SLAM for dummies

同时定位与建图技术方法教程

翻译：何若男
Github：hehern
水平一般，请包含，有错误之处，请联系我！

目录

SLAM for dummies..... 1
2.简介：..... 3
3.关于 slam..... 3
4.硬件..... 3
a.机器人..... 3
b.测距设备..... 4
5.slam 流程..... 4
6.雷达数据..... 7
7.里程计数据..... 8
8.路标..... 8
9.地标提取..... 9
尖峰地标..... 9
RANSAC（随机采样一致算法）..... 10
Multiple strategies（多种策略）：..... 12
10.数据关联..... 12
11.EKF..... 13
过程概述..... 13
The matrices（矩阵）..... 14
The system state: X(系统状态 x)..... 14
The covariance matrix: P（协方差矩阵 P）..... 14
The Kalman gain: K（卡尔曼增益 K）..... 15
The Jacobian of the measurement model: H（测量模型的雅各比矩阵）..... 15
The Jacobian of the prediction model: A（预测模型的雅各比矩阵 A）..... 16
The SLAM specific Jacobians: Jxr and Jz（slam 特殊雅各比矩阵）..... 16
The process noise: Q and W（过程噪声 Q 和 W）..... 17
The measurement noise: R and V（测量噪声 R 和 V）..... 17
Step 1: 使用里程计信息更新当前状态..... 17
Step 2:根据重新观察的地标更新状态..... 18
Step 3: 将新地标添加到当前状态..... 19
12.结束语..... 20

2.简介：

本文档的目的是对移动机器人 SLAM 领域进行教程介绍。SLAM 即同时定位与建图技术。有很多的该主题相关的论文，但是对于该领域的是新手，则需要花费很多时间进行研究以及实施 slam 设计的许多复杂情况。因此希望以清晰简明的方式介绍注意，并同时保持至少需要了解文档的前提条件。实际上在阅读完本文后，应该可以坐下来实施基本的 slam.

slam 可以通过多种方式实现。首先，有大量的不同硬件可以使用。其次。slam 更像是一个概念，而不是单一算法。slam 涉及很多步骤，这些不同步骤可以使用许多不同的算法来实现。在大部分情况下，我们用单一方法解释这些不同的步骤，但涉及了其他可能的方法以便于进一步阅读。

撰写本文的动机主要是为了帮助自己更好地理解 slam。通过教总是可以更好地了解一门学科。再次，现在所有的 slam 论文都是非常理论化的，主要专注于 slam 小领域的创新。本文的目的非常实用，并将重点放在简单的基本 slam 方法上可以被那些刚刚接触 slam 的人来使用。对于那些对 slam 有一定背景知识的人，我们为此使用了 EKF（扩展卡尔曼滤波）slam。本文比较完全但是并不完美。我们的意思是，我们涵盖了实施以及运行的所有基本步骤。还必须注意的是，slam 问题尚未完全解决，而且该领域仍在进行着大量研究。

为了使入门变得容易，我们提供了所有的代码，因此基本上只是一个下载、编译、插入硬件（SICK 激光雷达、ER1 机器人），并执行应用程序。额外热切是即插即用的。我们已经使用了 Microsift Visual5 c#，并且代码在.Net Framework 框架下进行编译。大部分的代码非常简单明了，可以被当作伪代码读，因此移植到其他语言或者平台上应该也很容易。

3.关于 slam

slam 术语是同时定位与建图的缩写。它最初是由 Hugh Durrant-Whyte 和 John J. Leonard 在基于 Smith, Self ，Cheeseman 三个人的工作上开发而来的。最初他们命名为 smal，后来对其进行了更改。slam 多解决的问题是，移动机器人在未知的环境中构建地图，并同时使用地图进行导航。

slam 由多个部分组成：地标提取、数据关联、状态估计、状态更新以及地标更新。解决每一个小的部分都有很多方法。每个部分我们都会举几个例子。这也意味着某些部分可以被一种新的方法来代替。例如，我们可以采用两种方式解决地表提取问题，并且评价下这两种方式。这个想法是，我们可以使用我们自己的新颖的想法来实现以 sita 及扩展这些算法。我们已经决定将重点放在室内环境的移动机器人上。您可以选择更改其中的一些算法，以便例如可以在其他的环境中使用。

slam 适用于 2D 和 3D 运动，我们接下来只考虑 3D 运动。

如果读者熟悉 slam 的一般概念并处于入门级别上，如已经通过了该主题的大学课程。这将很有帮助。有很多该领域的精彩介绍，参见【6】【4】.这对了解卡尔曼滤波也很有帮助。背景信息总是有帮助的，因为它将使您更轻松地了解本教程，但并不能理解本文的全部内容。

4.硬件

机器人的硬件非常重要。要进行 slam，需要一个移动机器人和一个测距设备。我们考虑的移动机器人是轮式室内机器人。本文档主要专注软件实现，不会探索复杂的运动模型（机器人运动的模型）例如人形机器人、自动水下航行器、自动飞机、具有怪异车轮装置的机器人等。我们这里介绍了一些常用关于移动机器人 slam 的基本测量设备。

a.机器人

要考虑的主要参数有易用性、里程计性能和价格。机器人通过车轮的旋转来估计自身的位置，这个性能反应了里程计的性能。机器人的里程计精度不能超过移动时的 2cm/m 以及旋转时的 2°/45°。典型的机器人驱动程序允许机器人报告其在笛卡尔坐标系下的坐标位置（x，y），同时也报告机器人当前的方向。

可以选择从头开始创建机器人，这可能非常耗时，但也是一种学习经验。也可以选择购买现成的机器人如 ER1。RW1 机器人目前暂不售卖，但是，他是世界上很多计算机科学实验室中经常用到的。RW1 机器人的测距很差，这使得估计当前位置以及 slam 相当困难。ER1 是目前我们正在用的，它小而且便宜。200 美金可以购买用于学术使用，300 美金可以购买用于个人使用。它带有一个相机和机器人控制系统。我们在附录以及网站中提供了非常基本的驱动。

b.测距设备

如今所使用的测距设备艇长时激光雷达。他们的精确度非常高、高效，而且它的输出结果不需要太多计算处理。不利的一面是他们价格非常的昂贵。一台 SICK 品牌的激光雷达价值 5000 美金。激光雷达的问题是，它扫面物体的表面，包括玻璃，这种情况下输出数据可能是错的（雷达线束是红外光谱，可以穿过玻璃，当雷达线束照在玻璃上时，一部分光会穿透玻璃照在后面的物体上，这就导致测距不准的情况）。激光雷达同时也不能在水下使用，因为水会干扰光线，并且会减少范围。

第二个选择是声呐（毫米波雷达？）。声呐在前几年被大量使用。他们与激光雷达相比价格非常便宜。他们的测量结果与激光雷达不好相比，而且经常会给出不好的输出。激光雷达单线扫描一周 360° 精度最小可以达到 0.25°，声呐的波束宽度可以达到 30°。但是，在水下他们是最佳选择，类似于海豚的导航方式。典型的应用是宝丽来声呐。它最初是为了测量宝丽来相机拍照时的最大距离。声呐已经成功地在【7】中使用。

