

# 中国科学技术大学

# 学士学位论文



## 面向移动服务机器人的行人追踪跟随的 研究与实现

作者姓名： 韦清

学科专业： 计算机科学与技术

导师姓名： 陈小平 教授

完成时间： 二〇一九年五月十五日

University of Science and Technology of China  
A dissertation for bachelor's degree



# **Research and Implementation of People Tracking and Following with Mobile Service Robots**

Author: Qing Wei

Speciality: Computer Science and Technology

Supervisor: Prof. Xiaoping Chen

Finished time: May 15, 2019

## 目 录

中文内容摘要	3
英文内容摘要	4
第 1 章 绪论	5
1.1 研究背景及意义	5
1.2 相关工作	5
第 2 章 视觉行人追踪算法	7
2.1 行人检测	7
2.1.1 常用人工特征描述子	8
2.1.2 分类器	15
2.2 基于动态的追踪	16
2.2.1 Boosting 算法	17
2.2.2 卡尔曼滤波	17
2.2.3 粒子滤波	19
2.2.4 均值漂移算法	20
2.2.5 相关滤波	21
2.2.6 GOTURN	22
2.2.7 TLD	22
第 3 章 ROS 导航和可佳导航简介	25
3.1 ROS 导航	25
3.1.1 坐标系转换	25
3.1.2 里程计	25
3.1.3 建图	26
3.1.4 定位	26
3.1.5 导航控制	26
3.2 可佳导航	27

第 4 章 以可佳机器人为基础的行人追踪系统	28
4.1 输入设备	28
4.1.1 Kinect	28
4.1.2 2D 激光	28
4.2 视觉追踪系统	28
4.2.1 总体架构	28
4.2.2 目标人物注册	28
4.2.3 行人追踪	29
4.2.4 目标丢失恢复	30
参考文献	31

## 中文内容摘要

服务机器人是一种能完成有益于人类的服务工作的半自助或全自主型机器人，分为个人/家庭服务机器人和专业机器人。专业机器人一般在特定场景中使用，如商业服务、物流、医疗、救援等；而个人/家用服务机器人主要在日常生活场景中进行与人进行交互，提供家政服务、陪伴、娱乐、辅助学习等多种功能，包括家政机器人、娱乐休闲机器人、助老助残机器人等。其中，个人/家庭服务机器人为本文研究内容所适用的对象。

随着人工智能与物联网技术迅速发展，个人家庭用机器人、公共服务机器人的应用场景、服务模式和市场也在不断拓展，成为了当下的一个研究热点。为了精准理解当前环境和有效执行指令，能够精确可靠地自动识别目标人物并对其进行追踪陪同，是这类移动服务机器人的人机交互中的一项重要且必要的功能。

本文将针对室内移动机器人的行人跟随问题做如下三个方面的研究：

(1) 常用目标跟随算法的原理与实现：主要从行人检测和追踪两个角度进行调研论述。主要介绍了用于行人检测的几个常用特征，如颜色直方图、LBP、HOG、SIFT 特征，基于机器学习的几种分类器，包括 KNN、SVM、RandomForest、AdaBoost，以及几种常用的追踪算法，包括卡尔曼滤波、粒子滤波、均值漂移、相关滤波等。

(2) ROS 上的机器人导航技术：主要包括 ROS 提供的 ROS 导航包，以及中科大研发的可佳机器人的导航功能。

(3) 可佳机器人上行人跟随系统的实现：结合行人的 HOG 特征和 HSV 直方图特征，使用 CSR-DCF 算法进行行人追踪，并单独训练一个 SVM 分类器以防止追踪器发生误判/漏判，以及帮助追踪器从追踪失败中恢复。

**关键词：**计算机视觉；机器人；行人检测；目标追踪

## Abstract

A

**Key Words:**

## 第 1 章 绪论

### 1.1 研究背景及意义

服务机器人是一种半自助或全自主工作的机器人。它能完成有益于人类的服务工作（不包括从事生产的设备），可以在真实且不可预测的环境中自动进行决策和行动来完成确定的任务。服务机器人分为个人/家庭服务机器人和专业机器人。专业机器人一般在特定场景中使用，如商业服务、物流、医疗、救援等；而个人/家用服务机器人主要在日常生活场景中进行与人进行交互，提供家政服务、陪伴、娱乐、辅助学习等多种功能，包括家政机器人、娱乐休闲机器人、助老助残机器人等。其中，个人/家庭服务机器人为本文研究内容的应用对象。

随着人工智能与物联网技术不断成熟，服务机器人作为智能硬件之一，不断地丰富其自身功能及其实现更强大的性能，服务机器人市场也在迅速发展，家庭用机器人、智能公共服务机器人的应用场景和服务模式不断拓展。为了精准理解当前环境和有效执行指令，能够精确可靠地自动识别目标人物并对其进行追踪陪同，是这类移动服务机器人的人机交互中的一项重要且必要的功能。这项功能可以用于在博物馆或医院等场景中对用户进行指引，在家庭中陪伴用户、与用户进行交互，助老、助残等，在投入应用后，可以有效减少在这些场景中的人工成本，便捷用户的生活和工作。

移动机器人的核心技术包括导航定位、地图创建、路径规划、任务分配和目标跟踪等。在本文中，会主要介绍目标跟踪，尤其是其中的行人定位模块，通过对 ROS 导航的简要介绍涉及机器人导航定位、地图创建、路径规划的大致概念，并最终实现一个有简单导航模块和较为鲁棒的行人追踪模块的机器人系统。

### 1.2 相关工作

自从有了服务机器人的概念以来，就有了很多对于机器人目标跟随技术的研究。移动机器人由于可携带多种传感器，由此衍生出了各种使用了传感器的行人识别和追踪方法。如使用视觉图像、激光传感器、热成像传感器<sup>[1]</sup>、声音传感器<sup>[2]</sup>等，由于单一传感器信息较少，也出现了不少多传感器融合信息的行人定位方式<sup>[3]</sup>。

激光传感器由于还被广泛运用到机器人的定位、建图、导航中，是机器人最

为重要的传感器，所以也常常被运用到行人跟随的任务中。不过激光传感器由于成本限制，多使用 2D 激光来检测同一水平面的物体，竖直空间上可以检测到的范围有限，所以激光行人追踪的方法最为经常使用的特征是行人的双腿<sup>[4]</sup>。在激光图像中，人的双腿会有一种明显的模式，与背景区分开，通过滤波器抽取人的腿部特征，进行聚类，便可得到一个人腿模式识别器，再结合粒子滤波或卡尔曼滤波等技术，便可以实现对行人腿部的持续追踪。但这种识别器在有遮挡时，或暂时失去目标后，无法准确地重新建立对目标的追踪。此外，从人腿信息中基本不可能对不同行人加以区分，所以为了系统的鲁棒性，行人腿部识别通常还是需要加入视觉信息，如结合面部识别，来进行一些情况下的行人的区分<sup>[5-6]</sup>。

视觉行人追踪领域已有很多研究，是计算机视觉领域中的一个重要研究对象。在机器人跟随用户的任务中使用视觉行人追踪，已有方法中包括了对行人的面部识别<sup>[7-]</sup>（要求行人必须面对相机），衣着识别<sup>[8-9]</sup>（主要通过对衣着颜色特征的提取），身体轮廓识别<sup>[1]</sup>（使用 HOG、边缘描述子等特征），行人步态识别<sup>[10-11]</sup>等。为了有效利用可佳机器人所配备的 Kinect RGB-D 相机，在本文的系统实现中，会对行人检测和追踪常用的特征进行分析，选择合适的特征和追踪器，使用视觉行人追踪得到人物在 RGB 图像中的位置，再结合 RGB-D 相机的深度图像，得到目标行人在地图中的 3D 坐标。



## 第 2 章 视觉行人追踪算法

计算机视觉中的行人追踪，主要包括密集跟踪方法，即基于行人检测和识别的追踪，以及稀疏跟踪方法，即基于目标动态的追踪。

密集跟踪算法实际上并没有“跟踪”物体，而是在视频不同的时间点的一系列帧上扫描和检测物体的位置。由于每次的目标检测都是独立地在当前帧上进行的，所以每次检测时，都需要处理图像中的所有像素，所以用这种方法进行目标跟踪，计算量会比较大。此外，这种方法对分类器的要求也会较高，对于目标被遮挡的情况也不能很好地进行处理。

由于目标的运动通常是连续的，我们可以根据物体的动态信息，对其可能的运动轨迹进行预测，并结合其上一帧所在位置和对当前帧的观察，得出其当前位置，这就是稀疏跟踪方法。由于已知物体在上一帧时的位置，所以对当前帧识别时，只需要检测上一帧物体所在位置附近的像素，这样一来，相对于密集跟踪方法，就减少了大量的计算。此外，由于我们结合了对物体运动的预测和观察来进行估计，在一些情况下准确度也会较高，例如，当目标发生较大形变或被暂时遮挡时，目标检测很可能下会直接失败，但由于在追踪时我们还记录了物体最后出现的位置和对其行为的预测，所以可以在一定准确程度上继续追踪物体。但在这种算法在物体速度较快时，由于不会进行全图扫描，也可能会失去对物体的追踪，当目标物暂时从视野中消失一段时间时，可能难以重新找回物体。

