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： 二〇一九年五月十四日

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 14, 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 · · · · · 6
1.2.1 基于视觉信息的行人检测与追踪 · · · · · 6
1.2.2 基于激光的行人检测与追踪 · · · · · 7

第 2 章 行人追踪算法 · · · · · 8
2.1 视觉行人追踪 · · · · · 8
2.1.1 行人检测 · · · · · 8
2.1.2 基于动态的追踪 · · · · · 17
2.2 激光行人追踪 · · · · · 26

第 3 章 ROS 导航和可佳导航简介 · · · · · 27
3.1 ROS 导航 · · · · · 27
3.1.1 坐标系转换 · · · · · 27
3.1.2 里程计 · · · · · 27
3.1.3 建图 · · · · · 28
3.1.4 定位 · · · · · 28
3.1.5 导航控制 · · · · · 28
3.2 可佳导航 · · · · · 29

第 4 章 以可佳机器人为基础的行人追踪系统 · · · · · 30
4.1 输入设备 · · · · · 30
4.1.1 Kinect · · · · · 30
4.1.2 2D 激光 · · · · · 30
4.2 视觉追踪系统 · · · · · 30
4.2.1 总体架构 · · · · · 30
4.2.2 目标人物注册 · · · · · 30

4.2.3 行人追踪 · · · · · 31
4.2.4 目标丢失恢复 · · · · · 31

参考文献 · · · · · 32
附录 A 补充材料 · · · · · 34

中文内容摘要

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健康的服务工作，但不包括从事生产的设备。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所适用的对象。

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研究：

（1）常用目标跟随算法的原理与实现；

（2）ROS 导航和可佳导航介绍；

（3）可佳机器人上行人跟随系统的实现。

关键词：计算机视觉；机器人；目标追踪；路径规划

Abstract

A service robot is a robot which operates semi- or fully autonomously to perform services useful to the well-being of humans and equipment, they exclude manufacturing operations, and they are capable of making decisions and acting autonomously in real and unpredictable environments to accomplish determined tasks. There are two types of service robots, personal/domestic service robots and professional robots. Professional robots are typically used in specific occasions, including business, delivery, medical, rescue, etc. Personal/domestic service robots, which include cleaning robots, elder care and medical companions, entertainment and leisure robots, home education and training robots, are the specific research objects in this paper.

In order to accurately understand the current environment and effectively execute the instructions, one of the important and necessary ability for a personal service robot, is to automatically recognize and track a person precisely and robustly.

This paper is consist of three parts:

(1) The theories and implementation of various existing object tracking algorithms;

(2) Introduction of ROS navigation and kejia navigation;

(3) The implementation of the people following system on kejia robot.

Key Words: Computer Vision; Robotics; Object Tracking; Path Planning

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健康的服务工作，但不包括从事生产的设备。它们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所适用的对象。

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，是移动服务机器人的人机交互中的一项重要且必要的功能。

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。移动机器人的智能指标包括三个方面：自主型、适应性和交互性。

国际机器人联合会 (International Federation of Robotics) 对服务机器人做了如下定义：

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和设备的服务工作，但不包括从事生产的设备。

服务机器人通常也是移动机器人。

从 20 世纪 80 年代中期开始，机器人已从工厂的结构化环境进入人的日常生活环境——医院、办公室、家庭和其他杂乱及不可控环境，成为不仅能自主完成工作，而且能与人共同协作完成任务或在人的指导下完成任务的智能服务机器人。

20 世纪 90 年代末，世界服务机器人协会 (International Service Robot Association) 才第一次定义了服务机器人的概念：能够进行感知、思考和行动，并以此来有益于和扩展人类的能力和提高人类的生产效率的机器。

服务机器人被看重的就是交互能力

随着人工智能与物联网技术不断发展，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。在技术层面，我国服务机器人与国外

相比，仍存在较大差距比如在机器人基础算法、核心软件、人工智能硬件落地等方面存在短板。国内服务机器人总体尚处于初级发展阶段，半数以上的产品处于研发试验阶段，但其增长速度较快。根据相关数据显示，2017 年全年我国的整体机器人规模市场达到 1200 亿元，其中服务机器人占据 28% 的市场，服务机器人增长率明显高于工业机器人的发展。

现今，随着物联网的发展和人们对智能化的要求，在日常生活中协助或娱乐人类的个人服务机器人市场正在迅速发展。2017 年，个人/家庭服务机器人市场价值增长了 27%，达到 21 亿美元；总数增加了 25%，达到约 850 万台。据估计，近 610 万台机器人被用于家庭工作。

个人和国内应用中的机器人技术经历了强劲的全球增长，地板清洁机器人，机器人和用于家庭教育的机器人（后者越来越多地被称为社交机器人）越来越成为人们生活的一部分。未来的产品愿景指向具有更高复杂性，能力和价值的家用机器人，例如用于支持老年人的辅助机器人，帮助做家务和娱乐。

1. 随着人口老龄化趋势的加重，服务机器人市场迎来了爆发增长期。家庭用机器人，智能公共服务机器人应用场景和服务模式不断拓展。随着人类寿命的演唱，老龄化趋势的加重，给医疗健康机构带来越来越大的压力，养老问题兔鳄家明显，社会对老年人护理的需求大大增加。智能养老设备，如智能服务机器人，的出现极大的弥补了由老年人口激增，护工、养老机构等养老资源匮乏所带来的养老服务供需缺口。此处可引用一段华为项目比赛的简介？）

2. 行人跟随是智能服务机器人的人机交互中的一项重要技术。它要求机器人能够准确识别指定目标，通过对目标的跟随来保证更好地完成人机交互，同时，在移动过程中强调安全。

