉目标追踪算法且目前常用的目标追踪数据集并不包含录制时镜头内参信息, 而 SLAM 系统需要该信息来进行建图, 故录制实验数据集, 本节实验在离线视频上评估追踪算法在应对场景变化, 目标遮挡等问题时的表现.

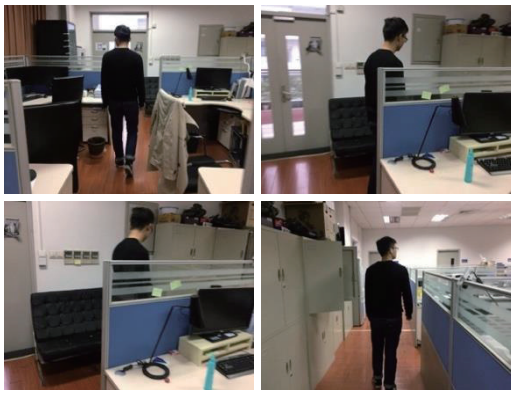
本文 3 段视频上比较了 4 个算法, 比较的 4 个算法是 CT (Compressive Tracking)[9], TLD (Tracking-Learning-Detection)[10], SPT (Super Pixel Tracking)[21], STRUCK (Structured output tracking with kernels)[23].

2.3.4.1 视觉目标追踪数据集

视频数据集的录制设备为 iPhone 6s, 自动对焦参数设置为 0.74F. 数据集为 lab1, lab2, lab3. 数据集录制选取常见的室内场景. 视频中的目标, 主要为人在室内正常的走动, 过程中有不同程度的遮挡, 尺度变化和光照变化. 视频如图 9 所示.

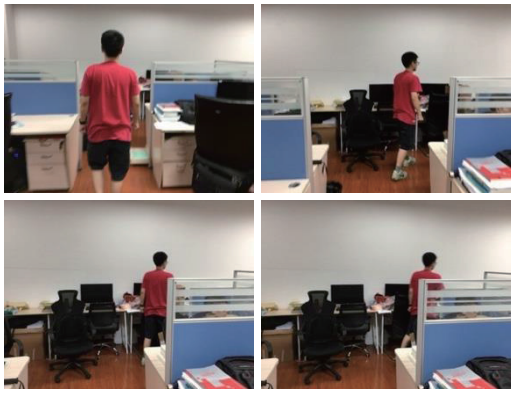
2.3.4.2 视觉目标追踪评测指标

实验结果使用两个指标来衡量. 第一个评价指标是成功率, 帧内追踪得分为$score = area(ROI_T \cap ROI_G)/area(ROI_T \cup ROI_G)$. ROI_T 是追踪算法得到的目标框, ROI_G 是标注的目标真实位置. 如果在某一个帧里的得分 (score) 大于 0.5, 则认为该帧追踪成功. 第二个评价指标是中心位置偏移 (center location error). 偏移值为追踪算法得到的目标框中心坐标和标注的目标中心之间的距离长度.



(a) 第 3440 帧, 3557 帧, 3565 帧, 3632 帧

(b) 第 2302 帧, 2344 帧, 2442 帧, 2539 帧



(c) 第 6062 帧, 6183 帧, 6221 帧, 6232 帧

图 9 目标追踪的数据集. (a) Lab1 视频中目标短暂严重遮挡; (b) Lab2 视频中目标长期部分遮挡; (c) Lab3 视频中目标迅速且持续被严重遮挡

2.3.4.3 实验结果和分析

表 3 和 4 给出了算法评测结果. 效果最好的用字体加粗来表示, 效果次好的用斜体来表示.

从表 3 中可以看出, 本文提出的基于单目 SLAM 的目标追踪算法的成功率在 3 个测试视频中的 1 个视频上取得第一, 1 个视频上取得第二, 1 个视频上与第二相差无几的效果. 尤其是本文提出的算法 SPT+MapPoint, 相对于 SPT 在长期部分遮挡的情况下取得了较大的提升. 由于追踪算法在丢失之后得出的目标位置是随机的, 并不能很好的反映算法的定位能力, 因而平均中心位置偏移在此仅列出, 具体的价值需要由使用场景来确定.