比起基于检测的追踪，动态的目标追踪是在行人追踪任务中的主要方法，但行人检测在追踪算法中也是不可少的一环。下面将主要介绍在单张图像上的行人检测算法，和以此为基础的动态行人追踪。

### 2.1 行人检测

经典的行人检测方法包括提取人工特征，将待检测目标作为正样本，不含该目标的图像作为负样本，使用分类器进行分类，再在全图上进行匹配。此外还有使用卷积神经网络（convolutional neural networks）来提取图像特征或进行分类的方法，是现在行人检测研究的主要方向，但深度神经网络在目前还难以同时兼顾准确和实时性要求，此处主要介绍人工特征和支持向量机（Support Vector Machines, SVM）、Boosting 等分类器的使用。

### 2.1.1 常用人工特征描述子

特征描述子是一种对图片的表示方法，它通过提取图片中的关键信息并丢弃多余信息来对图片信息进行简化。通常地，特征描述子算法可以将一个 RGB 三通道的图片转化成一个特征向量。为了做到精确地进行图像识别、目标检测，我们必须首先明确什么是关键的、有用的信息，什么是冗余信息。在目标检测领域，有以下的常用特征描述子被提出，并都有过较为成功的应用。

#### 1. 颜色直方图

颜色特征是物体最为直观的特征之一，它具有旋转不变性，且不受目标的大小和形状的变化影响，在颜色空间中分布大致相同，从而具有较高的鲁棒性。颜色直方图是描述颜色特征最常用的描述子，它是对目标表面颜色分布的统计，描述了不同色彩在图像中所占的比例，具有稳定性好、抗部分遮挡、计算方法简单和计算量小等优点。但它无法描述图像中颜色的局部分布及每种色彩所处的空间位置，即无法描述图像中的某一具体的对象或物体，只是对一个图像块的颜色分布的统计，所以也有其使用局限性。

在行人检测领域，颜色直方图通常能够很好地描述行人的肤色或所着服饰的颜色，但在检测的时间段内，如果行人的衣着服饰发生改变，可能会导致在接下来的时间检测失败。此外，杂乱的背景色和不同光照强度和颜色也会产生较大影响。

颜色直方图可以基于不同的颜色空间进行统计，其中，最常用的是 RGB 空间和 HSV 空间。数字图像一般都以 RGB 三通道形式进行存储，故这里首先以 RGB 空间为例。分别统计每个像素的 R、G、B 数值出现的频度，绘制出它们分别对应的直方图，以一张人物照片 2.1 为例，图 2.2 为其 RGB 分别对应的统计直方图。将直方图上每一个桶（bin）的高度作为一个分量，便可以得到一个统计向量，将 R、G、B 分别对应的直方图向量连接起来便可得到输入图像的 RGB 直方图特征向量。

虽然 RGB 图像被如此普遍地使用，但以 RGB 直方图作为图片的特征描述子却有一些缺点。首先，RGB 这三个重要的分量取值和其所生成的颜色之间的联系并不直观，即两种颜色在 RGB 空间中的距离无法描述它在人眼中的直觉色差。此外，使用 RGB 直方图的检测器容易受局部光照变化的影响而产生错误。所以在计算机视觉中，相比起 RGB 颜色空间，更常采用 HSV 颜色空间来表示颜色。HSV 是一种将 RGB 色彩空间中的点在圆柱坐标系中的表示方法，相对于

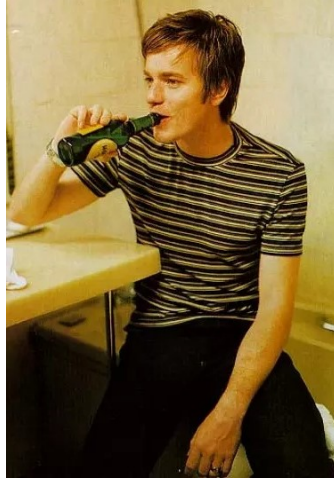


图 2.1 原图

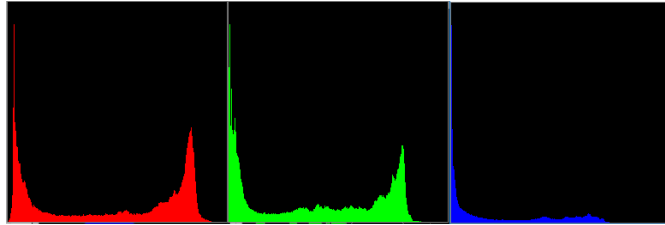


图 2.2 左：B 通道颜色直方图；中：G 通道颜色直方图；右：R 通道颜色直方图

RGB，它能够更加直观地表示色彩的明暗、色调以及鲜艳程度，方便进行颜色之间的对比。此外，由于 HSV 单独提取了颜色的明暗，也可以一定程度上抵抗光照明暗带来的影响。Sural et al.<sup>[12]</sup> 的实验显示，使用 HSV 直方图进行行人识别的结果比起 RGB 直方图可以有明显提高。

HSV 即色相 (Hue)、饱和度 (Saturation)、亮度 (Value)。色相即表示物体的颜色，取一个  $0^\circ$  到  $360^\circ$  的标准色轮，按照该颜色位置的角度来度量色相，即颜色点在圆柱坐标系中所在坐标的  $\varphi$  分量；饱和度是指颜色的强度或纯度，表示色相中灰色分量所占的比例，它使用从 0% (灰色) 至 100% (完全饱和) 的百分比来度量，即圆柱坐标系的  $\rho$  分量；亮度是颜色的相对明暗程度，通常使用从 0% (黑色) 至 100% (白色) 的百分比来衡量，即在圆柱坐标系中的  $z$  分量。这些颜色坐标一般都落在圆柱坐标系的一个圆锥体中，如图 2.3 所示。

在计算 HSV 直方图时，需要首先把 RGB 空间坐标映射到 HSV 空间。给定  $(r, g, b)$  分别是一个颜色的红、绿、蓝坐标，它们的值是在 0 到 1 之间的实数， $max$  为  $r$ ,  $g$  和  $b$  之中的最大值， $min$  为其中的最小值，则从  $(r, g, b)$  到  $(h, s, v)$  的转换公式如下：<sup>[13]</sup>

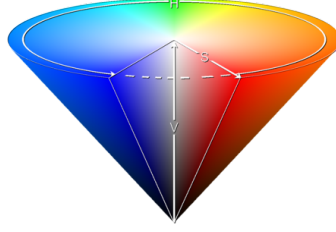


图 2.3 HSV 模型可以使用圆柱坐标系中的一个圆锥形子集表示

$$h = \begin{cases} 0^\circ & \text{if } \max = \min \\ 60^\circ \times \frac{g-b}{\max-\min} + 0^\circ, & \text{if } \max = r \text{ and } g \geq b \\ 60^\circ \times \frac{g-b}{\max-\min} + 360^\circ, & \text{if } \max = r \text{ and } g < b \\ 60^\circ \times \frac{b-r}{\max-\min} + 120^\circ, & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max-\min} + 240^\circ, & \text{if } \max = b \end{cases}$$

$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max-\min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

$$v = \max$$

HSV 直方图的计算与 RGB 类似，只是将颜色空间有所差异，我们同样使用图片2.1计算其 HSV 直方图，见图2.4。

## 2. 局部二值模式

局部二值模式 (Local Binary Pattern, LBP) 是一种用来描述图像局部纹理特征的特征描述子，具有旋转不变性和对光照变化不敏感等优点，由 Ojala et al.<sup>[14]</sup> 在 1994 年首次提出。

LBP 的计算方法非常简单。每个像素都与它相邻的八个像素按指定的顺序 (如顺时针、逆时针) 作比较，来确定其特征值。对于中心像素大于某个相邻像素的，该像素对应的二进制位设置为 0，否则设置为 1，比较了中心点相邻的八个像素后，就得到了一个 8 位的二进制数，这个数字即为中心像素的特征值，计算过程如图2.5为例。图 reffig:lbp 为对图2.1的灰度图上的每个像素计算 LBP 值得到的 LBP 图谱，可以看出，LBP 特征受噪点影响较重，能够突出表现物体的表面纹理，也能在一定程度上表现出物体的边缘。