第三种是可以选则使用视觉方式。传统上，在计算机上大量使用了视觉，并且由于光线变化也容易出错。给 i 顶一个没有照明地房间,视觉系统几乎肯定无法工作。不过最近几年在该领域内已经取得了一些有趣的进展。通常系统可以使用立体相机系统或者 triclops 系统来测量距离。使用视觉就像人类注视着这个世界一样，因此与激光雷达和声呐相比更加的直观。同时图片中的信息和激光雷达和声呐相比信息更加的丰富。这曾经是瓶颈，因为所有的信息都要被处理，但是随着算法以及计算机计算能力的进步，这已经变得不是问题。基于视觉的距离测量已经在【8】中成功使用了。

我们选择了 SICK 激光雷达。它使用非常的广泛，而且它对眼睛没有危险，并且在 slam 使用时具有不错的性能。测量误差在+-50mm，这个数据看起来非常大，但在实际中这个无法非常小。最新的 SICK 激光雷达的测量误差低至+-5mm。

5.slam 流程

slam 过程包括很多步骤。过程的目的是使用环境来更新机器人的位置。由于机器人的里程计（这赋予了机器人位置）通常是错误的，我们不能直接依靠里程计。我们可以使用激光雷达扫描的环境来纠正机器人的位置。这是通过从环境中提取特征，并在机器人四处移动时重新观察来完成的。EKF 是 slam 流程的核心。它负责更新基于这些特征机器人认为的位置。这些特征通常被称为地标，在接下来的几章中将于 EKF 一起进行解释。EKF 会对机器人位置的不确定性以及他在环境中看到地标的不确定性进行跟踪估计。

slam 过程概述如下：

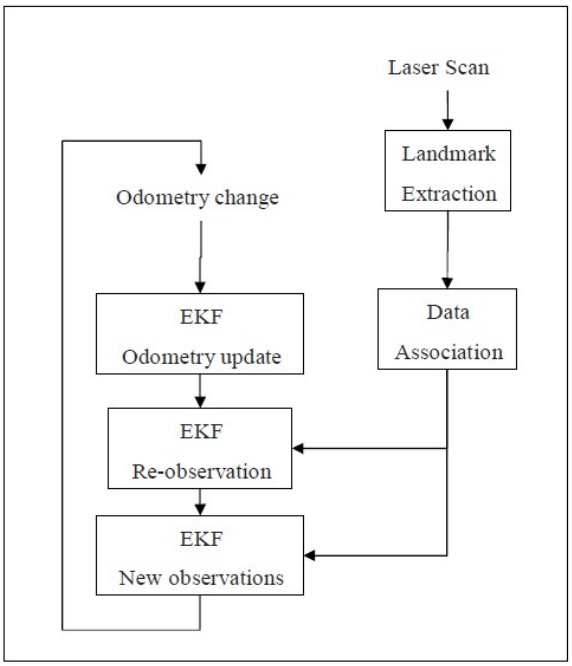


Figure 1 Overview of the SLAM process

图 1slam 流程

当由于机器人的移动而使里程计发生变化时，机器人的最新位置的不确定性由里程计经过 EKF 更新得到。路标是从机器人的新位置环境中提取出来的。然后，机器人尝试将现在得到的路标与之前看到的路标关联在一起。再次观察的路标将用于更新机器人在 EKF 中的位置。以前未见过的路标将作为新路标添加到 EKF 中观察，以便于在以后可以再次观察。所有这些步骤将在下一章以非常实用的方式介绍 ER1 机器人实现。应当指出，在这些步骤的任何时候，EKF 都会对当前机器人的位置进行估计。

下图将更详细地介绍此过程：

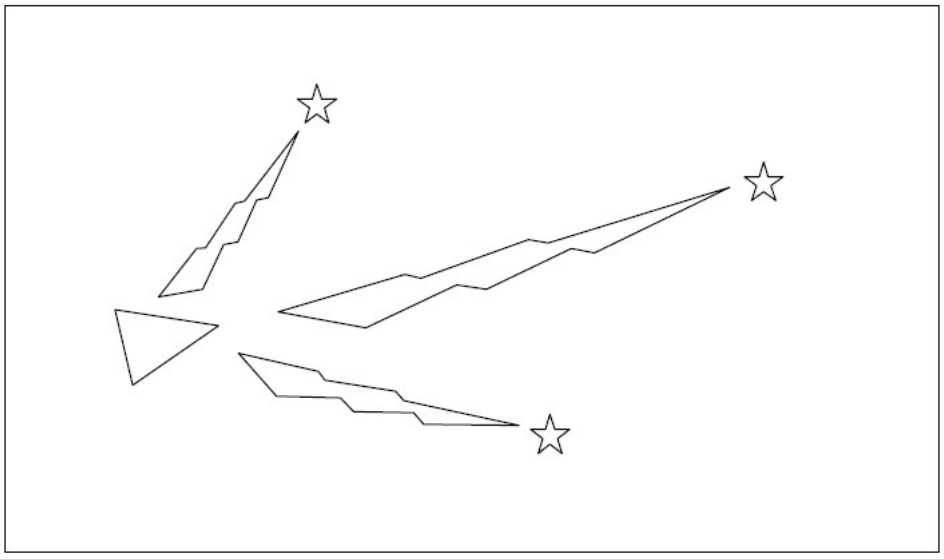


Figure 2 The robot is represented by the triangle. The stars represent landmarks. The robot initially measures using its sensors the location of the landmarks (sensor measurements illustrated with lightning).

图 2 机器人由三角形表示，星星代表地标。机器人使用其传感器测量地标地位置（传感器测量如图中闪电图示）。

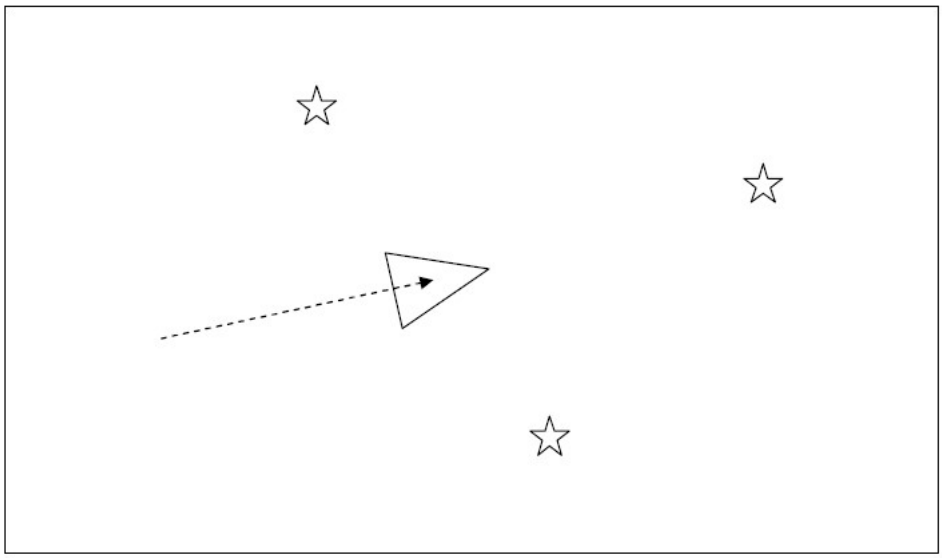


Figure 3 The robot moves so it now thinks it is here. The distance moved is given by the robots odometry.