1.2 相关工作

Literature 中已经有很多 following 相关的研究工作。常用的行人追踪方法分为基于视觉信息的行人检追踪、基于激光信息的行人追踪，以及多传感器融合的方法。

1.2.1 基于视觉信息的行人检测与追踪

基于检测的追踪和基于追踪的算法:https://zhuanlan.zhihu.com/p/32826719

大部分方法使用了粒子滤波 (particle filters)，多假设追踪 (multiple hypothesis

tracking)，卡尔曼滤波 (Kalman filters) 的方法

[1]

1. 基于人脸识别的跟踪

(2) 人脸识别的速度和正确率均已达到一个很高的层次，但在实际的激动机器人跟随场景中，人不是一直面对移动机器人。

(3) 基于模板匹配的跟踪

(4) 基于轮廓信息的跟踪

1.2.2 基于激光的行人检测与追踪

(1) 使用几何特征识别目标

(2) 基于运动检测识别目标

第 2 章　行人追踪算法

2.1　视觉行人追踪

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。

在密集跟踪方法中，我们实际上并没有“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，所以以这种方法进行目标跟踪，计算量会比较大。

由于目标的运动通常是连续的，稀疏跟踪方法即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，得出其当前位置的算法。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，此外，当物体被暂时遮挡时，目标检测在此时很大可能下会直接失败，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。但在物体速度较快时，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。

下面主要介绍在每帧上的行人检测，和以此为基础的动态行人追踪。

2.1.1　行人检测

经典的基于检测的追踪方法包括提取人工特征，在使用分类器进行分类，分类器包括支持向量机（Support Vector Machines, SVM），随机森林分类器（Random Forest）和各种 Boosting 算法（如 AdaBoost）。

1. 常用特征描述子

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。

（1）颜色直方图

颜色特征具有旋转不变性，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，但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。颜色直方图具有稳定性好、抗部分遮挡、计算方法简单和计算量小的特点。

颜色直方图可以基于不同的的颜色空间，其中，最常用的是 RGB 空间和 HSV 空间。以 RGB 空间为例，分别统计每个像素的 R、G、B 数值落在 [0,255] 上每个点的频度，绘制出直方图，该图片的颜色特征即为三个长为 256 的向量，分别表示其的红色、绿色、蓝色的统计分布。以一张人物照片2.1为例，图2.2为其 RGB 颜色直方图。

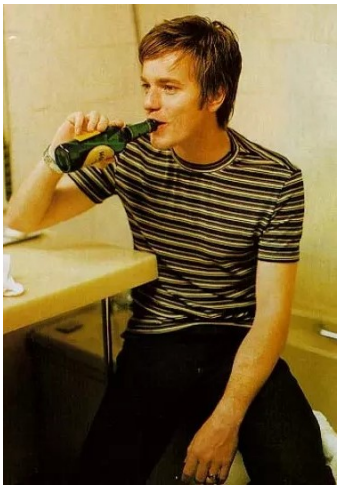

图 2.1　原图

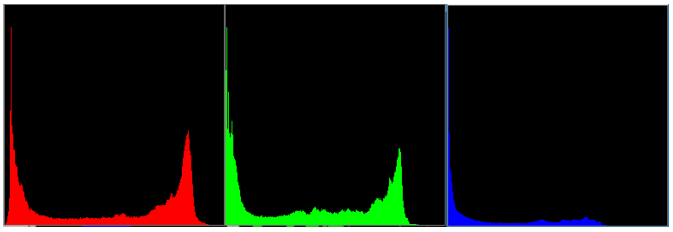

图 2.2　左：B 通道颜色直方图；中：G 通道颜色直方图；右：R 通道颜色直方图

但 RGB 颜色空间的均匀性非常差，且两种颜色之间的直觉差异色差不能表示为改颜色空间中两点间的距离，RGB 这三种颜色的分量的取值与所生成的颜色之间的联系并不直观。

在计算机视觉中，我们常采用 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 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.[2] 的实验显示，使用 HSV 直方图进行行人识别的结果相比 RGB 直方图有了明显提高。

HSV 即色相（Hue）、饱和度（Saturation）、亮度（Value）。色相即表示物体的颜色，如红色、黄色等，在 0° 到 360° 的标准色轮上，按位置度量色相；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；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。

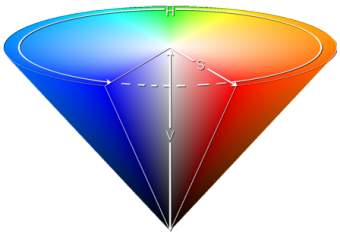

图 2.3　HSV 模型可以使用圆柱坐标系中的一个圆锥形子集表示

由于大部分数字图像都是基于 RGB 空间进行表示的，我们需要首先把 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:[3]