为了使得 LBP 描述子有旋转不变性，Ojala et al.<sup>[15]</sup> 提出了一个 LBP 的具有

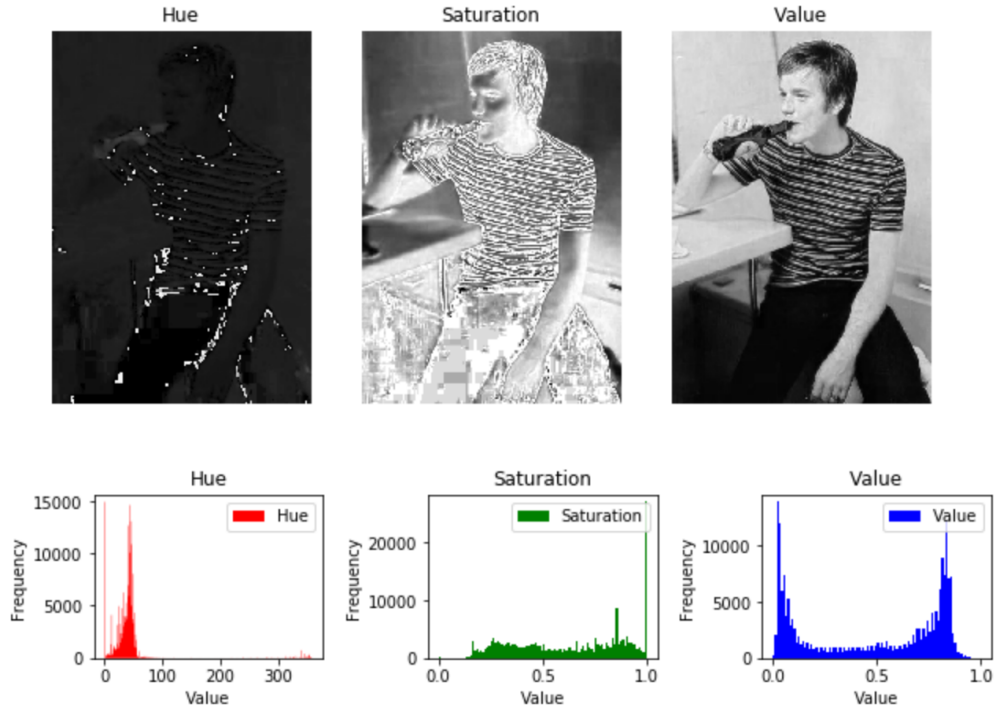


图 2.4 左：H 通道灰度图谱和颜色直方图；中：S 通道灰度图谱和颜色直方图；右：V 通道灰度图谱和颜色直方图

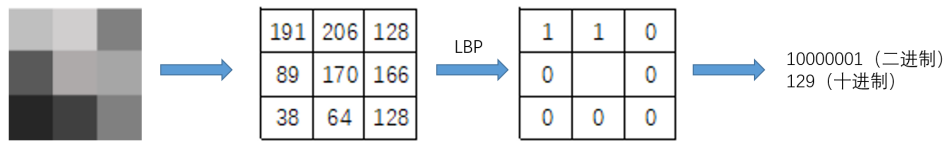


图 2.5 计算 3x3 像素块中中心点的 LBP 值

旋转不变性扩展方法，即不断旋转其邻域，得到一系列的 LBP 值，取其最小值作为该点的局部二值模式值。

在计算一个图片的 LBP 描述子时，首先将图片分成固定大小的单元格（如  $16 \times 16$  像素），在计算出每个像素的 LBP 值后，统计每个单元格内的 LBP 值直方图，再串联所有单元格的直方图，即可得到该图片的 LBP 特征向量。

### 3. 方向梯度直方图

方向梯度直方图 (Histogram of Oriented Gradient, HOG) 是目前行人识别中最广泛使用的特征描述子之一。Dalal et al.<sup>[16]</sup> 在 2005 年提出了由 HOG 特征结合 SVM 分类器进行行人检测的方法，在此之后，该方法被广泛应用到了图像识别领域中，并尤其在行人检测中获得了巨大的成功，也出现了很多改进和变体。

在 HOG 特征描述算法中，通过计算和统计图像局部区域的梯度方向直方图来构成特征。由于在物体的边缘和角落处图片的颜色会进行突变，故在这些区

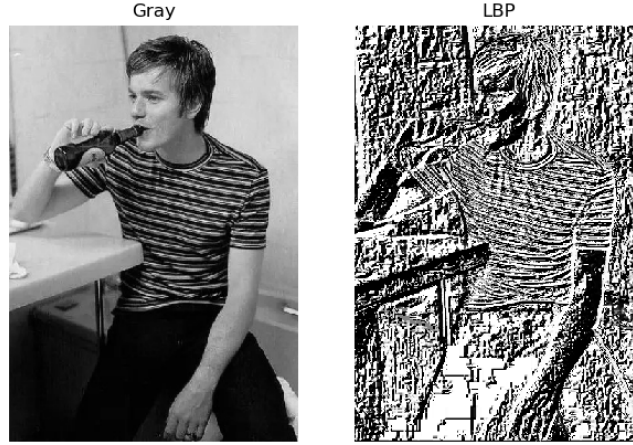


图 2.6 左：灰度图；右：由灰度图计算得到的 LBP 图谱

域，梯度的幅值也会变大。一般来说，边缘和角落比起平坦区域包含更多关于物体形状的信息，而通过对边缘和角度的描述，HOG 正可以很好地描述局部目标的表面质地和形状信息。但同时，由于梯度的性质，HOG 特征描述子对噪点比较敏感，且由于 HOG 主要描述了物体的轮廓，所以很难处理遮挡问题。

为了计算方向梯度，我们可以简单地使用核向量（kernel） $[-1, 0, 1]$  和  $[-1, 0, 1]^T$  对原图进行卷积，分别得到横向和纵向上的有向梯度。除了这种方法之外，还可以使用  $[-1, 1], [1, -8, 0, 8, -1]$  和 Sobel 算子等作为核向量，不过根据 Dalal et al.<sup>[16]</sup> 的实验，使用最简单的  $[-1, 0, 1]$  进行计算的梯度，在以 HOG 为特征进行的图像识别中效果反而最佳。

在每个像素处，方向梯度都具有大小和方向。对于彩色图像，我们分别计算 RGB 三个通道的梯度。对原图片上的每个像素点  $(x, y)$ ， $f(x, y)$  为其 R、G、B 值中的一个，该通道上的横向和纵向方向梯度为：

$$g_x(x, y) = [-1, 0, 1] * f(x, y) = -f(x+1, y) + f(x-1, y),$$

$$g_y(x, y) = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * f(x, y) = -f(x, y+1) + f(x, y-1)$$

由此便可得到有向梯度的幅值和方向：

$$|g(x, y)| = \sqrt{g_x(x, y)^2 + g_y(x, y)^2}$$

$$\theta(x, y) = \tan^{-1} \left( \frac{g_y(x, y)}{g_x(x, y)} \right)$$

使用以上公式在 RGB 颜色空间上计算图2.1的梯度值大小，如图2.7所示。可



见，这张梯度图谱已经省略了图中很多不必要的信息，如颜色几乎一致的背景等，且突出了人物的轮廓、五官、衣服的花纹等用于识别的重要信息。

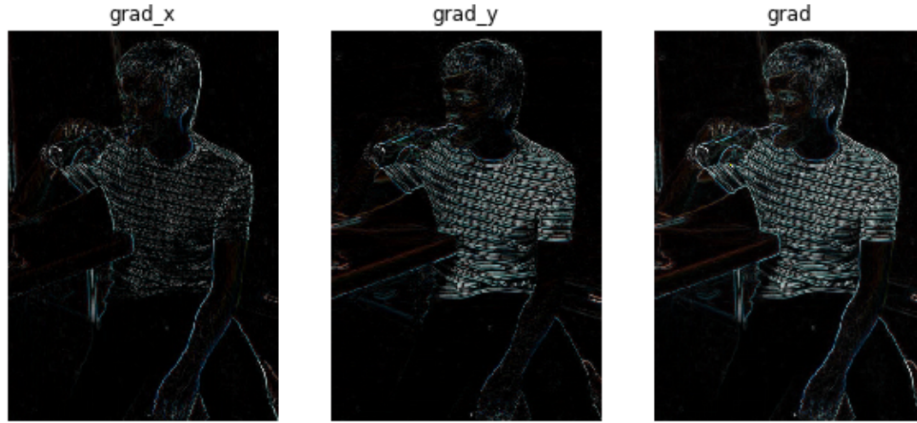


图 2.7 左：横向梯度绝对值；中：纵向梯度绝对值；右：梯度大小

此时在每个像素上得到的梯度有 3 个，分别对应原图的 RGB 分量，在计算 HOG 特征向量时，我们选取这三个通道的梯度幅度的最大值作为该点处的梯度大小，最大值对应的通道的梯度角度为该点处的梯度方向。

方向梯度直方图统计的即为梯度大小为权重的梯度方向。梯度的方向范围为  $[0^\circ, 360^\circ)$ ，但是我们实际在统计方向时，采用的却是  $[0^\circ, 180^\circ)$  的统计范围，通过计算  $\theta(x, y) \bmod 180^\circ$  来代替原有的角度值，即把相差  $180^\circ$  的两个角度视为同一个梯度方向。实验表明，这种统计方式得到的结果往往比采用  $[0^\circ, 360^\circ)$  范围的原方向效果更好<sup>[16]</sup>。

值得注意的是，由于图像的梯度是由各像素点周围的颜色值大小计算得到的，所以也会受光照的影响，例如，将所有像素值除以 2 来使图像变暗，这时所有梯度大小也会随之减半，梯度直方图每个 bin 的高度也会减半。而在一张图片中，每个局部区域的光照可能会有所不同，为了降低这些影响，在进行方向梯度统计时，并不会直接统计一整个图片的方向梯度直方图，而是以  $8 \times 8$  像素的区域为一个单元格（cell）来分别进行统计，再在此基础上进行规范化（normalization）。这样会降低光照等噪音对特征描述子质量的影响，使 HOG 描述子更加稳定鲁棒。这个过程可以图 2.8 为例。

在进行规范化时，最常用的方法是在单元格的基础上取一个更大的块（block），每块的大小为  $16 \times 16$  像素，即包括 4 个单元格，将每一个块的 4 个 HOG 向量作为一个整体进行 L2 规范化。即  $\mathbf{v} \leftarrow \mathbf{v} / \sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}$ ，其中  $\epsilon$  是一个防止零除的足够小的正常数。以每个直方图取 9 个 bins 为例，在规范化一个块