图 3 机器人在移动，此时我们认为它在这个位置。机器人移动地距离由机器人里程计给出。

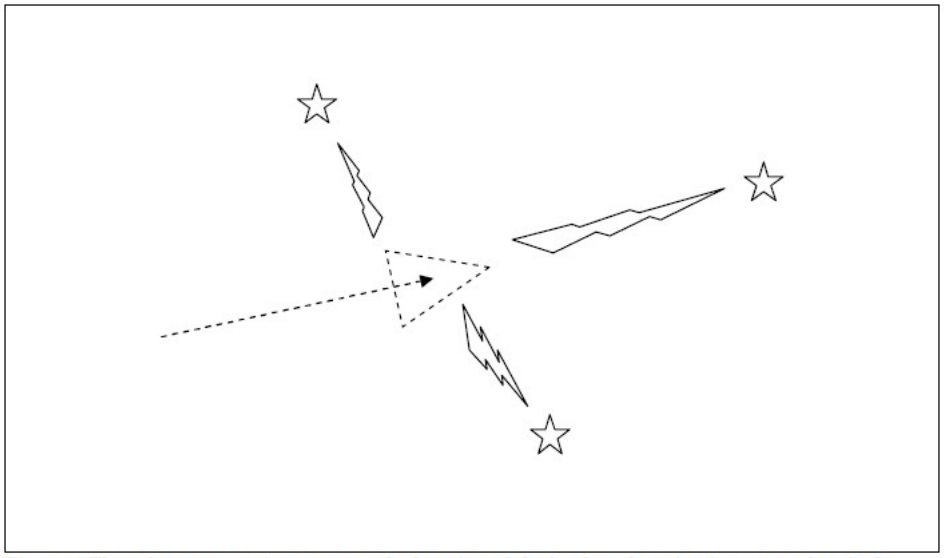


Figure 4 The robot once again measures the location of the landmarks using its sensors but finds out they don't match with where the robot thinks they should be (given the robots location). Thus the robot is not where it thinks it is.

图 4 机器人再次使用其传感器测量地标位置，但发现他们与机器人认为地位置不符（由里程计推算的位置）。因此，机器人不在它自认为所在的位置。

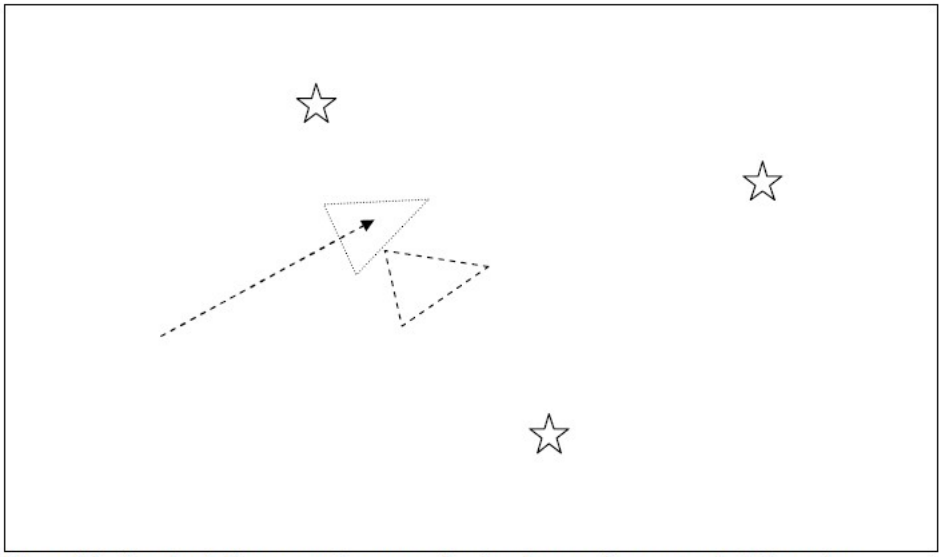


Figure 5 As the robot believes more its sensors than its odometry it now uses the information gained about where the landmarks actually are to determine where it is (the location the robot originally thought it was at is illustrated by the dashed triangle).

图 5 由于机器人更加相信传感器测量的结果而不是里程计，因此现在使用地标测量得到的信息来确定其所在的位置（粗虚线三角形表示机器人原本以为所在的位置，细虚线表示根据路标得到的机器人位置）。

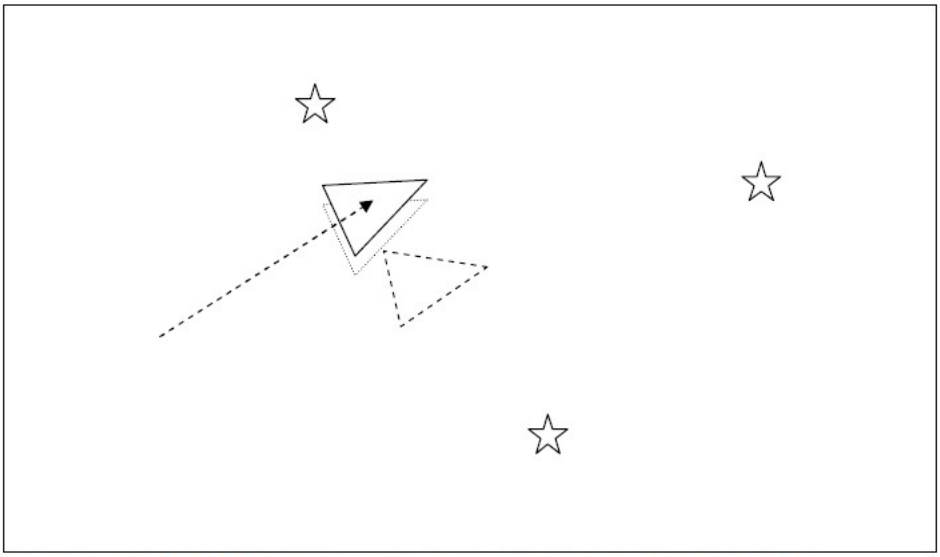


Figure 6 In actual fact the robot is here. The sensors are not perfect so the robot will not precisely know where it is. However this estimate is better than relying on odometry alone. The dotted triangle represents where it thinks it is; the dashed triangle where odometry told it it was; and the last triangle where it actually is.