表 3 算法追踪成功率 (单位: %)

数据	算法					
	TLD	CT	STRUCK	SPT	SPT+ MapPoint	相对SPT 提升
Lab1	77.56	58.41	**90.13**	57.44	*82.39*	42.75
Lab2	60.40	30.40	*69.60*	48.20	**84.20**	74.68
Lab3	72.60	*82.60*	**88.80**	81.80	81.00	–0.97

表 4 算法中心位置偏移 (单位: 像素)

数据	算法					
	TLD	CT	STRUCK	SPT	SPT+ MapPoint	相对SPT 错误减少 (%)
Lab1	*47.23*	103.01	**22.94**	113.73	55.96	50.79
Lab2	72.41	128.55	*64.27*	101.04	**41.79**	58.64
Lab3	61.02	35.84	**20.85**	28.21	*21.49*	23.8

测试视频 Lab1 中目标有较为短暂的严重遮挡, TLD 算法和 CT 算法逐步向背景漂移, STRUCK 表现最好. TLD 算法筛选出大量代表性正负样例, 在短期的严重遮挡并伴随视角的快速变化的情况下, 迅速丢失目标, 但是当目标重新以相似视角出现时可以找回目标. CT 算法由于采用了逐帧更新的模式, 在遮挡之后迅速漂移, 目标重新出现之后无法找回. STRUCK 筛选出的正负支撑向量能够有效区分目标和背景, 在短暂的严重遮挡下表现最好. SPT 算法由于在模型跟新的时候采取和 CT 类似的不加区分的将目标框内的图像信息认作是目标, 迅速漂移. 本文提出的 SPT+MapPoint 的算法能够有效判断遮挡, 阻止不属于目标的图像信息进入模型, 且在目标脱离遮挡之后, 重新追踪成功. 相对于 SPT 取得了显著的 42.75% 的提升.

Lab2 中目标同时存在光照变化, 部分遮挡和尺度变化. CT 算法依旧最先漂移. TLD 算法能够较好处理尺度变化, 但是对于目标的外观变化, 光照变化等情况存在一些问题, 当这些问题同时出现的时候, 算法的表现一般. STRUCK 表现较好, 但是在持续的遮挡情形下, 也会逐步漂移. 本文提出的算法 SPT+MapPoint 由于能够较好的进行遮挡判定, 相对于 SPT 算法取得了 74.68% 的相对提升.

Lab3 中目标从最初的无遮挡到部分遮挡到最终被

严重遮挡的变化过程很快, 在这个过程中 TLD 算法最先丢失, CT 紧随其后. STRUCK 算法表现最好. 由于目标很快被严重遮挡. 本文提出的算法相比于 SPT 而言, 没有提升.

以上视频的总体结果来看, 在比较的四种算法中, STRUCK 表现最好. 本文提出的算法性能高于 STRUCK 或与 STRUCK 接近. 但是相对于没有利用地图点信息的原始 SPT 算法而言, 在利用地图点信息之后, 取得了非常明显的提升. 在家用机器人追踪目标的应用场景下, 面对经常出现的长期部分遮挡, 光线变化, 目标尺度变化等问题时, 本文提出的算法在实验数据上取得较好的成绩.

2.4 机器人主动跟随

2.4.1 跟随目标

由于单目摄像头无法得到可靠的深度信息. 视觉目标追踪算法仅能给机器人提供目标相对于机器人正前方的角度偏移, 因而机器人的主动跟随的控制目标是使得目标人位于机器人摄像头的水平成像中心上.

$$c = \arg\min_{c}(C_{tx} - C_{ix}) \tag{9}$$

其中 c 代表机器人的控制指令, C_{tx} 代表目标在图像中的水平位置, C_{ix} 代表图像的水平中心点.

2.4.2 跟随实验

本文的主要研究内容是目标人的发现与视觉追踪, 并且实现机器人的主动跟随, 不涉及到机器人的全局路径规划和避障能力的研究. 跟随部分实现机器人在无障碍和有障碍两种情形下的主动跟随.

本研究基于的机器人平台是小强机器人, 其主要参数如表 5 所示.

表 5 机器人平台主要参数

设备名称	设备参数
CPU	I7-4500U酷睿双核1.8 GHz
内存	8 G
显卡	Intel HD Graphics 4400
摄像头	单目
红外	0.2 m避障能力

图 10 和图 11 为机器人主动跟随结果. 机器人运动控制的目标是保持跟随的人位于机器人摄像头成像水平中心位置. 在图 10 和图 11 中, 人最初在右边, 机器人面朝人前进, 当人移动到左边之后, 机器人转而向左前方前进. 图 11 中间图中人被凳子遮挡.