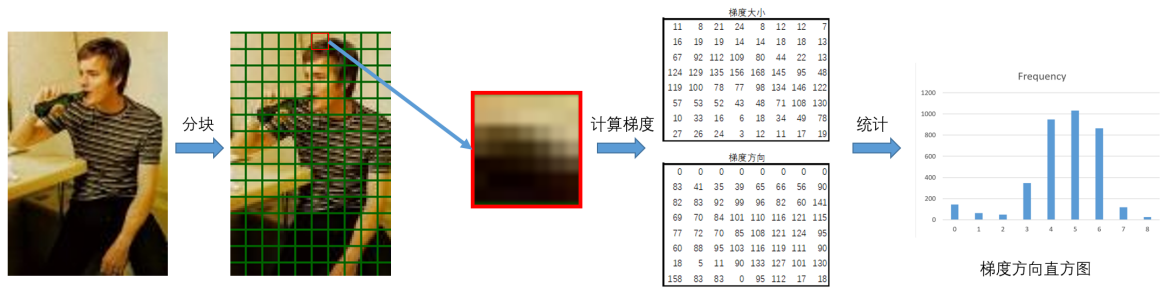


图 2.8 统计单元格梯度直方图的过程示意图

之后，我们得到了一个长为  $4 \times 9 = 36$  的向量，即这个块最终的 HOG 向量。再以 8 像素的步长 (stride) 移动这个块，对下一个块进行规范化，即两个相邻块之间有 2 个单元格的重叠。如此循环，直到整张图的每一个单元格都被计算，再将所有经过的块的向量合并在一起，作为整张图的 HOG 描述子。

#### 4. 尺度不变特征变换

尺度不变特征变换 (Scale-invariant feature transform, SIFT) 是一种不受图片尺度和旋转影响，并在一定程度上不受光照和相机视角影响的特征描述子，它将图像数据转换为有关局部特征的尺度不变坐标。同时，通过在空间和频率域的准确定位，SIFT 还可以减少遮挡和噪音带来的干扰。SIFT 可以使用有效的算法从图片中提取出大量的独特特征，可以在几乎所有尺度和位置上密集地覆盖图像，例如，一个  $500 \times 500$  像素的图像可以最多生成 2000 个稳定的特征点。SIFT 算法中独特的关键点描述方法，可以使单个特征从大型特征数据库中能得到高概率的匹配。但在较为杂乱的图像中，背景中的很多特征可能会生成错误的匹配，不过通过识别关于目标物及其在新图像中的位置，尺度和方向的关键点的子集，可以从完整匹配集中过滤出正确的匹配<sup>[17]</sup>。

为了最大限度地降低提取特征的计算量，SIFT 使用级联过滤 (cascade filtering) 方法来检测关键点，先使用高效的算法使用弱分类器检测出一些候选位置，然后再进一步详细检测，将更加耗时的计算只在通过了初始测试的候选点位置上进行。但即使如此，SIFT 特征描述子的计算量仍难以在行人检测和追踪中满足实时性要求<sup>[18]</sup>。故虽然 SIFT 特征描述子虽有诸多非常好的特性，但在有实时性要求的行人检测和追踪中并不常被使用。



### 2.1.2 分类器

分类器的作用是通过拟合一个从样本到标签的映射函数，将图像中的区域分类为目标对象或背景。分类器的训练需要正负样本，分别对应目标区域和背景区域。在对分类器进行一定规模的训练后，此时将一个图像上的区域块的特征向量作为分类器的输入，分类器即会给出该区域包含目标物体的概率。

在实际使用时，得到了一个图像帧后，选取不同的检测窗口的大小，即尺度，然后以给定尺度的检测窗口扫描整个图像，计算每个检测窗口的特征向量，使用分类器得出其中包含目标物的概率，如果这个概率超过给定的阈值，则认为该检测窗口中即包括待追踪物体。

#### 1. 最近邻算法

常用追踪器 TLD 中便采取了最近邻算法（K-Nearest Neighbors）作为级联分类器的最后一级。KNN 算法是从已有正确标签的样本集合中，找出与待分类特征向量最近的  $k$  个向量，以其中的多数的类别来决定该待测特征向量的分类。KNN 算法的思想非常简单，一般使用欧式距离或巴氏距离作为向量间距离的度量，但随着样本集变大，计算  $k$  个最近邻的计算量也会很大，变得相当耗时，此外，它对数据的局部结构也非常敏感，也没有考虑不同分量的权重。

#### 2. 随机森林

随机森林是一个包含多个决策树的分类器。决策树作为常用的机器学习方法有其缺陷，即训练时容易出现过学习，在训练集上具有低偏差和高方差的特点。而随机森林将多个深决策树的结果进行平均以降低方差，每个深决策树都在一个数据集上的不同部分进行训练，由此提高决策树的性能。事实上，随机森林分类器和 K 近邻分类器十分相似，都可以被看作是“对邻居向量结果的加权”。与 KNN 算法不同，随机森林在决定类别时可以评估各个分量的重要性，且对于不平衡的分类数据集，可以平衡误差，学习过程也较为快速。但随机森林仍然存在过拟合的问题，对于取值划分较多的分量上评估的权值偏大。

#### 3. SVM

支持向量机（Support Vector Machine, SVM）是一个监督学习的分类器。为了减少计算量，在用于行人识别时，一般采用线性支持向量机。它在特征空间中通过训练拟合一个最优超平面，在预测时通过将样本和最优超平面进行比较，判断它出现在某一类别的概率。对于线性不可分的样本，通过使用非线性算法将数据从低维空间转化到高维空间使其线性可分，再从高维空间采用线性回归。

SVM 对于在线学习的支持并不算好，通常只在初始化时进行对模型的拟合。

#### 4. AdaBoost

由于 SVM 计算量较大，为了提高速度，也可以使用 AdaBoost 代替。AdaBoost 是 Boosting 分类器的一种，它是一种级联分类器，即它是由几个弱分类器级联组成的。每一级分类器都会排除一些可能性较低的背景窗口，留下一些候选窗口传入下一级分类器进行计算。这样一来，在计算时大部分背景窗口在开始的几级分类器中就会很快被排除掉，剩下的很少一部分候选区域再通过后续的分类器，大大提高了整体运行速度。以使用 HOG 作为特征为例，相同情况下 HOG+AdaBoost 的预测时间只有 HOG+SVM 方法的十分之一。这里的弱分类器可以使用分类回归树（classification and regression tree, CART）、朴素贝叶斯模型等。

在训练初始化时，AdaBoost 给训练集的每个样本指定一个相同的权重，表明在训练时选取该样本进行学习的概率，接着调用弱分类器进行迭代学习，每次迭代后更新训练集上样本的权重，对于训练失败的样本赋予更大的权重，这样就使得下一轮的训练中，能够聚焦于那些更加难以分类、更富信息的样本上。

Boosting 算法还可以拓展为在线学习的分类器 online Boosting，这些在线学习器在追踪领域有着很广泛的应用<sup>[1]</sup>。在线学习，即在初始时训练一个简单模型后，可以在预测时进行学习的分类器。

## 2.2 基于动态的追踪

在基于动态的追踪中，我们假定在此前的几乎所有帧中都已成功地跟踪了对象，并保有对其运动模式和之前位置的记录，目标是根据这些已有信息找到在当前帧中物体的位置。物体的运动模型给了一个它当前位置的粗略预测，此外，还需要根据该目标对象在先前的帧中的外观记录（即特征）对物体的位置进行更加精确的估计，我们仍可以使用在目标检测中提过的物体特征，如颜色直方图，HOG 等。根据这些外观的模型和特征，我们可以在由物体运动模型所预测的物体位置的邻域中进行搜索，以提高速度和准确度。根据外观进行分类的原理与目标检测相同，但如何将物体的运动模型等信息目标检测结合起来，就需要使用下面提到的几种算法。

### 2.2.1 Boosting 算法

在目标追踪中，我们常使用在线分类器进行目标识别，即在运行时即使训练的分类器。分类器的训练过程中，最初的正样本由使用者提供，在一张包含目标人物的图像中手动或使用某种检测算法选出一个框（bounding box, bbox）。分类器将该图像中的 bbox 作为正样本，在 bbox 之外取出若干个负样本进行训练拟合。

在收到下一帧后，在物体原位置的所有相邻的位置上运行分类器，取得分最高的一个 bbox 作为当前帧的物体位置，再以当前帧检测出的 bbox 作为正样本，背景中提取负样本继续训练分类器。Boosting 算法原理非常简单，相应地，追踪效果也比较平庸，由于它每次会选择得分最高的位置作为当前帧的检测结果，可能并不能选取到正确的目标位置，且容易出现漂移现象。

在 Boosting 算法的基础上，Babenko et al.<sup>[19]</sup> 提出了多示例学习（multiple instance learning, MIL）算法。不同于传统的分类器将每一个实例进行分类的方法，MIL 将若干个实例归到一个包（bag）中，即正样本包和负样本包。只要包中的一个图像是正样本，就将其认为是正样本包，相对的，只有包中所有实例均为负样本，才将其认为是负样本包。构建正样本包的方法是首先包含目标物在当前图像中的图像块，并以此为中心，将该位置周围的小邻域中的图像块都包括其中。这样以来，即使被跟踪对象的当前位置不准确，只要将目标物作为中心的图像块被作为邻域放入了正样本包中，分类器就可以以它进行训练。