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

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

$$v = max$$

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。

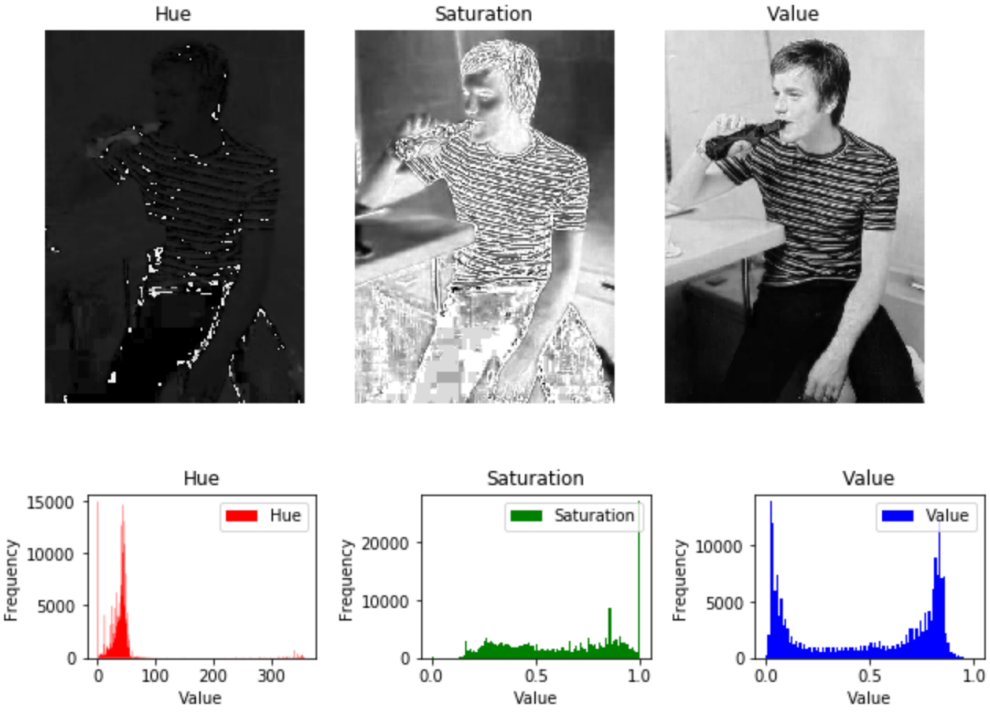

图 2.4　左：H 通道灰度图和颜色直方图；中：S 通道灰度图和颜色直方图；右：V 通道灰度图和颜色直方图

（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.[4] 在 1994 年首次提出。

LBP 的计算方法非常简单。每个像素都根据它相邻的八个像素按规定的顺序（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，如图2.5所示，将每个点的 LBP 值使用灰度图表示，得到 LBP 图谱如2.6为例。

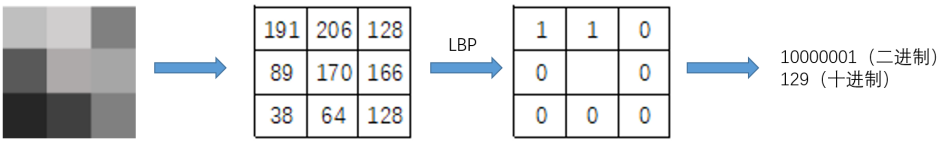

图 2.5　计算 3x3 像素块中中心点的 LBP 值

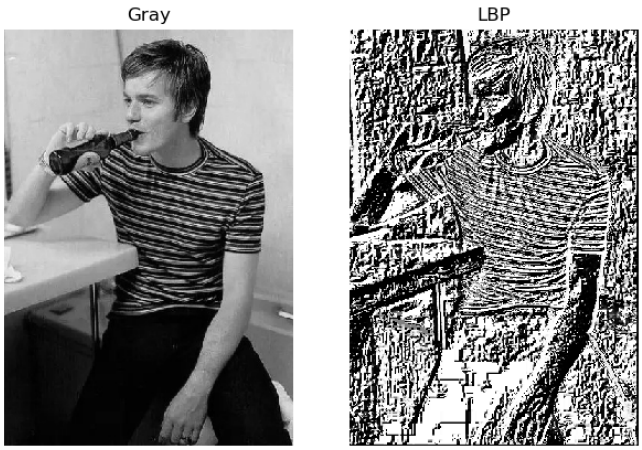

图 2.6　左：灰度图；右：由灰度图计算得到的 LBP 图谱

为了使得 LBP 描述子有旋转不变性，Ojala et al.[5] 提出了一个 LBP 的具有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 16×16 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.[6] 在 2005 年提出 HOG 结合 SVM（支持向量机，support vector machine）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 特征描述符中，它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，故在这些区域，梯度的大小会很大，显然，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 特征描述字对噪点比较敏感，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内核（Kernel）$[-1, 0, 1]$ 和 $[-1, 0, 1]^T$ 对原图进行过滤，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 算子等作为内核，不过根据 Dalal et al.[6] 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：

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

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

梯度大小和方向分别为：

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

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

使用以上公式在 RGB 颜色空间上计算图2.1的梯度值，如图2.7所示。这张梯度图像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，且在同时突出了人物的轮廓。

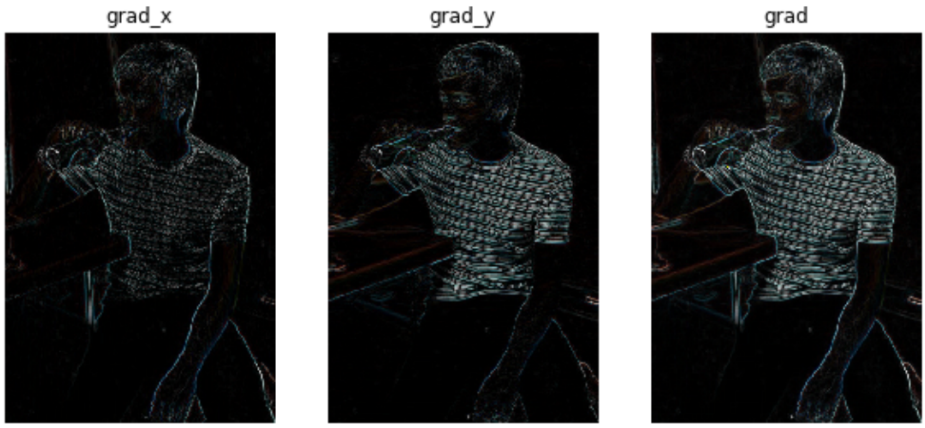

图 2.7　左：横向梯度绝对值；中：纵向梯度绝对值；右：梯度大小

图2.7中包括了 RGB 三个通道在每一个像素点上的梯度值，在计算 HOG 特征向量时，我们选取三个通道的梯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实际上是梯度的方向。梯度的方向在 $[0°, 360°)$ 上，但是我们实际在统计方向时，采用的却是 $[0°, 180°)$ 的统计范围，计算 $\theta(x, y)$ mod $180°$ 来代替原有的角度值，即将相差 $180°$ 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 $[0°, 360°)$ 范围的原方向更好[6]。在统计梯度方向时时，我们还需要使用梯度的大小作为对应方向的权重。