图 6 实际上，机器人在黑色实线三角形位置。传感器也不是完美的，所以机器人不会精确地知道自身所在地位置。但是，这样的估计结果也比只依赖于里程计得到的结果准确。

6.雷达数据

slam 过程的第一步是获取环境数据。当我们使用激光雷达时，我们得到的是激光雷达数据。我们使用的 SICK 激光雷达的测量角度范围是 100° 或者 180° 。垂直分辨率为 0.25° ， 0.5° ， 1.0° ，表示激光束的宽度间隔为 0.25° ， 0.5° ， 1.0° 。典型的激光扫描仪输出看起来像这样：2.98、2.99、3.00、3.01、3.00、3.49、3.50，....，2.20、8.17、2.21。激光扫描仪的输出是从右到左以米未单位。

如果激光雷达由于某些原因无法确定特定角度的确切距离，将返回一个最高值，我们使用 8.1 作为阈值来判断该值是都为错误。某些激光雷达可以测量超过 8.1 米的距离范围。最后，应注意激光雷达扫描起来速度特别快，使用串口需要至少 11hz 频率查询。与激光雷达对接的代码可以在附录 B：SICK LMS 中找到。

7.里程计数据

slam 的另一个重要方面就是里程计数据。里程计数据的作用是通过测量机器人的轮子的移动，提供一个机器人的大致位置作为 EKF 的初始估计值。从 ER1 机器人中获得里程计数据非常简单，需要使用内置的网络服务器。只需要将文本字符串发送到特定的端口，服务器就会返回答案。关于里程计数据和激光雷达数据的困难部分在于时间同步。如果在稍后时间检测到里程计数据，则在某个时间 t 的激光雷达数据将是过时的。为了确保他们在同一时间有效，可以对数据进行推断。由于控件时已知的，因此与 i 段里程计数据最容易。预测激光雷达的检测结果是很难的。如果可以控制何时返回测量值，那么最简单的方法是让激光雷达数据和里程计数据同时显示。

8.路标

路标是很容易重新观察并从环境中区别的特征。机器人使用这些来确定自己的位置（自定位）。想象机器人如何工作的一种方式是蒙上自己的眼睛。如果蒙住眼睛在房间里四处走动，您可能会伸出手并触摸物体或拥抱墙壁，这样就不会迷路。特征性的物体，例如通过触摸门框可能会帮助您确定您的位置。声呐和激光雷达是机器人的触觉。以下是来自不同环境的良好地标示例：

图 7：自由女神像是一个很好的地标，因为它是独特的，并且可以从不同环境以为位置上看到，例如陆地上、海上或者空中。

图 8：码头上的木柱可能是水下航行器的好标志。

如你所见，机器人使用的地标类型通常取决与机器人所处的环境。

地标应该是可以被从不同位置不同角度重复观察得到的。地标应该足够独特，以便可以轻松地从一个时间步到另一个时间步识别，而不会混淆他们。换句话说，如果在之后地某个时间点重新观察两个路标，应该能很容易地确定哪个路标是我们西拿钱所见地路标。如果两个路标非常接近，这可能很难。您决定地机器人识别地路标不能太少，否则机器人会花费大量的时间，还有可能会迷路。

如果你决定将某个东西作为路标，那么它应该是静止的。使用一个人作为路标是个坏主意。这个标准的原因相当直截了当。如果路标不能总是在同一地点，那么机器人怎么通过给定的路标判断出自身的位置呢。

合适的路标的关键要点如下：

- 地标应易于重新观察；
- 地标应该各个路标应相互区分；
- 地标应在环境中丰富；
- 地标应该是固定不会动的。

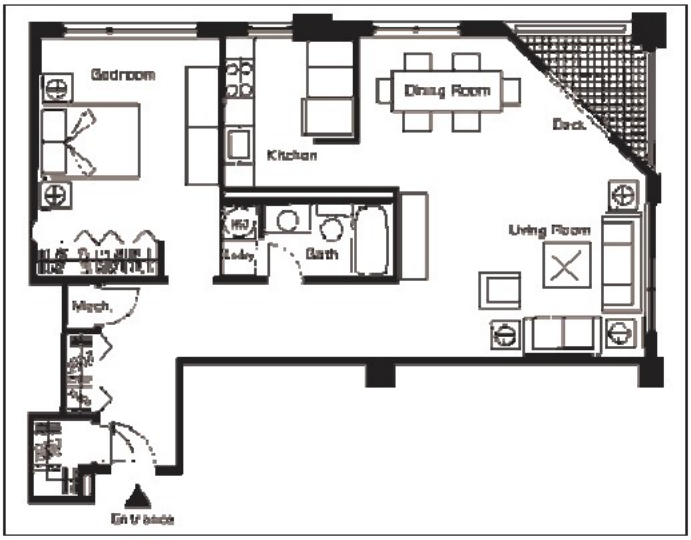


Figure 9 In an indoor environment such as that used by our robot there are many straight lines and well defined corners. These could all be used as landmarks.

图 9 在我们使用的机器人室内环境中，有很多直线和明确定义的拐角。这些都可以用作地标。

9.地标提取

一旦我们确定了机器人应该使用的地标，我们就需要从机器人的传感器输入中以某种方式可靠地提取它们。

如简介中所述，有很多种方式可以进行地标提取。这主要取决于尝试提取哪种类型地地标以及使用什么传感器。

我们将介绍使用激光雷达地基本地标提取算法。我们使用两种地标提取算法成为 Spikes 和 RANSAC。

尖峰地标

尖峰地标提取是用极值来查找地标。通过在激光扫描范围内找到两个值相差特定值来识别它们，例如 0.5m。这将在雷达扫描中找到最大变化值。当某些激光线束从墙面反射回来，而有些线束没有撞到墙壁，而是从墙后面的东西反射回来。

Figure 10: Spike landmarks. The red dots are table legs extracted as landmarks.

图 10 尖峰地标，红点是提取作为路标的桌子腿

峰值也可以通过使三个值 ABC 彼此相邻来找到峰值，从 A 减去 B 并从 C 减去 B 并将两个数字相加将得到一个值。此方法更适合发现峰值，因为它会发现实际峰值，而不仅仅是固定的距离差异。

尖峰地标依赖于两个激光线束之间的变化很大的情况。这意味着该算法将在平滑环境中失效。

RANSAC（随机采样一致算法）

随机采样一致算法是一种可用于从激光雷达扫描结果中提取线条的方法。这些线可以用来作为地标。在室内环境中，直线经常被检测到，因为墙通常是直的。

RANSAC 通过随机获取激光雷达样本，然后使用最小二乘法近似找到贯穿这些点的最佳拟合线，从而找到这些边界线。完成此操作后，ransac 会检查有多少激光雷达点接近此最佳拟合线。如果该数字高于某个阈值，则可以安全地假定我们已经看到一条线（墙）。此阈值称为 consensus。