但 Boosting 和 MIL 算法都有着共同的缺点，它们难以判断出是否已经对目标物失去了追踪而错误地选定了其他位置，此外，当目标物在一段时间内被遮挡的情况下，它们都难以恢复追踪。

### 2.2.2 卡尔曼滤波

卡尔曼滤波是一种假定目标物体的运动服从线性高斯分布，以此对目标的运动状态进行预测，将预测结果与观察模型进行比较，根据误差更新预测模型，估计物体的当前位置的方法。它不是单纯地在前一帧目标物位置周围作检测，而是主动对其运动状态进行建模，预测它即将出现的位置<sup>[20]</sup>。

卡尔曼滤波器对离散时间的控制过程的状态  $x \in \mathbf{R}^n$  进行估计，该过程可以由一个马尔科夫链表示：

$$x_{k+1} = A_k x_k + B u_k + w_k$$

同时，提供了对系统当前状态的测量  $z \in \mathbf{R}^m$ ：

$$z_k = \mathbf{H}_k x_k + v_k$$

其中，随机变量  $w_k$  和  $v_k$  分别表示系统和测量误差，假定它们是互相独立的，并服从正态分布：

$$p(w) \sim N(0, Q),$$

$$p(v) \sim N(0, R).$$

$n \times n$  的矩阵  $\mathbf{A}$  将系统在时间  $k$  和  $k+1$  时的状态相关联起来，不存在驱动函数或系统噪音。 $n \times l$  的矩阵  $\mathbf{B}$  将控制输入  $u \in \mathbf{R}^l$  与系统状态  $x$  相关联。 $m \times n$  的矩阵  $\mathbf{H}$  将系统状态和对系统的测量  $z_k$  相关联。

我们根据时间  $k$  前的过程，计算  $\hat{x}_k^- \in \mathbf{R}^n$  作为为时间  $k$  时的先验（a priori）状态估计，并根据对系统状态的测量  $z_k$  计算后验（a posteriori）状态估计  $\hat{x}_k \in \mathbf{R}^n$ 。我们将先验和后验估计误差定义为：

$$e_k^- \equiv x_k - \hat{x}_k^-, e_k \equiv x_k - \hat{x}_k.$$

则先验和后验估计误差协方差分别为：

$$P_k^- = E[e_k^- e_k^{-T}],$$

$$P_k = E[e_k e_k^T]$$

使用先验估计  $\hat{x}_k^-$  和实际测量  $z_k$  来计算后验状态估计  $\hat{x}_k$ ：

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}(z_k - \mathbf{H}_k \hat{x}_k^-)$$

在上式中， $\mathbf{H}_k \hat{x}_k^-$  是根据先验估计对测量值的预测， $(z_k - \mathbf{H}_k \hat{x}_k^-)$  被称为测量残差（residual）。残差反映了先验估计及预测方法相对于实际测量的插值。 $n \times m$  的矩阵  $\mathbf{K}$  是最小化后验误差协方差的增益（gain）。将上式代入求  $P_k$  的公式中，取结果相对于  $\mathbf{K}$  的导数，并设为 0，可以求得：

$$\mathbf{K} = \frac{P_k^- \mathbf{H}_k^T}{\mathbf{H}_k P_k^- \mathbf{H}_k^T + \mathbf{R}_k}$$

卡尔曼滤波分为两组方程：时间更新方程和测量更新方程。时间更新方程根据当前状态和误差协方差估计，预测下一时间的先验估计；测量更新方程用于根

据所获得的新的测量，再结合先验估计来获取一个已优化的后验估计，这个后验估计又被传回时间更新方程。如此循环，完成一个预测-校正的过程，以自动化地对模型进行更新，对状态进行估计。

时间更新方程包括：

$$\begin{aligned}\hat{x}_{k+1}^- &= A_k \hat{x}_k + B u_k \\ P_{k+1}^- &= A_k P_k A_k^T + Q_k\end{aligned}$$

测量更新方程包括：

$$\begin{aligned}K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K(z_k - H_k \hat{x}_k^-) \\ P_k &= (I - K_k H_k) P_k^-\end{aligned}$$

$Q_k$  和  $R_k$  均为常数，分别与  $w$  和  $v$  相关，估计误差协方差  $P_k$  和增益矩阵  $K_k$  将会在计算中迅速收敛，并保持不变。

卡尔曼滤波器限定了系统噪声必须符合正态分布，且必须为线性系统，而在实际使用中，很难同时满足要求，此时精度就会较低。

### 2.2.3 粒子滤波

粒子滤波器 (particle filters) 是一种基于概率密度的粒子表示的顺序蒙特卡罗方法 (sequential Monte Carlo methods)，它可以应用在任意状态-空间模型，可以对非线性、非高斯系统的动态进行建模，是传统的卡尔曼滤波的一般化方法<sup>[21]</sup>。

首先对跟踪目标进行建模，并定义一种相似度度量确定粒子与目标的匹配程度。在目标搜索的过程中，它会先按照一定的分布（比如均匀分布或高斯分布）在全局撒一些粒子，统计这些粒子与目标的相似度，确定目标可能的位置。在可能性较高的位置上，下一帧加入更多新的粒子，确保在更大概率上跟踪上目标。

首先在目标周围均匀地或按高斯分布随机散布一些粒子，它们在图中的位置分别为  $\{x_i, i = 0, \dots, N\}$ ，每个粒子都计算其所在区域内的特征向量，与初始的目标区域的特征作比较，得到一个相似度，将这些相似度归一化得粒子的权重  $\{w_i, i = 1, \dots, N\}$ ， $\sum_{i=1}^N w_i = 1$ 。则在这张图像上，目标最可能在位置为  $\sum_{i=1}^N w_i x_i$ 。之后，我们根据每个粒子的权重  $w_i$  进行粒子重采样，在概率较大的位置多撒粒子，减少概率小的位置的粒子个数，将重采样后得到新的粒子集再通过顺序重要

性采样算法 (Sequential Importance Sampling) 得到新的粒子集, 用于下一帧的目标识别。

粒子滤波器相对于卡尔曼滤波器, 虽然适用范围更广, 但计算量也更大。

## 2.2.4 均值漂移算法

均值漂移算法 (MeanShift) 是用于定位图中概率密度最大的位置的算法, 常常结合 HSV 颜色直方图进行目标跟踪。以利用 HSV 颜色直方图模型为例, 给定了一个初始的包括目标行人的搜索窗口, MeanShift 算法首先将图片的 RGB 分量转化为 HSV 分量, 统计该搜索窗口内的 H 分量直方图, 将直方图归一化后, 我们便可以得到所有 H 值 ( $\in [0^\circ, 360^\circ]$ ) 对应的概率。在统计直方图时, 为了避免由于低光照导致的错误数据, 将 V 值低于某一阈值的像素点不予统计。接着, 将全图的所有像素值都用它的 H 分量所对应的概率表示, 得到的图像被称为反向投影图, 如图2.9所示。MeanShift 所谓的求图中概率密度最大的位置, 即是求反向投影图中平均值最大的窗口, 在 CamShift 中用一种类似梯度下降法的方法实现, 求得局部最大值。对于新的一帧, 首先求当前搜索窗口的质心位置, 即将像素概率作为权重, 求所有位置的加权平均, 接下来将这个质心作为新的搜索窗口的中心, 重复计算其质心, 如此循环, 直到搜索窗口收敛, 即为概率密度的局部极大值。

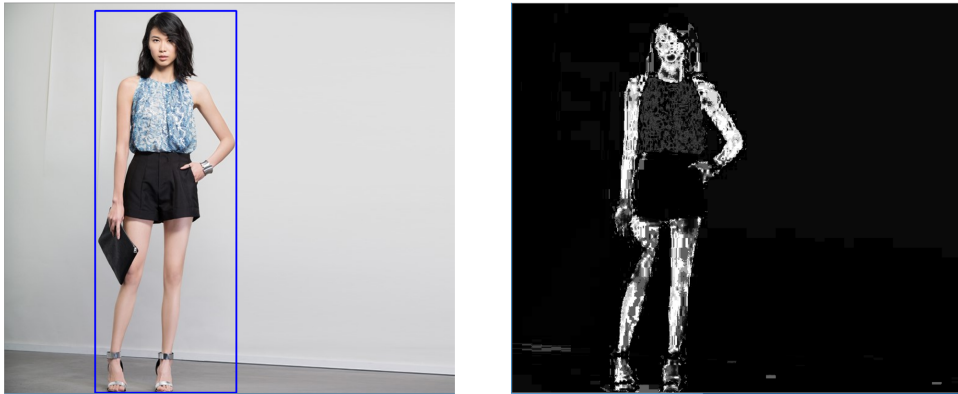


图 2.9 左: 原图像, 蓝色框内为初始提供的含目标行人的搜索窗口; 右: 反向投影图

在 MeanShift 算法的基础上, Bradski<sup>[22]</sup> 提出了一种对 MeanShift 算法在视频上的扩展, 即连续自适应均值漂移算法 (Continuously Adaptive MeanShift, CAMShift)。CAMShift 算法对视频的所有帧都进行 MeanShift 运算, 并将上一帧的结果作为下一帧的初始搜索窗口。相对于 MeanShift, 它在所有帧上都重新