图 10 无遮挡机器人主动跟随实验结果

图 11 有遮挡机器人主动跟随实验结果

3 结语

本文详细介绍了在低成本轮式单目机器人上对于目标人的视觉发现和跟随的研究. 本文主要研究了场景变化检测算法和视觉目标追踪算法, 并介绍了神经网络在单目机器人上进行目标人检测的经验. 实验表明结果表明基于关键场景的场景变化检测算法运行速度快 (39 fps), 能够有效减少检测算法的运行次数, 提高系统运行效率, 减少机器人卡顿. 针对室内场景下人进行训练的小型深度网络在检测效果和运行速度之间取得了较好的平衡, 和大型深度网络的差距不大. 结合 SLAM 过程改进的超像素追踪算法能够较好的处理遮挡, 光照变化等问题. 在实验平台上, 机器人在有障碍物存在的情况下成功跟随人.

参考文献

1 Feyrer S, Zell A. Detection, tracking, and pursuit of humans with an autonomous mobile robot. Proceedings of International Conference on Intelligent Robots and Systems (IROS'99). Kyongju, Korea. 1999. 864–869.

2 Hirai N, Mizoguchi H. Visual tracking of human back and shoulder for person following robot. Proceedings of 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Kobe, Japan. IEEE. 2003. 527–532.

3 Hassan MS, Khan AF, Khan MW, *et al*. A computationally low cost vision based tracking algorithm for human following robot. Proceedings of the 2nd International Conference on Control, Automation and Robotics (ICCAR). Hong Kong, China. 2016. 62–65.

4 Collins RT, Lipton AJ, Kanade T, *et al*. A system for video surveillance and monitoring. Pittsburgh: Carnegie Mellon University, 2000.

5 Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. Proceedings of the 7th International Joint Conference on Artificial

Intelligence. Vancouver, BC, Canada. 1981. 674–679.
6 Stauffer C, Grimson WEL. Adaptive background mixture models for real-time tracking. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Fort Collins, CO, USA. 1999. 252.
7 Vidal R, Yi M, Soatto S, *et al*. Two-view multibody structure from motion. International Journal of Computer Vision, 2006, 68(1): 7–25. [doi: 10.1007/s11263-005-4839-7]
8 Wang CC, Thorpe CS, Thrun S, *et al*. Simultaneous localization, mapping and moving object tracking. The International Journal of Robotics Research, 2007, 26(9): 889–916. [doi: 10.1177/0278364907081229]
9 Zhang KH, Zhang L, Yang MH. Real-time compressive tracking. In: Fitzgibbon A, Lazebnik S, Perona P, *et al*, eds. European Conference on Computer Vision. Berlin, Heidelberg. Springer. 2012. 864–877.
10 Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409–1422.
11 Babenko B, Yang MH, Belongie S. Visual tracking with online multiple instance learning. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, USA. 2009. 983–990.
12 Viola P, Jones Ml. Rapid object detection using a boosted cascade of simple features. Proceedings of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Kauai, HI, USA. 2001. I-511–I-518.
13 Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA. 2005. 886–893.
14 Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe, NV, USA. 2012. 1097–1105.
15 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA. 2014. 580–587.
16 Girshick R. Fast R-CNN. Proceedings of IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1440–1448.
17 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 779–788.
18 Mur-Artal R, Montiel JMM, Tardós JD. ORB-SLAM: A versatile and accurate monocular SLAM system. IEEE Transactions on Robotics, 2015, 31(5): 1147–1163. [doi: 10.1109/TRO.2015.2463671]
19 Achanta R, Shaji A, Smith K, *et al*. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(11): 2274–2282. [doi: 10.1109/TPAMI.2012.120]
20 Everingham M, Van Gool L, Williams CKI, *et al*. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 2010, 88(2): 303–338. [doi: 10.1007/s11263-009-0275-4]
21 Wang S, Lu HC, Yang F, *et al*. Superpixel tracking. Proceedings of 2011 IEEE International Conference on Computer Vision (ICCV). Barcelona, Spain. 2011. 1323–1330.
22 Bradski GR. Real time face and object tracking as a component of a perceptual user interface. Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision, 1998. WACV'98. Princeton, NJ, USA. 1998. 214–219.
23 Hare S, Golodetz S, Saffari A, *et al*. Struck: Structured output tracking with kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(10): 2096–2109. [doi: 10.1109/TPAMI.2015.2509974]