以下算法描述了激光雷达边界线提取过程。算法假设将雷达数据转换到笛卡尔坐标系下，坐标转换参见附录 A。最初假设所有雷达点与线无关。在该算法中，我们仅从未关联地点中采样提取线束。

```
while
 .仍存在未关联地激光雷达点
 .未关联地激光雷达点地个数大于consensus值
 .已经获取地线个数少于N、
do
 .随机选择雷达点数据
 .在D度范围内随机采样S个雷达数据（例如，从随机采样的数据中随机选择读书在10度以内的5个点）
 .使用这S个样本的原始数据采用最小二乘法拟合线
 .确定最佳拟合线X厘米内有多少雷达点
 .如果在该线上的雷达点个数超过常数C，则执行以下操作：
     .根据已经确定的所有旧最佳拟合线上的所有点数据，计算新的最小二乘拟合线
     .将此最佳拟合线添加到我们已经提取的线中
     .从所有未关联的点中删除该线上的所有点
od
```

可以通过调整以下参数调整该算法：

N-尝试查找线的最大个数

S-计算初始线的样本数

D-初始度数中样本的度数

X-关联线的最大距离

C-直线上必须存在的最小点数

EKF 实现的方式是假定地标品味与机器人位置的一定范围内。通过计算线中距离世界坐标中的一个固定点的最近一个点，我们可以轻易地将一条线转换为一个固定点。使用机器人的位置以及直线中该固定点的位置来计算距离和方位是很简单的。

使用简单的三角函数可以轻松地计算出这一点，这里说明原点为固定点：

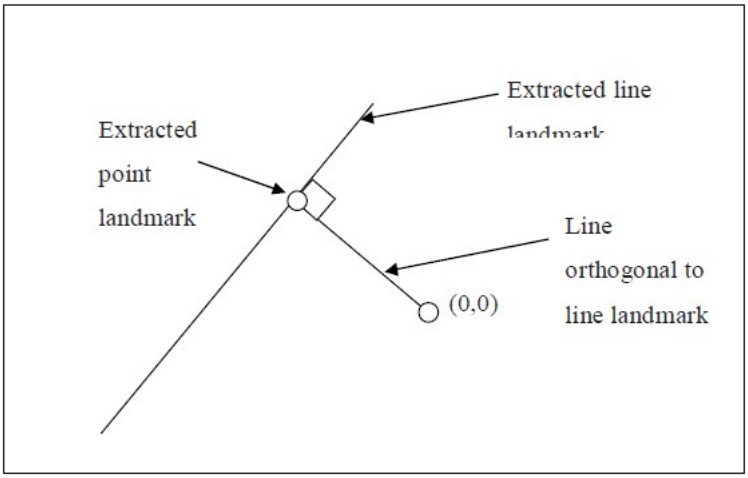


Figure 11 Illustration of how to get an extracted line landmark as a point.

图 11 如何将获取地直线转换为点地示意图

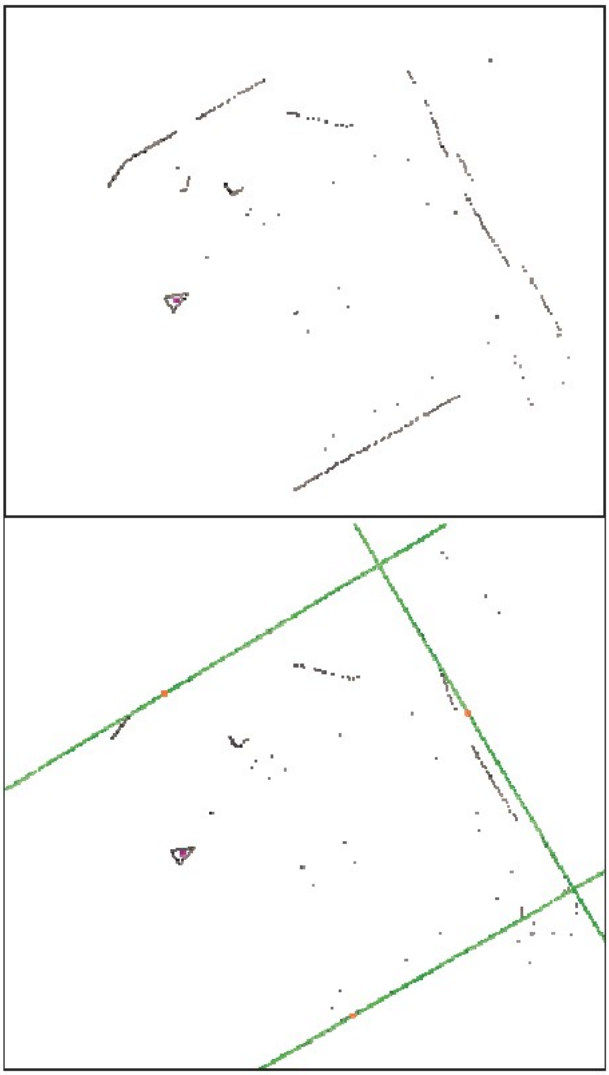


图 12 ransac 算法在激光雷达数据中找到地主要直线。绿线是代表地标地最合适的线。红点代表线转换为的点。通过更改 ransac 参数，我们还可以提取小墙段。他们不被认为是非常可靠的地标，因此未使用。最后，在机器人上方的是一个人，ransac 对于检测人同样具有鲁棒性。

另一种实施方式是，扩展 EKF 的实现，使其可以处理线而不仅仅只是点。但是这很复杂，因此在本教程中不会处理。

Multiple strategies（多种策略）：

我们提供了两种不同的地标提取方法。两种都提取了不同的地标，并适用于室内环境。但是，尖峰特征非常简单，并且在有人的环境中并不鲁棒。原因是，尖峰会将人视为尖峰，因为理论上他们是良好的地标。

使用 ransac 提取线特征，不会将人作为地标，因为他们没有线特征。我们不会介绍的第三种方法称为扫描匹配方法，匹配两个连续的激光扫描数据。

10.数据关联

数据关联的问题是不同雷达扫描线束的匹配问题。我们也称为重观察坐标。为了说明这是什么意思，我们将举一个例子：

对于我们人类而言，我们可以将椅子作为地标。假设我们在一个房间看到一个特定的椅子。现在我们离开房间，然后在以后的某个时间点返回房间，如果我们看到房间的椅子，并说它与我们之前看到的椅子是同一把椅子，那么我们已经将椅子与旧椅子相关联了。这似乎很简单，但是数据关联其实是很难做的。假设房间里有两把椅子，看上去几乎一样，当我 i 们返回房间时，我们无法准确地区分哪把椅子时我们最初看到地那把椅子（因为他们看起来都一样）。我们最好的选择是，左边的那个必须是我们之前在左边看到的那个，右边那个之前右边看到的那个。

实际上，数据关联可能会出现以下问题：

-你可能不会每次都重新观察地标

-你可能会观察到一些地标，但是之后再也观察不到了

-你可能错误地关联了地标

如地标一章描述，重新观察地标是很容易地。因此，上面地前两种情况对于地标来说是不能接受地。换句话说，他们都是不良地标。即使你有一个非常好的地标提取方法，你也可能会碰到这些问题，因此最好定义一个合适地数据关联策略以将其最小化。

最后一个问题，错误地将地标关联是毁灭性的。这表示机器人所认为的位置可能与实际位置不同。

现在我们将定义一个处理这些问题的数据关联策略。我们假定已经建立了一个数据库来存储我们之前见过的地标。该数据库通常最初是空的。我们设置的第一条规则是，除非我们已经看过 N 次，否则我 i 们并不认为这是 slam 中的地标。这消除了不良地标提取的情况。下文中将进一步验证。

1.当你进行新的激光扫描时，请使用地标提取提取所有的可见地标；

2.将每个地标与我们在数据库中看到的 N 次以上的最接近的地标进行关联；

3.将每对关联地标（提取的地标与数据库中的地标）进行关联验证：

a.如果通过了验证，表示数据库中的该地标与我们重新观察得到的地标相同，因此请增加数据库中看到它的次数。

b.如果未通过验证，则将该地标添加到数据库中作为新地标，并将看到它的次数设置为 1.