按照初始窗口计算反向投影图，此外，在每一帧上，当 MeanShift 算法收敛后 CAMShift 算法都会根据目前搜索窗口中的像素概率值之和更新搜索窗口的大小，这样以来就可以一定程度上解决目标在尺度上的变化、形变和部分遮挡。

### 2.2.5 相关滤波

相关滤波法，就是根据当前帧的信息和之前帧的信息训练出一个相关滤波器，然后与新输入的帧进行相关性计算，得到的置信图便是预测的输入帧中物体的位置概率，得分最高的点或块就是最可能的跟踪结果。

KCF 算法 (Kernelized Correlation Filters) 利用了傅里叶域中，两个矩阵的卷积可以被转换为逐元素的点乘的性质，在达到与以往的更加复杂的算法的效果的基础上，降低了计算所需要的资源和时间。它是目前相关滤波算法中，效果最佳，也最常用的算法之一，Henriques et al.<sup>[23]</sup> 在 2015 年提出。

KCF 算法中实验了三种核函数，包括线性核函数、多项式核函数和高斯核函数。都使用较少的计算得出了较好的预测结果。它通过使用输入向量生成循环矩阵，一方面通过位移增加了样本的个数，一方面利用了循环矩阵可被离散傅里叶变换 (DFT) 对角化的特性，将岭回归中的求逆运算、矩阵乘法等转换为了计算量小的矩阵间逐元素计算，从而大大提高运行速度。

由于 KCF 使用傅里叶变换，导致滤波器尺寸和图像块尺寸必须一致，这样就限制了检测的尺度，导致 KCF 算法对尺度变化的适应性不强，当物体在图像中的尺度发生变化时，会导致 KCF 算法不能正常地对物体进行追踪。此外，KCF 使用循环矩阵虽然可以增加分类器的样本数量，但也导致了有些训练数据并不真实。为了解决这些问题，Lukezic et al.<sup>[24]</sup> 在 2017 年提出了通道和空间稳定的判别性相关滤波器 (Discriminative correlation filter with channel and spatial reliability, CSR-DCF)。根据实验，它有着比 KCF 算法更高的精度，相对的，帧率较 KCF 更低。

CSR-DCF 算法提出了空间置信度和通道置信度。空间置信度图是由反向投影图和先验概率得到每个像素点是目标的后验概率的在图中的分布图，反向投影图如均值漂移法中所述，由颜色直方图得到，先验概率由像素离目标所在位置中心的距离决定，使用 Epanechnikov 核函数进行计算。后验概率二值化后便可得到目标在图中所在区域。



## 2.2.6 GOTURN

GOTURN 是一个基于深度神经网络的跟踪算法<sup>[25]</sup>，在深度学习的跟踪算法中，GOTURN 由于其达到 100FPS 的帧率脱颖而出，它对于视角的变化、光照、变形等具有鲁棒性，但不能很好地处理遮挡。

GOTURN 网络以视频当前帧和上一帧作为输入，输出当前帧的目标所在区域的 bounding box 的位置。GOTURN 的网络结果如图2.10所示。其中的卷积层（Conv Layers）用于抽取图像特征，全连接层（Fully-Connected Layers）用于特征比较，找出当前帧上的目标位置。上一帧的剪切（crop）已知，当前帧的剪切是以上一帧的跟踪目标为中心，截取两倍目标大小的区域作为搜索区域，在该区域内进行回归。

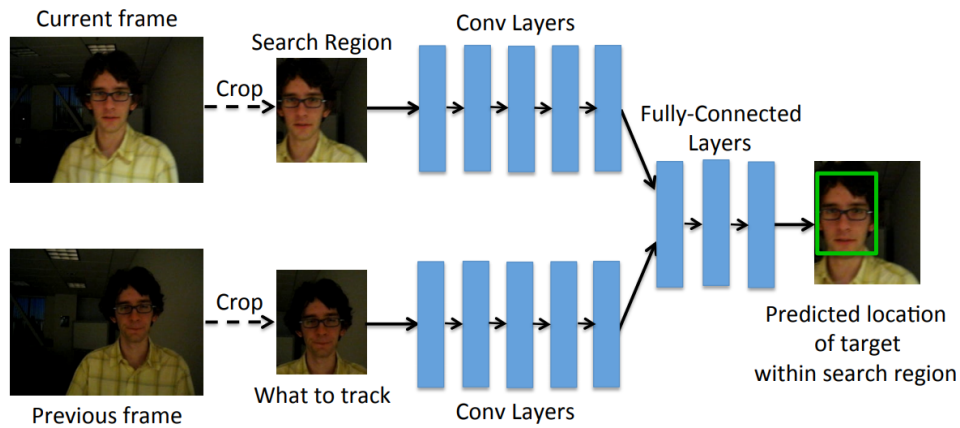


图 2.10 GOTURN 神经网络结构

## 2.2.7 TLD

TLD，即 Tracking-Learning-Detection 是一种单目标长时间的目标追踪算法，TLD 算法将长期追踪的任务分解成三个：跟踪、学习和检测。追踪模块在每个帧上不停地追踪对象的位置；探测模块在适当是时候对追踪其进行校正；TLD 中还提出了 P-N 学习模块来识别检测器的错误并更新检测器，<sup>[26]</sup>。

TLD 中使用 Median-Flow 作为追踪器，使用窗口扫描法和级联分类器作为检测器。级联分类器分为三个阶段：图像块方差（patch variance）、集合分类器（ensemble classifier）和最近邻（nearest neighbor）。

图像方差分类器计算目标物所在图像块的灰度值方差，并丢弃所有灰度值方差少于其 50% 的待测图像块。50% 这一阈值可以根据具体应用调整，它限制了目标的最大外观变化。通过图像方差分类器的图像块再经过集合分类器，它由



$n$  个基本分类器组成，每个集合分类器  $i$  对图像块进行像素比较，并得到一个二进制码  $x$ ，由  $x$  得到一个后验概率  $P_i(y|x)$ ，其中  $y \in \{0, 1\}$ 。将每个基本分类器的后验概率进行平均，并将  $y = 1$  的后验概率小于 50% 的图像块舍去。最后一级分类器是最近邻分类。

对一帧的探测和追踪都完成后，TLD 取追踪器得到的目标所在边界框 (bounding box) 和探测器得到的边界框中置信最大的结果作为最终估计，当二者都没有得到边界框，则认为目标在这一帧中不可见。由于 TLD 不是一直采用追踪器的结果，所以边界框在帧之间可能会发生跳动。

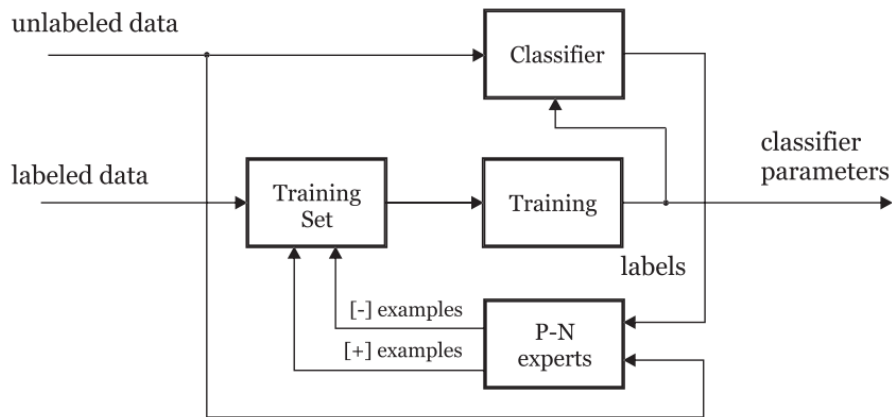


图 2.11 TLD 学习模块结构

学习模块是 TLD 算法的新颖之处，它在第一帧进行初始化，并在运行时更新探测器以改善其性能。学习模块分为四个部分：待训练的分类器、训练数据集、监督学习、P-N 专家，如图 2.11 所示。P-N 专家 (P-N experts) 用于在运行时生成训练所需的正样本和负样本并识别检测器的错误，P 专家识别检测器的漏报 (false negatives)，并将其添加到具有正标签的训练集中，N 专家识别被检测器预测为正的负样本 (false positives)，并将其添加到负标签的训练集中。此外，P 专家还用于增加正样本的数量，当获得一个含有目标的界限框后，P 专家将其通过几何变换生成若干个仿射的界限框，如将其进行  $\pm 1\%$  的偏移， $\pm 1\%$  的尺度变换， $\pm 10^\circ$  的平面内旋转等操作，这样可以通过增加正样本的数量来提高分类器的鲁棒性。N 专家则通过在图像的界限框之外的区域取若干个图像块作为负样本。

为了判断检测器的错误，P 专家利用视频中的时间结果，记录目标的轨迹并假设目标延轨迹移动，由此预测当前帧中的目标物位置。如果检测器将当前位置标记为负，则认为是漏报，由 P 专家将其标记为正；N 专家利用视频中的空间结构，并假设目标物在一帧中只能出现在一个位置，它分析当前帧中检测器和跟踪

器产生的所有结果，选择置信度最高的结果，并将与所选界限框不重叠的图像块标记为负。需要注意的是，P-N 专家所判断的错误并不总是正确的，它们的假设都有失效的情况，但尽管误差存在，P-N 学习模块仍能够改善检测器的性能，即这种误差在一定条件下是允许的。