在计算直方图时，我们取 9 个组（bins），分别对应 $0°, 20°, 40°, \cdots, 160°$，若一个像素处的梯度正好为 20 的整数倍，将其梯度大小加到对应的 bin 中；否则，按照比例，将其加入相邻的两个 bins 中。以图2.8为例。这样一来，HOG 特征描述子即为一个长为 9 的向量，每一个分量的大小对应直方图中相应 bin 的高度。

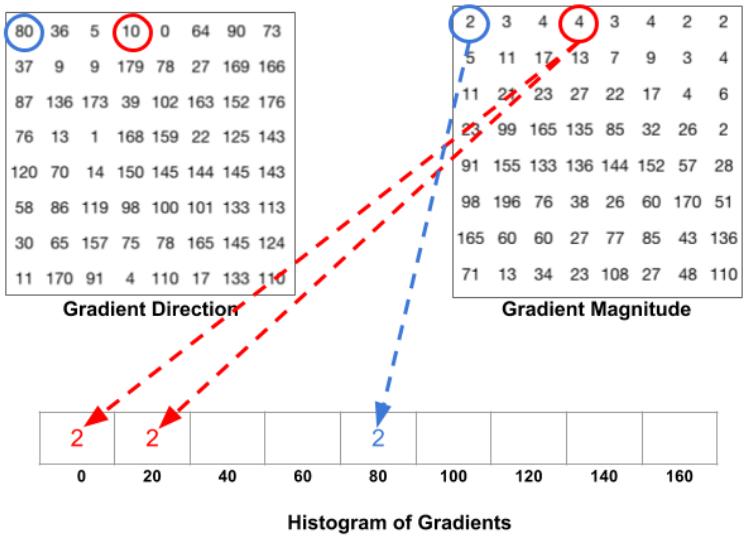

图 2.8　统计梯度方向直方图的方法示意

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，这时所有梯度大小就也会减半，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 8×8 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。此过程如图2.9所示。

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 16×16 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 $\boldsymbol{v} \leftarrow \boldsymbol{v}/\sqrt{\|v\|_2^2 + \epsilon^2}$, 其中 ϵ 为一个足够小的正常数。在规范化一个块之后，我们得到了一个长为 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

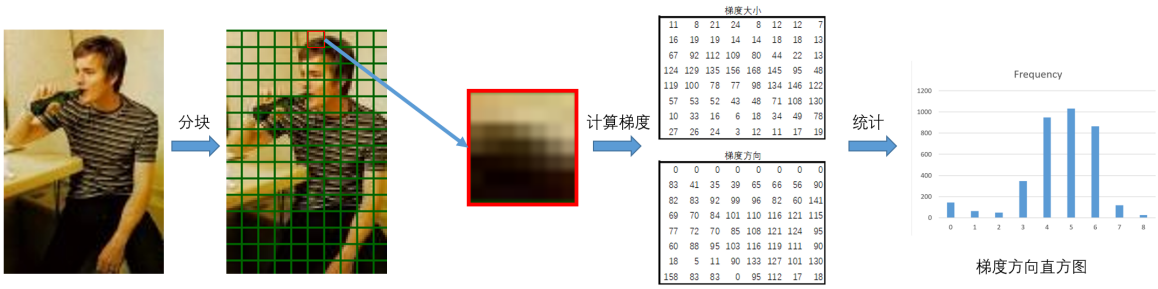

图 2.9　统计单元格梯度直方图的过程示意图

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，可以在所有的尺度和位置上密集地覆盖图像，例如，一个 500×500 像素的图像可以最多生成 2000 个稳定的特征。SIFT 方法中的关键点描述非常独特，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不会从数据库中得到正确的匹配，故而在正确的匹配之外，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[7]。

为了最大限度地降低提取特征的成本，使用级联过滤（cascade filtering）方法来检测关键点，先使用高效的算法来检测出一些侯选位置，然后再进一步详细检查，将更加耗时的计算只运用到通过了初始测试的候选点位置。

Lowe[7] 提出的 SIFT 生成图像特征的主要步骤如下：

I. 尺度空间极值检测：通过高斯差分方程（difference-of-Gaussian function），在所有可能的尺度下搜索稳定的特征，来找出尺度和方向不变的潜在相关点。对于输入图像 $I(x, y)$ 和可变尺度高斯核函数 $G(x, y, \sigma)$，可以计算出图像的尺度空间 $L(x, y, \sigma)$：

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

“$*$”为 x, y 上的卷积计算操作，$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$，$\sigma$ 为尺度空间因

子，反映了图像被模糊的程度，σ 越大，对应的尺度也越大。

为了有效地检测出尺度空间中稳定的关键点位置，使用高斯差分方程来与图像进行卷积，由一个常数因子 k，得出 $D(x,y,\sigma)$：

$$D(x,y,\sigma)=(G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y)=L(x,y,k\sigma)-L(x,y,\sigma)$$