当将当前观察地标与数据库进行关联时，该技术称为最近邻方法。计算最近的地标的最简单方法时计算欧几里得距离。另一种方法时计算更好但更复杂的马氏距离。在我们的方法中并没有使用，因为 ransac 地标通常相距较远，这使得使用欧几里得距离变得更加合适。

验证门使用了以下事实：我们的 EKF 实现对地标检测的不确定性有一定的限制。因此我们可以通过检查地标是否位于不确定区域内来确定观察到的地标是否是数据库中的地标。该区域实际上可以以图像方式绘制，并且称之为错误椭圆。

设置常数 λ，通过以下公式将观察地标与数据库地标进行关联：

$$\nu_i^T \mathbf{S}_i^{-1} \nu_i \leq \lambda.$$

vi 是 EKF 中定义的 innovation, Si 是协方差矩阵.

11.EKF

扩展卡尔曼滤波用于从里程计数据和地标观察值估计机器人的状态（位置）。通常以状态估计的形式描述 EKF（机器人已经有了一个完善的地图）。也就是说，EKF 不用于 slam 的地图更新过程。在 slam 与状态估计 EKF 中，尤其是状态已更改，可能很难弄清楚如何实现，因为几乎从未在任何地方提及它。我们将经历所有这些。大部分 EKF 是标准的，作为常规 EKF，一旦建立了矩阵，它基本上只是一组方程式。

过程概述

一旦地标提取和数据关联到位，slam 过程可以视为三个步骤：

1.使用里程计数据更新当前状态估计；

2.通过重新观察地标更新估计状态；

3.将新的地标添加到当前状态。

第一步很容易。这只是控制对机器人旧状态估计的补充。机器人当前位置为（x，y）旋转角度为 theta ，施加的控制为（dx，dy）旋转变化为 dtheta。第一步的结果也就是新状态为（x+dx，y+dy）,角度为 theta+dtheta。

第二步，考虑再次观察到的地标。使用当前的位置估计，可以估计出地标应该在哪里。这通常存在一些差异，称之为 innovation。因此，innovation 基本上是机器人的位置状态估计与实际位置之间的差异。在第二步中，还将更新每个观察到的地标的不确定性以反映最近的变化。例如，当前地标位置的不确定性很小。当前位置的具有较低不确定性的重新观察地标将增加地标确定性，即地标相对于机器人当前位置的变化。

第三步，使用当前位置的有关信息并添加有关新地标和就地表之间关系的信息，将新地标添加到状态（机器人世界地图）中。

The matrices（矩阵）

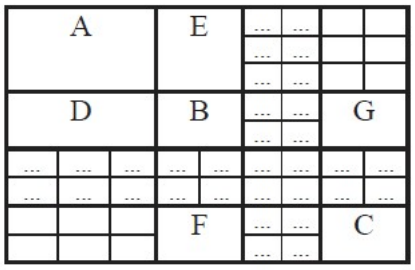
应当注意的是，在不同的论文中，对于相同的变量有很多不同的概念名称。我们使用一些相当普遍的概念。

The system state: X(系统状态 x)

x_r
y_r
$theta_r$
x_1
y_1
...
...
x_n
y_n

状态矩阵 X 和协方差矩阵一起是系统中最重要的矩阵之一。它包含机器人的位置 x，y 和 theta。 此外，它还包含每个地标的位置 x 和 y。矩阵如右侧图所示。将矩阵作为垂直矩阵很重要，以确保所有方程式均有效。X 的大小为 1 列宽，高度为 3+2*n 行，其中 n 为地标个数。通常，保存的值以米或者毫米为单位。使用那个度量单位并不重要，重要的是使用相同的单位。角度以度或者弧度保存。同样，角度的单位也要统一。

The covariance matrix: P（协方差矩阵 P）



快速数学回顾：两个变量的协方差表示两个变量的相关程度的度量。相关性是用于测量变量之间线性相关程度的概念。

协方差矩阵 P 是系统中非常中心的矩阵。它包含机器人位置的协方差，机器人位置和地标之间的协方差，以及地标之间的协方差。右图显示了协方差矩阵 P 的内容。第一个单元格 A 包含机器人位置的协方差，它是一个 3*3 的矩阵（x，y，theta）。B 是第一个地标的协方差，他是一个 2*2 矩阵，因为地标是没有方向 theta 的。这一直持续到 C，C 是最后一个地标。单元格 D 为机器人状态和第一个地标之间的协方差。单元格 E 为第一个地标与机器人状态之间的协方差，D 和 E 是互为转置的关系。F 包含最后一个地标和第一个地标之间的协方差，而 G 包含第一个地标与最后一个地标之间的协方差，同样可以通过转置 F 来推导。因此即使协方差矩阵看似复杂，它实际上还是非常系统地建立起来地。最初，由于机器人没有看到任何地标，协方差矩阵 P 仅包括矩阵 A。必须使用一些默认值初始化协方差矩阵地对角线。这反映了初始位置地不确定性。如果某些计算中不包含初始不确定性，在某些实际实施情况下会出现奇异错误。因此，即使相信初始位置是正确的，最好还是包含一些初始误差。

The Kalman gain: K（卡尔曼增益 K）

x_r	x_b
y_r	y_b
t_r	t_b
$x_{1,r}$	$x_{1,b}$
$y_{1,r}$	$y_{1,b}$
...	...
...	...
$x_{n,r}$	$x_{n,b}$
$y_{n,r}$	$y_{n,b}$