## 第 3 章 ROS 导航和可佳导航简介

### 3.1 ROS 导航

ROS (Robot Operating System) 是一个开源的专用于机器人软件开发的操作系统。ROS 中提供了一个模块化的简单 2D 导航系统 ROS Navigation，其主要架构如图 3.1 所示。在已有的导航结构的基础上，我们可以以插件的形式添加所需的行人识别和追踪模块，并根据此信息和用户的要求进行行人跟随。

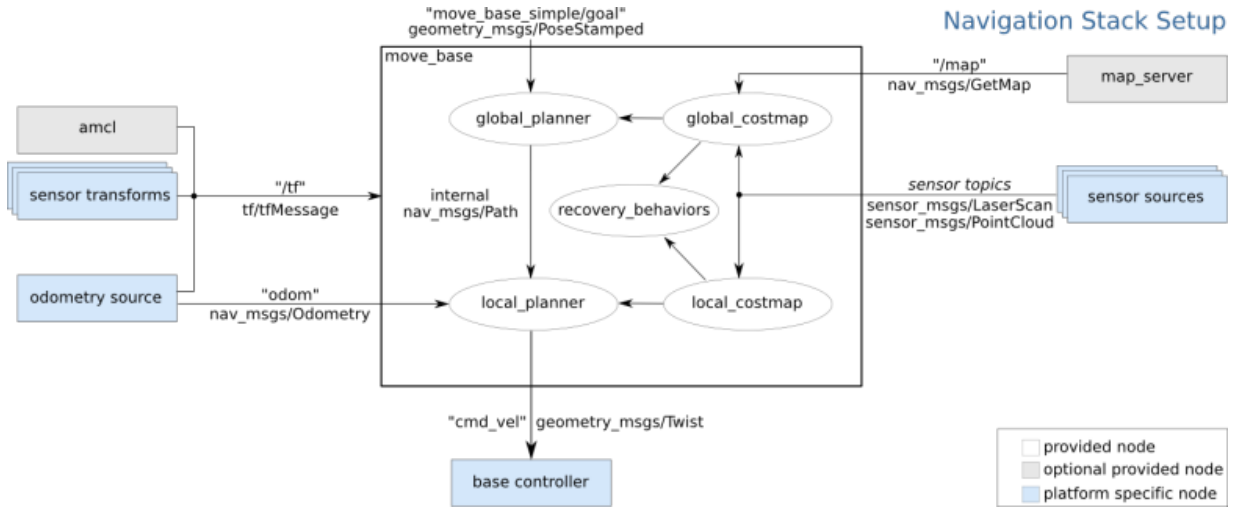


图 3.1 ROS 导航架构

#### 3.1.1 坐标系转换

在机器人系统中存在数个坐标系，包括机器人底座为中心的坐标系，2D 激光传感器为中心的坐标系，摄像机为中心的坐标系，全局地图上的坐标系等。为了能使各个传感器得到的不同坐标系下的信息方便准确地结合使用，ROS 导航提供了 tf 系统，由用户指定不同坐标系之间的 tf 变换，并存储起来，维护一个 tf 变换树。在调用 tf 变换时，只需要指定原坐标系的结点和需要变换的目标坐标系的节点，tf 系统就会自动计算出两个坐标系之间的相对位置和角度，并完成坐标的转换。

#### 3.1.2 里程计

机器人的里程计包括机器人的位姿 (pose) 和速度 (twist)，其中，机器人的位姿和其姿势，可以由机器人的初始位置和机器人的控制单元的速度，通过运

动模型计算得到的，也可以通过激光扫描数据对机器人位置的定位得到。里程计信息并被发布给局部规划器，用于路径规划。

### 3.1.3 建图

对于有里程计和固定水平激光测距仪的机器人，在地面平坦的情况下，可以使用 gmapping<sup>[27]</sup> 的方式进行建图。此外还可以选择 cartographer<sup>[28]</sup> 方法，在没有里程计，激光测距仪不是完全水平的情况下，如使用者手持激光测距仪的情况下也可以进行建图。建图是实时的，即把当前激光扫描到的物体根据当前的定位加入地图中，所以在室内环境下时，在使用前首先操作机器人将室内环境扫描一遍，即可建立室内的地图。地图使用一个描述地图元数据的 YAML 文件和一个编码了图中占用/自由点的 image 文件存储起来，ROS 导航中提供一个 map\_server 节点用于发布地图数据。

由于地图是静态存储的，而环境通常是动态的，为了应对机器人实际运行中可能遇到的意想不到的动态障碍，还需要维护代价地图（costmap）。代价地图采用传感器数据和来自静态地图的信息，以一定的频率进行更新，来存储和掌握实际环境中的障碍物信息。在 ROS 导航中，维护两个代价地图，分别用于在整个环境上的全局和长期规划，以及局部区域内的规划和避障。

### 3.1.4 定位

自适应蒙特卡罗定位（adaptive Monte Carlo localization, amcl）模块是机器人的定位模块，在建立地图后使用，使用粒子滤波算法进行对机器人当前位置的估计。通过在图上均匀地撒上一些点，再随着机器人在图中的运动，计算每个点的机器人当前位置的概率，减少概率小的位置点的密度，增加概率大的位置点的密度，在粒子不断收敛后即可较为准确地得到机器人在图中的位置。

### 3.1.5 导航控制

导航控制模块即图3.1中的 move\_base 模块，包括全局规划器、局部规划器和恢复机制。全局导航支持 A\* 算法和 dijkstra 算法来在全局代价地图上找到前往下一个目的地的最短路径，在机器人开始移动之前就首先被计算出来。局部规划器监控了传感器信息，结合了里程计信息、全局和局部代价地图来选择机器人在全局路径的局部分块中应选择的最佳速度（线速度和角速度），传送给 base\_control 模块。同时，局部规划器也可以动态地重新规划机器人的局部路径

以进行避障，使用动态窗口法（Dynamic Window）<sup>[29]</sup> 进行局部避障，使用路径展开法（Trajectory Rollout）<sup>[30]</sup> 进行局部规划和控制。恢复机制用于出现了异常情况，机器人无法进行决策时使用，ROS 导航提供了两种恢复行为，使用静态地图在用户指定的更大范围外恢复代价地图，或通过使机器人 360° 旋转来尝试清出空间。

### 3.2 可佳导航

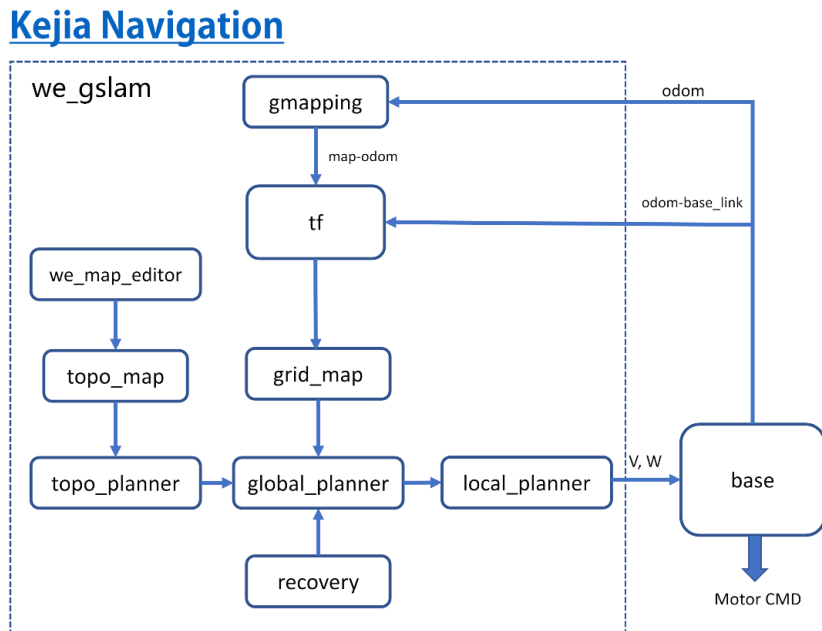


图 3.2 可佳导航架构

可佳导航的架构如图3.2所示，它同样使用 ROS 系统用于软件的运行和各模块之间的通信。在导航方面，它相对于 ROS 导航，还加入了拓扑导航和语义地图。通过手动注释如房间、门、家具等对象的大致位置和区域，自动生成一个拓扑地图，该拓扑地图会用于全局路径规划和定位。

可佳导航使用 A\* 算法进行全局和局部的路径规划，并使用向量场直方图（vector field histogram, VFH）<sup>[31]</sup> 进行局部避障。可佳导航使用了一个状态机来进行任务的规划和决策。在收到任务目标后，它首先在全局地图上运行 A\* 算法，并将全局路径分成一些局部目的地（waypoints），将最近的 waypoint 作为 VFH 算法的目标点，运行局部的避障。如果遇到了障碍物，将障碍物更新到本地代价地图中，重新运行 A\* 的局部路径规划，如果此处的局部路径规划失败，则运行恢复机制<sup>[32]</sup>。

## 第 4 章 以可佳机器人为基础的行人追踪系统

### 4.1 输入设备

#### 4.1.1 Kinect

Kinect 深度相机可以同时提供 RGB 图像和深度图像。

使用 RGB-D 相机 Kinect 作为彩色图像和深度信息的输入，而 Kinect 的深度图像只能有效判断 80 厘米之外的物体的距离，且由于 Kinect 通过红外发射器和红外摄像头来获得深度信息，所以受阳光照射的影响较大，所以在实际使用中，Kinect 深度图像可能会出现不稳定和空洞现象。