为了检测中 $D(x,y,\sigma)$ 函数的局部最大值和最小值，每个点都与它在同尺度上的 8 个邻点，以及相邻的两个尺度上的各 9 个邻点相比较（相邻尺度上 $\sigma_{s+1}=k\sigma_s$）。只有它的值比相比较的 26 个相邻点都要小或者都要大时，才选取该点作为候选点。

II. 关键点精确定位：对每个候选点上，拟合一个精细的模型来确定其位置和尺度，并根据其稳定性来选择关键点。

对尺度空间函数 $D(x,y,\sigma)$ 进行最多二阶的泰勒展开：

$$D(\boldsymbol{x})=D+\frac{\partial D^T}{\partial \boldsymbol{x}}\boldsymbol{x}+\frac{1}{2}\boldsymbol{x}^T\frac{\partial^2 D}{\partial \boldsymbol{x}^2}\boldsymbol{x}$$

其中 $\boldsymbol{D}$ 和其微分在给定候选点处被计算，$\boldsymbol{x}=(x,y,\sigma)^T$ 是相对于该点的偏移向量。为了求 $\boldsymbol{D}$ 的局部极值的位置，对上式求导并设 $\boldsymbol{D}'$ 为 0，则极值的位置的偏移量 $\hat{\boldsymbol{x}}$ 和 $\boldsymbol{D}$ 的局部极值点分别为：

$$\hat{\boldsymbol{x}}=-\frac{\partial^2 D^{-1}}{\partial \boldsymbol{x}^2}\frac{\partial D}{\partial \boldsymbol{x}}$$

$$D(\hat{\boldsymbol{x}})=D+\frac{1}{2}\frac{\partial D^T}{\partial \boldsymbol{x}}\hat{\boldsymbol{x}}$$

为了保证极值点的稳定性，需要剔除低对比度的极值点。若 $|\boldsymbol{D}(\hat{\boldsymbol{x}})|$ 的值小于 0.03（假定每个像素的值的大小在 $[0,1]$ 之间），则将该极值点舍弃。

III. 方向赋值：根据局部的图像梯度方向，将一个或多个方向赋值给每个关键点位置。后续在该图像上进行的所有操作都将根据为每个特征所指定的方向、尺度和位置进行变换（transform），从而为这些特征提供方向、尺度不变性。

得到了每个关键点的尺度 σ 后，由特征点为中心，计算出同一尺度下周围区域每个点 (x,y) 的梯度大小 $m(x,y)$ 和方向 $\theta(x,y)$：

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}$$

$$\theta(x,y)=\tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y)))$$

使用直方图统计关键点邻域内像素对应的梯度方向和大小。

IV. 生成特征描述子：校正旋转门主方向以确保旋转不变性，生成关键点描述子并进行归一化处理，以去除光照的影响。

SIFT 准确率较高，对于尺度和旋转不敏感，但由于求 SIFT 特征向量计算量较大，耗时较长，所以在需要实时图像识别时并不常被使用。

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，分类器即会输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，选取不同的尺度，尺度决定了检测窗口的大小，然后在图像上选取相应尺度的检测窗口，计算它的特征向量，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。在检测完一个追踪窗口后，按照给定的步长在图像上滑动检测窗口，继续进行检测，直到整个图片在所有给定尺度上都被检测过。

（1）SVM

支持向量机（Support Vector Machine，SVM）是一个监督学习的模型。为了减少计算量，在进行行人识别时，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。

（2）AdaBoost

由于 SVM 计算量较大，为了提高速度，也可以使用 AdaBoost 代替。由于 AdaBoost 使用多级分类器级联的方式，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。

2.1.2 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。

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.[8] 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. 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[9]。

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：

$$x_{k+1} = \boldsymbol{A}_k x_k + \boldsymbol{B} u_k + w_k$$

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

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

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：

$$p(w) \sim N(0, \boldsymbol{Q}),$$
$$p(v) \sim N(0, \boldsymbol{R}).$$

$n \times n$ 的矩阵 $\boldsymbol{A}$ 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 $\boldsymbol{B}$ 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 $\boldsymbol{H}$ 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：

$$\boldsymbol{P}_k^- = \boldsymbol{E}[e_k^- e_k^{-T}],$$
$$\boldsymbol{P}_k = \boldsymbol{E}[e_k {e_k}^T]$$

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 $\hat{x}_k$：

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

在上式中，$\boldsymbol{H}_k \hat{x}_k^-$ 是根据先验估计对测量值的预测，$(z_k - \boldsymbol{H}_k \hat{x}_k^-)$ 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 $\boldsymbol{K}$ 是最小化后验误差协方差的增益（gain）。将上式代入求 $\boldsymbol{P}_k$ 的公式中，取结果相对于 $\boldsymbol{K}$ 的导数，并设为 0，可以求得：

$$\boldsymbol{K} = \frac{\boldsymbol{P}_k^- \boldsymbol{H}_k^T}{\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k}$$

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-矫正的过程，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：

$$\hat{x}_{k+1}^- = A_k \hat{x}_k + Bu_k$$
$$P_{k+1}^- = A_k P_k A_k^T + Q_k$$

测量更新方程包括：

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

$\boldsymbol{Q}_k$ 和 $\boldsymbol{R}_k$ 均为常数，分别与 w 和 v 相关，估计误差协方差 $\boldsymbol{P}_k$ 和增益矩阵 $\boldsymbol{K}_k$ 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。

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洛方法 (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[10]。

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 $\{x_i, i = 0, \cdots, N\}$，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 $\{w_i, i = 1, \cdots, N\}$，$\sum_{i=1}^{N} w_i = 1$。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。

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（$\in [0°, 360°]$）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2.10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。