计算卡尔曼增益 K 可以找出我们对观察到地地标地信任程度，以及我们希望从他们提供地信息中获得多少。如果我们看到机器人应该向右移动 10cm，则根据地标，沃恩使用卡尔曼增益来找出我们实际校正地位置量可能只有 5cm，因为我们不完全相信地标，而是在里程计和地标之间找到折衷方案。这通常是通过地标的不确定性、距离测量设备的测量精度以及里程计的性能确定的。如果距离测量设备的性能比机器人里程计的性能差，我们当然不会非常信任它，因此卡尔曼增益会很低。相反，如果测量设备的测距性能非常好，则卡尔曼增益将会很高。矩阵可以从右侧看到。第一行显示了状态 X 的第一行应从 innovation 中获得多少。第一行的第一列描述了从地标中获得的增益，第二列描述了从里程计中获得的增益。两者更新的都是机器人位置 X 值。矩阵前三行是机器人位置，后面每两行是地标位置，高度为 3+2*n，其中 n 为地标数量。

The Jacobian of the measurement model: H（测量模型的雅各比矩阵）

测量模型的的雅各比矩阵与测量模型密切相关，因此我们先了解以下测量模型。测量模型定义了如何计算测量目标（地标）的距离与方位。由以下公式 H 表示：

$$\begin{bmatrix} \text{range} \\ \text{bearing} \end{bmatrix} = \begin{bmatrix} \sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2} + v_r \\ \tan^{-1}\left(\frac{\lambda_y - y}{\lambda_x - x}\right) - \theta + v_\theta \end{bmatrix}$$

lambda x 是地标位置的 x 坐标，x 是机器人位置的 x 坐标，lambda y 是地标位置的 y 坐标，y 是机器人位置的 y 坐标。Theta 是机器人的角度。这提供了地标的预测距离和方位，雅各比矩阵公式如下：

$$\begin{bmatrix} \frac{x - \lambda_x}{r} & \frac{y - \lambda_y}{r} & 0 \\ \frac{\lambda_y - y}{r^2} & \frac{\lambda_x - x}{r^2} & -1 \end{bmatrix}$$

H 公式展示了 range 和 bear 随着 x y theta 的变化而变化的程度。第一行的第一个元素是相随与 x 轴变化的 range 变化，第二个元素是关于 y 轴的变化。最后一个元素是关于 Theta 的变化，即机器人的旋转。当然该值是 0，因为 range 不会随着机器人的旋转而改变。第二行是对 bearing 的改变。这是常规 EKF 状态估计常用 H 的内容。做 slam 时，我们需要地标的一些其他值：

X_r	Y_r	T_r	X_1	Y_1	X_2	Y_2	X_3	Y_3
A	B	C	0	0	-A	-B	0	0
D	E	F	0	0	-D	-E	0	0

但是用矩阵 H 时，对于第二个地标，我们将使用上面的矩阵。上面的行仅供参考，他不是矩阵的一部分。这意味着与常规 EKF 状态估计一样，前 3 列时常规 H。对于每一个地标，我们添加 2 列。当对上面的地标 2 使用 H 矩阵时，我们将 x2 设置为-A -D 并将 y2 设置为 -B -E，其余地标列设置为 0.x2 y2 列的数据与原始的 H 矩阵前两列相同，只是添加了个负号。由于地标没有旋转，所以每个地标也只有两列。

The Jacobian of the prediction model: A（预测模型的雅各比矩阵 A）

与 H 一样，预测模型的雅可比矩阵业余预测模型密切相关，因此，让我们首先了解以下预测模型。预测模型定义了在给定旧位置和控制输入的情况下如何计算机器人的预测位置。使用以下公式完成，表示为 F：

$$f = \begin{bmatrix} x + \Delta t \cos\theta + q\Delta t \cos\theta \\ y + \Delta t \sin\theta + q\Delta t \sin\theta \\ \theta + \Delta\theta + q\Delta\theta \end{bmatrix}$$

其中 x 和 y 时机器人位置，theta 是机器人旋转，t 是推力的变化，q 是误差项。我们直接使用 ER1 系统的里程计信息获得位置的变化，因此我们直接使用 x y theta 以及过程噪声，如下所述：

x + Δx + Δx * q
y + Δy + Δy * q
theta + Δtheta + Δtheta * q

无论如何，我们计算雅各比矩阵 A 时，假设为线性模型。

$$\begin{bmatrix} 1 & 0 & -\Delta t \sin\theta \\ 0 & 1 & \Delta t \cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

与计算 H 矩阵相同，除了多了一行旋转。由于它仅用于机器人位置预测，因此不会扩展到其他地标。从第一个矩阵即预测模型可以看出， t * sin theta 等于△y，t * cos theta 等于△x。所以，我们的控制项可以表述如下：

1	0	-Δy
0	1	Δx
0	0	1

The SLAM specific Jacobians: Jxr and Jz（slam 特殊雅各比矩阵）

做 slam 时，有一些雅各比矩阵只在 slam 中使用。当然，这是在集成新功能的过程中，这是唯一与使用常规 EKF 进行状态估计的不同之处。第一个时 Jxr，它与测测模型的雅各比矩阵基本相同，只是没有旋转项。它时地标预测的雅各比矩阵，因此只有 xy 不包含 theta。

$$J_{xr} = \begin{bmatrix} 1 & 0 & -\Delta t \sin\theta \\ 0 & 1 & \Delta t \cos\theta \end{bmatrix}$$

雅各比矩阵 Jz 也是地标预测模型的雅各比矩阵，但是时关于【范围 方位】的。如下所示：

$$J_z = \begin{bmatrix} \cos(\theta + \Delta\theta) & -\Delta t*\sin(\theta + \Delta\theta) \\ \sin(\theta + \Delta\theta) & \Delta t*\cos(\theta + \Delta\theta) \end{bmatrix}$$

The process noise: Q and W（过程噪声 Q 和 W）

$c\Delta x^2$		
	$c\Delta y^2$	
		$c\Delta t^2$

假设测量过程中存在高斯噪声。噪声定义为 Q，他是一个 3*3 矩阵。通常通过将一些高斯样本 C 乘以 W 和 W 的转置来计算。

$$W = \begin{bmatrix} \Delta t\cos\theta & \Delta t\sin\theta & \Delta\theta \end{bmatrix}^T$$

$$Q = WCW^T$$

C 表示里程计的精确度。该值应根据机器人的里程计性能来设置，通常最容易通过实验来设置并调整该值。在大多数论文中，过程噪声要么单独表示为 Q，要么表示为 WQWt，基本上从不使用概念 C，但在这里我们展示了这两种方式。

The measurement noise: R and V（测量噪声 R 和 V）

rc	
	bd

我们还假设距离测量设备具有和距离方位呈正比的高斯白噪声。计算公式为 VRVt。V 是一个 2*2 的恒等矩阵。R 是一个 2*2 的对角线矩阵。在左上角，我们用范围 r 乘以一些常数 c 和 d，这些常数代表测量设备的精度。例如，如果距离误差的方差为 1cm，则 c 应为方差为 0.01 的高斯。如果角度误差为 1 度，则应将 d 设置为 1.通常，使用角度误差和角度大小成正比是没有意义的。