对于 Kinect 的深度图像在近距离精度较低的现象，考虑在较近距离内使用 2D 激光进行行人的 3D 位置判别。对于 Kinect 的空洞现象，考虑使用 KinectFusion 算法<sup>[33]</sup>或 OctoMap<sup>[34]</sup>，将深度图像投影到 RGB 图像中，以进行对相机视野中场景的 3D 重建。

#### 4.1.2 2D 激光

由于本文中的行人检测与追踪算法主要使用了视觉信息，所以这里 2D 激光主要用于建图、定位和导航。定位和导航在本文中不是主要内容，所以不再特别赘述。

### 4.2 视觉追踪系统

#### 4.2.1 总体架构

#### 4.2.2 目标人物注册

##### 1. 使用 OpenPose 系统找到目标人物

追踪系统在初始化时需要得到目标行人所在的大致位置，即提供行人在初始帧中的界限框（bounding box），并由此对追踪器和分类器进行初始化。为了能够有效地对目标用户进行识别，这里采用开源系统 OpenPose<sup>[35]</sup>来进行人体的骨骼识别。OpenPose 系统不仅可以判断出图像中所有人物的骨骼信息，还可以对人物的姿势进行识别，这带来了另一个好处，即我们可以要求被跟随的目标站在机器人前方做出一个指定的姿势，如举起右手，直到机器人提示注册成功，这样

一来,我们就可以在对目标人物的特征没有先验知识的情况下在初始化时发现他/她了。

## 2. 初始化分类器

根据第一帧由 OpenPose 识别出的目标人物所在区域作为正样本,随机提取背景作为负样本,通过拟合出一个分类器。该判别器的作用是在追踪算法失败后负责进行全局搜索和恢复。

根据之前所调研的结果,这里采用 HSV 直方图和 HOG 特征结合,作为图像的特征向量和分类器的输入。由于对人物的先验知识较少,且人物可能在追踪过程中发生姿态变化等影响检测算法准确度的形变,采用 Online AdaBoost 算法,以行人在每帧的外观作为样本不断进行学习,更新分类器。

### 4.2.3 行人追踪

#### 1. 行人追踪算法的选择

只测试了 KCF、CSR-DCF、TLD、MedianFlow 和 MOSSE 算法。在用于行人追踪时,帧率上, MOSSE 算法以约 450FPS 的帧率遥遥领先,其次是 270FPS 的 MedianFlow, 40FPS 的 KCF 算法, TLD 和 CSR-DCF 算法的帧率较低,在 15 到 20FPS 之间,但也勉强可以达到实时水平。在追踪准确度上, KCF 算法在尺度不变时准确度较高,能在一定程度上解决目标被遮挡的问题,且有可能在失败后再次恢复,但前面已经讨论过, KCF 算法无法适应目标尺度大小的变化; TLD 算法识别准确率较低,边界框会经常跳动或漂移,甚至会经常检测到错误的位置,但在目标在视野中消失一段时间后再回到画面时,只有 TLD 算法能识别到目标的回归并重新建立起追踪; CSR-DCF 算法识别准确度较高,能够稳定地检测目标的位置,且与 KCF 算法相比尺度可变,但当检测失败后, CSR-DCF 算法几乎无法恢复; MOSSE 算法同样是尺度不变的追踪,对于遮挡的情况不鲁棒,追踪不准确,且对于错误的追踪无法识别和恢复; MedianFlow 算法准确度相对也不高,在失败后无法恢复。

综合这些追踪器的表现,主要考虑使用 CSR-DCF 算法,但同时,针对它难以在失败后恢复的缺点,需要额外判断它在追踪时出现的错误和失败后的恢复。

#### 2. 更新分类器

在每一帧都会由追踪器得到目标所在的图像块,由分类器给出一个它符合行人特征的得分,当得分超过阈值  $threshold_1$  时,认为它是完全准确的追踪,并将其作为正样本加入分类器进行训练;当得分低于阈值  $threshold_2$  时,认为这个

追踪是失败的，使用分类器对图像进行扫描来搜索目标，搜索到，即用新的边界框重新开始追踪，若没有搜索到，即认为目标搜索失败。

#### 4.2.4 目标丢失恢复

追踪失败有几种可能：（1）目标从视野的左右两侧消失；（2）目标被完全部分或完全遮挡；（3）目标进入另外的房间。

对于可能（1）根据目标消失的位置及时转动机器人的视角，如果追踪器还是没能恢复追踪，则使用分类器在每一帧上对目标进行搜索，直到找到目标人物。对于可能性（2）（3），则使机器人尽快前往目标出现的上一个位置，并将视野转向行人可能所在的地点，用分类器进行目标搜索。



## 参 考 文 献

- [1] Treptow A, Cielniak G, Duckett T. Real-time people tracking for mobile robots using thermal vision. *Robotics and Autonomous Systems*, 2006, 54(9):729-739.
- [2] Zhou H, Taj M, Cavallaro A. Target detection and tracking with heterogeneous sensors. *IEEE Journal of Selected Topics in Signal Processing*, 2008, 2(4):503-513.
- [3] Susperregi L, Martínez-Otzeta J M, Ansuategui A, et al. Rgb-d, laser and thermal sensor fusion for people following in a mobile robot. *International Journal of Advanced Robotic Systems*, 2013, 10(6):271.
- [4] Arras K O, Grzonka S, Lubner M, et al. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. *2008 IEEE International Conference on Robotics and Automation*, 2008:1710-1715.
- [5] Kleinhagenbrock M, Lang S, Fritsch J, et al. Person tracking with a mobile robot based on multi-modal anchoring. *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, 2002:423-429.
- [6] Bellotto N, Hu H. Multisensor-based human detection and tracking for mobile service robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008, 39(1):167-181.
- [7] Wong C, Kortenkamp D, Speich M. A mobile robot that recognizes people// *Proceedings of 7th IEEE international conference on tools with artificial intelligence*. IEEE, 1995: 346-353.
- [8] Bellotto N, Hu H. Multimodal people tracking and identification for service robots. *International Journal of Information Acquisition*, 2008, 5(03):209-221.
- [9] Noceti N, Destro A, Lovato A, et al. Combined motion and appearance models for robust object tracking in real-time. *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009:412-417.
- [10] Mowbray S D, Nixon M S. Automatic gait recognition via fourier descriptors of deformable objects// *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, 2003: 566-573.
- [11] Koide K, Miura J. Identification of a specific person using color, height, and gait

- features for a person following robot. *Robotics and Autonomous Systems*, 2016, 84:76-87.
- [12] Sural S, Qian G, Pramanik S. Segmentation and histogram generation using the hsv color space for image retrieval. *Proceedings. International Conference on Image Processing*, 2002, 2:II-II.
- [13] Foley J D, Van Dam A, et al. Fundamentals of interactive computer graphics: volume 2. Addison-Wesley Reading, MA, 1982.
- [14] Ojala T, Pietikainen M, Harwood D. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *Proceedings of 12th International Conference on Pattern Recognition*, 1994, 1:582-585.
- [15] Ojala T, Pietikäinen M, Mäenpää T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2002(7):971-987.
- [16] Dalal N, Triggs B. Histograms of oriented gradients for human detection. *international Conference on computer vision & Pattern Recognition (CVPR'05)*, 2005, 1:886-893.
- [17] Lowe D G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 2004, 60(2):91-110.
- [18] Zhou H, Yuan Y, Shi C. Object tracking using sift features and mean shift. *Computer vision and image understanding*, 2009, 113(3):345-352.
- [19] Babenko B, Yang M H, Belongie S. Visual tracking with online multiple instance learning. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009:983-990.
- [20] Welch G, Bishop G, et al. An introduction to the kalman filter. 1995.
- [21] Arulampalam M S, Maskell S, Gordon N, et al. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 2002, 50(2):174-188.
- [22] Bradski G R. Computer vision face tracking for use in a perceptual user interface. 1998.
- [23] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 2015, 37(3):583-596.

- [24] Lukezic A, Vojir T, ˇCehovin Zajc L, et al. Discriminative correlation filter with channel and spatial reliability. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017:6309-6318.
- [25] Held D, Thrun S, Savarese S. Learning to track at 100 fps with deep regression networks. *European Conference Computer Vision (ECCV)*, 2016.
- [26] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 2012, 34(7):1409-1422.
- [27] Grisetti G, Stachniss C, Burgard W, et al. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 2007, 23(1):34.
- [28] Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2d lidar slam. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016:1271-1278.
- [29] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 1997, 4(1):23-33.
- [30] Gerkey B P, Konolige K. Planning and control in unstructured terrain. *ICRA Workshop on Path Planning on Costmaps*, 2008.
- [31] Borenstein J, Koren Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE transactions on robotics and automation*, 1991, 7(3):278-288.
- [32] Liu J, Zhang Z, Tang B, et al. Wrighteagle@ home 2017 team description paper. *RoboCup@ Home*, 2017.
- [33] Newcombe R A, Izadi S, Hilliges O, et al. Kinectfusion: Real-time dense surface mapping and tracking. *ISMAR*, 2011, 11(2011):127-136.
- [34] Hornung A, Wurm K M, Bennewitz M, et al. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 2013, 34(3):189-206.
- [35] Cao Z, Hidalgo G, Simon T, et al. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008*, 2018.