在 MeanShift 算法的基础上，Bradski[11] 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。

图 2.10　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

5. 相关滤波

相关滤波算法利用了在傅里叶域中，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。在相关滤波中，目前效果最佳，也最常用的是 KCF 算法（Kernelized Correlation Filters），由 Henriques et al.[12] 在 2015 年提出。

KCF 算法中有三种核函数，包括线性核函数、多项式核函数和高斯核函数。首先使用线性核函数为例，在线性回归函数中使用岭回归（Ridge Regression）以得到一个简单的闭式解。通过训练得到一个函数 $f(\boldsymbol{z}) = \boldsymbol{w}^T \boldsymbol{z}$，使得样本 $\boldsymbol{x}_i$ 及其回归目标 y_i 之间的方差最小，即：

$$\min_{\boldsymbol{w}} \sum_i (f(\boldsymbol{x}_i) - y_i)^2 + \lambda \|\boldsymbol{w}\|^2$$

类似 SVM，这里的 λ 是用于防止过拟合的正则化参数。由于岭回归的性质，上式可以得到一个简单的闭式解 $\boldsymbol{w} = (X^T X + \lambda I)^{-1} X^T \boldsymbol{y}$。其中矩阵 X 的第 i 行为 $\boldsymbol{x}_i$，向量 $\boldsymbol{y}$ 的第 i 个元素为其回归目标 y_i。I 为单位矩阵。上面已经提过，为了提高运算效率，将计算转到傅里叶域中进行，故数据一般都以复数计算，所以将矩阵装置替换为共轭转置，得到：

$$\boldsymbol{w} = (X^H X + \lambda I)^{-1} X^H \boldsymbol{y}$$

这个公式已经可以进行对 $\boldsymbol{w}$ 的计算了，但是由于计算中包括矩阵求逆，计算量很大，难以符合实时性要求。Henriques et al.[12] 巧妙地利用了循环矩阵的性质和离散傅里叶变换，将该运算简化成只需要元素点乘的版本，大大提高了运行速度。具体推导如下：

以单通道一维信号输入为例，对于一个 $n \times 1$ 的向量 $\boldsymbol{x}$，以它为生成向量计算一个循环矩阵 X：

$$X = C(\boldsymbol{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_3 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}$$

循环矩阵的模式是确定的，并完全由它的生成向量向量 $\boldsymbol{x}$ 决定。该矩阵的每一行都是 $\boldsymbol{x}$ 的一种循环位移向量。可以相当于将正样本图像向上/下移动不同的像素来得到新的样本，类似于 MIL 分类器中在正样本周围邻域取正样本包的做法，相当于数据增光，使分类器的效果更好。

除此之外，循环矩阵还有一个重要的特性，即对任意生成向量 $\boldsymbol{x}$，所有循环矩阵都可被离散傅里叶变换（DFT）对角化，即：

$$X = F\mathrm{diag}(\hat{\boldsymbol{x}})F^H$$

其中 $\hat{\boldsymbol{x}}$ 为 $\boldsymbol{x}$ 的离散傅里叶变换，F 为一个与 $\boldsymbol{x}$ 无关的常数矩阵，称为离散傅里叶变换矩阵（DFT matrix），它可以用于计算所有向量的离散傅里叶变换，即 $\mathcal{F}(\hat{z}) = \sqrt{n}F\boldsymbol{z}$。由此，如果我们将输入数据都用其循环矩阵表示，则可以得到：

$$X^H X = F\mathrm{diag}(\hat{\boldsymbol{x}}^*)F^H F\mathrm{diag}(\hat{\boldsymbol{x}})F^H$$

其中 $\hat{\boldsymbol{x}}^*$ 为 $\hat{\boldsymbol{x}}$ 的复共轭向量，由于对角矩阵是对称的，所以共轭转置的结果即为原矩阵的复共轭。此外，根据 F 的性质，$F^H F$ 即为单位矩阵 I。此外，由于在对角矩阵上进行的运算是逐元素的，我们使用 $\odot$ 表示向量或矩阵间的逐元素点乘，则上式可以简化为：

$$X^H X = F\mathrm{diag}(\hat{\boldsymbol{x}}^* \odot \hat{\boldsymbol{x}})F^H$$

将这个公式带入求 $\boldsymbol{w}$ 的公式中，可以最终将 $\boldsymbol{w}$ 的计算过程简化成：

$$\hat{\boldsymbol{w}} = \frac{\hat{\boldsymbol{x}}^* \odot \hat{\boldsymbol{y}}}{\hat{\boldsymbol{x}}^* \odot \hat{\boldsymbol{x}} + \lambda}$$

这里的分数代表逐元素的除法。在求出 $\hat{\boldsymbol{w}}$ 后，可以通过很简单的元素求出其的逆离散傅里叶变换。

//除了线性回归之外，还可以使用核函数拟合出非线性回归。在 KFC 方法中，非线性回归效率在训练和评估时都可以做到和线性回归相似的效率。首先设函数 $\varphi(\boldsymbol{x})$ 用于将线性输入特征向量映射到非线性空间。

// TODO

KCF 算法是缺点是对尺度变化的适应性不强。

6. GOTURN

GOTURN 是一个基于深度神经网络的跟踪算法[13]，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2.11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。