Step 1: 使用里程计信息更新当前状态

此步骤称为预测步骤，我们使用里程计信息更新当前状态。也就是说，我们使用赋予机器人的控制来计算机器人新位置的估计值。为了更新当前状态，我们使用以下公式：

$$\begin{bmatrix} x + \Delta t\cos\theta + q\Delta t\cos\theta \\ y + \Delta t\sin\theta + q\Delta t\sin\theta \\ \theta + \Delta\theta + q\Delta\theta \end{bmatrix}$$

在我们的简单的里程计模型中，我们可以简单地添加上一章中提到地控制：

$x + \Delta x$
$y + \Delta y$
theta + Δtheta

这应该在状态向量 X 地前三个空格中更新。我们还需要每次迭代更新 A 矩阵，即预测模型的雅各比矩阵：

1	0	$-\Delta y$
0	1	Δx
0	0	1

过程噪声矩阵 Q 也应该更新：

$c\Delta x^2$	$c\Delta x\Delta y$	$c\Delta x\Delta t$
$cc\Delta y\Delta x$	$c\Delta y^2$	$c\Delta y\Delta t$
$c\Delta t\Delta x$	$c\Delta t\Delta y$	$c\Delta t^2$

最后我们可以计算出机器人位置的新协方差矩阵。由于机器人位置的协方差只是 P 矩阵的左上角的 3*3 矩阵，因此更新如下：我们假设 Prr 矩阵是 P 矩阵的左上 3*3 矩阵。

$$P^{rr} = A\,P^{rr}\,A + Q$$

现在我们更新了机器人位置估计值以及该位置的协方差矩阵。我们还需要更新机器人以使其具有互相关性。这是协方差矩阵的前三行：

$$P^{ri} = A\,P^{ri}$$

Step 2:根据重新观察的地标更新状态

由于机器人的里程计误差，我们获得的机器人位置信息值并不完全准确。我们要补偿这些误差。这是通过使用地标来完成的。地标已经讨论过了，包含如何观察它们以及如何将他们与一直的地表信息相关联。现在，使用相关的地标，我们呢可以计算机器人的位移与我们认为的机器人位置相比较。使用位移，我们可以更新机器人位置，这是我们在步骤 2 中想要做的。此步骤针对每个重新观察的地标运行。直到步骤 3 我们才处理新的地标。由于协方差矩阵 P 和状态矩阵 X 较小，因此将新地标的合并延迟到下一个步骤将减少此步骤所需的计算成本。

我们将使用当前估计的机器人位置（x，y）来预测地标位置并保存（λx，λy）。使用以下公式：

$$\begin{bmatrix}\text{range}\\ \text{bearing}\end{bmatrix} = \begin{bmatrix}\sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2} + v_r\\ \tan^{-1}\left(\dfrac{\lambda_y - y}{\lambda_x - x}\right) - \theta + v_\theta\end{bmatrix}$$

我们得到了地标的距离和方位角 h，这在计算雅各比矩阵 H 时也可以看到。这可以与从数据关联中获得的地标的距离和方位角进行比较，我们将其表示为 z。首先我们需要更多的计算，在前面的章节中，我们有了雅各比矩阵 H：

X_r	Y_r	T_r	X_1	Y_1	X_2	Y_2	X_3	Y_3
A	B	C	0	0	-A	-B	0	0
D	E	F	0	0	-D	-E	0	0

如前所示，计算公式如下：

$$\begin{bmatrix} \frac{x-\lambda_x}{r} & \frac{y-\lambda_y}{r} & 0 \\ \frac{\lambda_y-y}{r^2} & \frac{\lambda_x-x}{r^2} & -1 \end{bmatrix}$$

同样，请记住，只有前三列和当前地标应该填写。

测量噪声矩阵 R 也应该更新以反映当前测量的位置与方位。rc 的一个很好的起始值是距离值乘以 0.01，这意味着距离误差是 0.01.bd 值最好为 1，这意味着测量误差中角度误差为 1 度。这误差不应该与角度 b 的大小成正比，这是没有意义的。

现在我们可以计算卡尔曼增益，应使用以下公式计算：

$$K = P * H^T * (H * P * H^T + V * R * V^T)^{-1}$$

卡尔曼现在包含一组数字，这些数字 y 指示了应根据重新观察的迪奥更新每个地标位置和机器人位置的程度。术语（H＊P＊HT + V＊R＊VT）称为增量协方差矩阵 S。在计算地标的数据关联一章时也使用了该术语。最后，我们可以使用卡尔曼增益来计算新的状态向量。

$$X = X + K * (z - h)$$

(z-h)不等于（0，0）的情况下，此操作将更新机器人位置以及地标位置。请注意，(z-h)得出的两个数字的结果，即距离和方位的变化量，表示为 V。

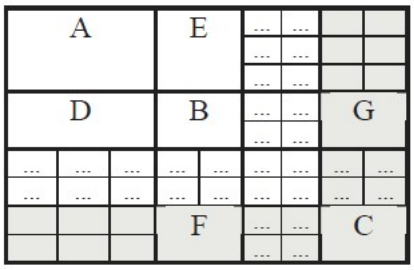
对每个匹配的地标重复此过程。

Step 3: 将新地标添加到当前状态

在这一步中，我们要用新的地标更新状态向量 X 和协方差矩阵 P。目的时拥有更多可以匹配的地标，这样的话机器人将具有更多可以匹配的地标。

首先，我们将新地标添加到状态向量 X 中：

$$X = [X\ x_N\ y_N]^T$$



另外，我们需要向协方差矩阵中添加新的行和列，如下图所示灰色区域。首先，我们在单元格 C 中添加新地标的协方差，也称为 Pn+1n+1,即第 n+1 个地标的协方差。 $P^{N+1N+1} = J_{xr} P J_{xr}^T + J_z R J_z^T$ 之后，我们添加机器人位置与新地标的协方差。这对应于协方差矩阵的左下角。

$P^{rN+1} = P^{rr} J_{xr}^T$

地标-机器人协方差是机器人-地标的协方差的转置，对应于协方差矩阵右上角。

$$P^{N+1r} = (P^{rN+1})^T$$

最后添加地标-地标协方差（最低行）：

$$P^{N+1i} = J_{xr}(P^{n})^T$$

同样的，对角阵另一侧的地标-地标协方差矩阵为其转置：

$$P^{iN+1} = (P^{N+1i})^T$$

这就完成了 slam 流程的最后一步，现在机器人已准备好再次移动，观察地标，关联地标，使用测距法更新系统状态，使用重新观察的地标更新系统状态并最终添加新地标。

12.结束语

这里介绍的 slam 是一个非常基本的 slam。有很大的改进空间，有些地方甚至没有涉及。例如，存在闭合回路问题。这个问题于机器人返回以前见过的地方有关。机器人应该识别出这一点，并使用新发现的信息来更新位置。此外，机器人更应该在机器人返回已知位置之前发现地标，使校正沿着路径传播回来。诸如 ATLAS 此类的系统与此相关。

也可以将此 slam 与占据栅格相结合，以人类可读的格式映射世界。除了将占据栅格用作人类可读的地图之外，占据栅格还可以用于路径规划。可以在此基础上构建 A*算法和 D*算法。