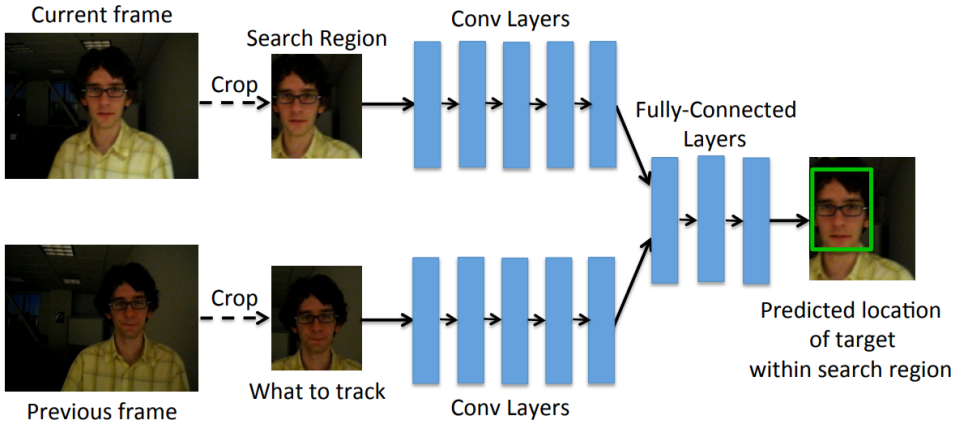

图 2.11　GOTURN 神经网络结构

7. CSR-DCF 算法

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）由 Lukezic et al.[14] 在 2017 年提出，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。

8. 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,[15]。

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。

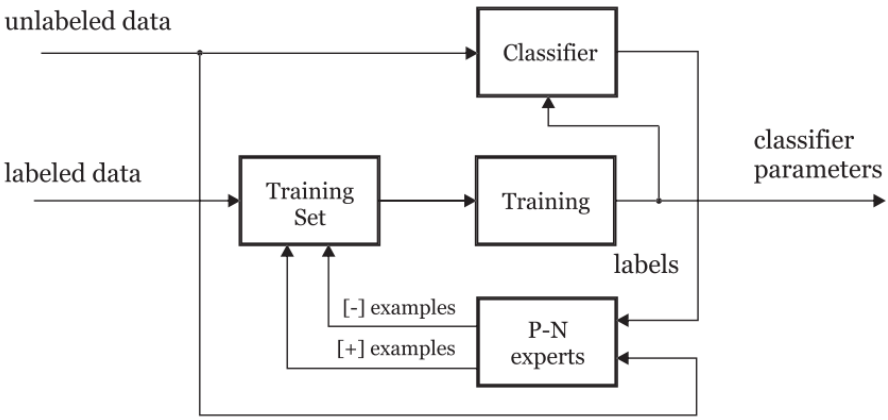

图 2.12　TLD 学习模块结构

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2.12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 ±1% 的偏移，±1% 的尺度变换，±10%° 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。

2.2 激光行人追踪

激光行人追踪的方法主要是对人双腿的追踪。在激光图像中，人的双腿在有一种明显的模式，即 max-min-max-min-max。

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。

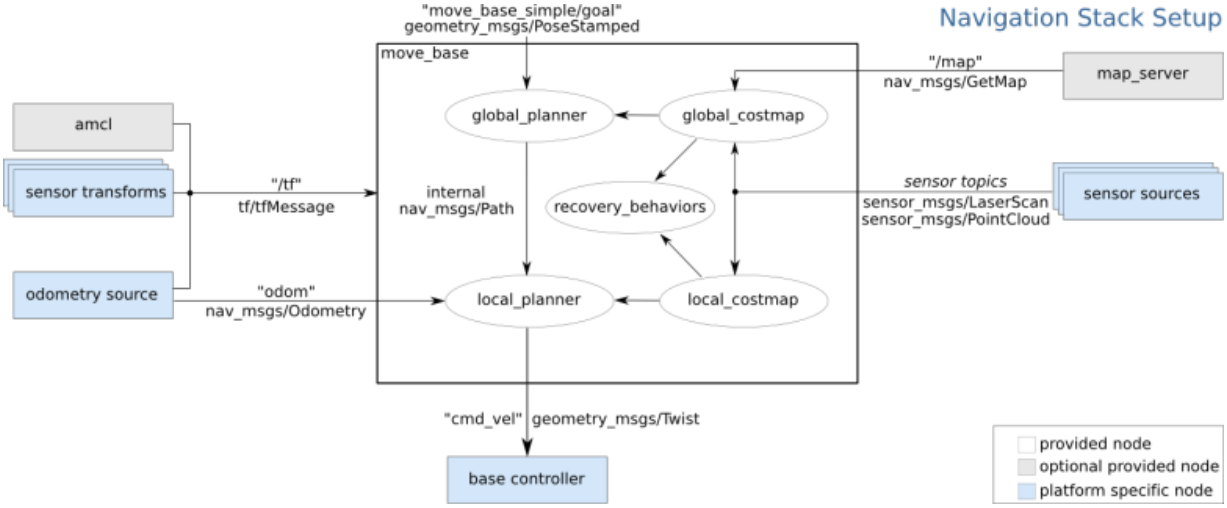

图 3.1　ROS 导航架构

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[16] 的方式进行建图。此外还可以选择 cartographer[17] 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）[18] 进行局部避障，使用路径展开法（Trajectory Rollout）[19] 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

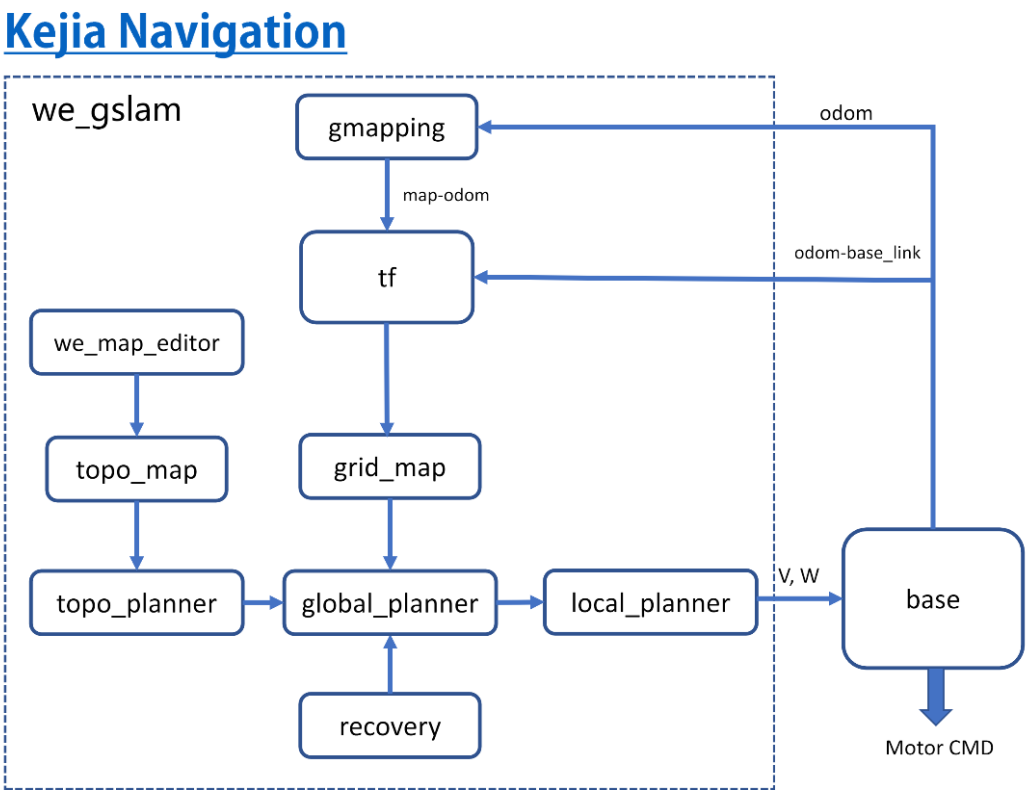

图 3.2 可佳导航架构

可佳导航的架构如图3.2所示。在导航方面，它相对于 ROS 导航，还加入了拓扑导航。使用 VFH（vector field histogram）[20] 进行局部避障。

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[21]，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

给人物设定一个初始的手势，由 OpenPose 系统识别出人物的骨架，进而提取出视野中所有人物的姿势。如要求目标人物将一只手举起，直到机器人识别出目标人物，判断出追踪区域（Region of Interest, ROI），并提示“识别成功”，即开始追踪。

2. 建立一个判别器：根据已有的 ROI 作为正样本，随机提取背景作为负样本，通过在线学习拟合出一个分类器。该判别器的作用是当追踪失去目标或出错时，当再在图像中检测到一个新人物时，判断对方是否是一开始的目标人物。

4.2.3 行人追踪

1. 使用 KFC 和 CSRT，根据“目标人物注册”中得到的 ROI 进行目标追踪。

KCF: 检测成功时的平均帧率：43FPS 在一段时间后框会有点歪框的大小不变 - 不能适应物体的不同尺度。好歹它能在初始尺度上保证人是一直在中心的。

TLD：15 准确率较低，框大小会变，但经常跳动/漂移。。甚至检测到错误的位置。当遮挡和出画面时会得到错误的结果，但恢复能力很强。

CSRT：帧率：18FPS 在人物短暂出画面或者部分遮挡时有时可以恢复，有时不行。。尺度可以在一定程度上变化，但框会变小，难以恢复到开始的水准。。failure 后难以 recover。

MedianFlow 速度超快：270 尺度可变化。晃了一会之后框就变大了。。恢复不到位，人不在框中心。failure 之后没法恢复。

MOSSE 速度超超超超快：450 人一旦出画面就没法恢复了，而且还不报错。。尺度不变，不能很好地处理遮挡。没多久框就漂了。。

主要考虑使用 CSRT，但需要考虑判断是否 failure 及怎样 recover

4.2.4 目标丢失恢复

1. 全局扫描找到行人，结合“目标人物注册”中的判别器，判断是否是目标人物。

2. 结合目标人物最后出现的位置，类似 TLD？

参 考 文 献

[1] Mucientes M, Burgard W. Multiple hypothesis tracking of clusters of people. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006:692-697.

[2] 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.

[3] Foley J D, Van Dam A, et al. Fundamentals of interactive computer graphics: volume 2. Addison-Wesley Reading, MA, 1982.

[4] 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.

[5] 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.

[6] Dalal N, Triggs B. Histograms of oriented gradients for human detection. *international Conference on computer vision & Pattern Recognition (CVPR'05)*, 2005, 1:886-893.

[7] Lowe D G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 2004, 60(2):91-110.

[8] 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.

[9] Welch G, Bishop G, et al. An introduction to the kalman filter. 1995.

[10] 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.

[11] Bradski G R. Computer vision face tracking for use in a perceptual user interface. 1998.

[12] 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.

[13] Held D, Thrun S, Savarese S. Learning to track at 100 fps with deep regression networks. *European Conference Computer Vision (ECCV)*, 2016.

[14] 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.

[15] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 2012, 34(7):1409-1422.

[16] 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.

[17] 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.

[18] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 1997, 4(1):23-33.

[19] Gerkey B P, Konolige K. Planning and control in unstructured terrain. *ICRA Workshop on Path Planning on Costmaps*, 2008.

[20] 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.

[21] Newcombe R A, Izadi S, Hilliges O, et al. Kinectfusion: Real-time dense surface mapping and tracking. *ISMAR*, 2011, 11(2011):127-136.

附录 A　补充材料

补充内容。