

正态分布变换：一个新的方法来匹配激光扫描

相关信息：

题目：The Normal Distributions Transform: A New Approach to Laser Scan Matching

来源：Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)

作者：Biber, Peter, and Wolfgang Straßer

摘要

摘要—2D范围内的扫描匹配是许多定位和建图算法的基本组件。大多数扫描匹配算法需要找到使用的，如点或线等特征之间的对应。我们为范围扫描提供了一个可选的表达，即正态分布变换。与占用栅格类似，我们将2D平面分成小的单元格。对于每一个单元格，我们分配一个正态分布，它局部地建模了测量一个点的概率。变换的结果是分段连续且可微的概率密度，我们可以使用牛顿算法将其和另一个扫描进行匹配。因此，不需要建立明确的对应关系。

我们详细介绍了该算法，并展示了其在相对位置跟踪以及同时定位和建图（SLAM）中的应用。真实数据的初步结果表明，即使不使用里程计数据，该算法也能够可靠并实时地建立未变化的室内环境地图（参见视频）。

1. 绪论

毫无疑问，同时定位与建图（SLAM）是移动机器人系统的一个基本能力。激光测距仪是获取需要输入的流行传感器，因为它们在各种各样的情况下都具有很高的可靠性以及低噪声。很多SLAM算法都是基于匹配两个距离扫描（range scans）或者将距离扫描匹配到地图上的能力。这里我们提出了一个新的方法来实现扫描匹配的低级任务，并且展示了如何用它来建图。

本文的主要目的是介绍正态分布变换（后面将用NDT表示）以及其在匹配一个扫描和另一个扫描或多个扫描面的应用。NDT将从单个扫描中重建的离散2D点集，转换成在2D平面上定义的分段连续且可微的概率密度。概率密度由一组很容易计算的正态分布组成。然后将第二次扫描与NDT的匹配定义为，第二次扫描的对齐的点，在此密度下得分和的最大值。

我们也提出了一个针对SLAM问题的，完美符合我们的匹配方案的简单算法。但是提出的匹配方案却不依赖于这一算法。因此，我们在第二章的相关工作中，只回顾了涉及到匹配两帧扫描的方法，而没有回顾像Thrun[15]或Gutmann[6]中方法一样构建整个地图的方法。无论如何，该方法的一个组件是一个感知模型，即在给定地图和位姿估计条件下的单个扫描的可能性。由于我们的方法恰好可以得到这种模型的度量，因此我们相信，我们的扫描匹配器也能够被集成到更复杂的SLAM算法中。

本文剩余部分组织如下：第三章引入NDT，第四章给出扫描匹配方法的概述，并定义将扫描与NDT比较的度量，它在第五章中使用牛顿算法进行优化。扫描匹配器在第六章中被应用于位置跟踪，并且在第七章中被应用于简单的SLAM方法中。我们最终展示了在真实数据中的一些结果，并以展望未来工作的方式总结本文。

2. 前言

匹配两个距离扫描的目的是找到进行扫描的两个位置之间的相对位姿。大多数成功的算法的基础是建立两个扫描间的**图元(primitives)**的对应关系。由此，可以推导测量误差并使其最小化。Cox使用点作为图元并将它们与在先验模型中给出的线进行匹配[3]。在Amos的工作中[7]，这些线是从扫描中提取的。Gutmann将从扫描中提取的线与模型[8]中的线进行匹配。匹配点和点这种最通用的方法是由Lu 和 Milios[9]提出的。这本质上是ICP(Iterated Closest Point, 迭代最近点)算法([1],[2],[18])应用到激光扫描匹配中的变种。我们与Lu 和 Milios使用相同的建图策略。如[10]中所提到，我们没有建立精确的地图，而是使用选择的已经恢复位姿的扫描的集合作为隐式地图。

所有的这些方法都必须建立明确的对应。我们的方法在这一点上有所不同，因为我们从不需要建立图元间的对应关系。也有一些其它避免解决对应问题的方法。在[12]中，Mojaev结合局部极坐标占用栅格的相关性和概率里程计模型来确定位姿（使用激光扫描仪和声呐）。Weiss 和 Puttkammer[17]使用角度直方图来恢复两个位姿之间的旋转。然后，使用在找到最常见的方向后计算出的x和y直方图，来恢复平移。可以通过使用第二个主方向来扩展此方法[7]。

我们的工作也被计算机视觉技术所启发。如果概率密度这个词被图像强度所代替，那么我们的方法与特征跟踪[13]或构成全景图[14]具有相似的结构，这些技术在每个相应位置使用图像梯度来估计参数。这里用的是正态分布的导数。与图像梯度相反，它们可以被解析地计算出来（即有解析解）。

3. 正态分布变换

这一章描述单帧激光扫描的正态分布变换（NDT）。这是本文的主要贡献。然后，在下面的章节中相对简单地描述NDT在位置跟踪和SLAM方面的应用。

NDT通过局部正态分布的集合来建模一帧激光扫描的所有重建2D点。首先，将机器人周围的2D空间分为整齐的固定大小的单元格（cell）。然后，对于每个至少包含3个点的单元格，进行如下操作：

1) 收集包含在这个box中的所有2D点 $\mathbf{x}_{i=1..n}$

2) 计算平均值 $\mathbf{q} = \frac{1}{n} \sum_i \mathbf{x}_i$

3) 计算协方差矩阵 $\Sigma = \frac{1}{n} \sum_i (\mathbf{x}_i - \mathbf{q})(\mathbf{x}_i - \mathbf{q})^t$

将在该单元格中测量一个样本点 \mathbf{x} 的概率建模为正态分布 $N(\mathbf{q}, \Sigma)$:

$$p(\mathbf{x}) \sim \exp\left(-\frac{(\mathbf{x}-\mathbf{q})^t \Sigma^{-1} (\mathbf{x}-\mathbf{q})}{2}\right)$$

和占用栅格类似，NDT建立平面的一个固定的细分。

可微分的描述。在我们展示例子之前，我们必须注意两个实现细节。但是占用栅格表示一个单元被占用的概率，而NDT则表示在单元格中每个位置测量一个样本(点)的概率。我们使用 $100cm \times 100cm$ 大小的单元格。

这样的表达有什么用呢？现在，我们以概率密度的形式对2D平面进行了分段连续且可微分的描述。在我们展示例子之前，我们必须注意两个实现细节。

为了尽量减少离散化的影响，我们决定使用四个重叠的栅格。即，首先放置单个边长为 l 的一个栅格，然后第二个，水平地平移 $\frac{l}{2}$ ，第三个垂直地平移 $\frac{l}{2}$ ，最终第四个，水平并垂直平移 $\frac{l}{2}$ 。现在每个2D点落入了四个单元格中。在本文的其余部分中，不会明确考虑到这一点，并且我们将描述我们的算法，就像每个点只有一个单元格。因此，如果计算出一个点的概率密度，则可以通过默认理解来完成，即评估所有四个单元格的密度并将结果相加。

第二个问题是，对于一个测量的无噪声的世界中的线，协方差矩阵会变得奇异且不可逆。实际上，协方差矩阵有时会接近奇异值。为了防止这种影响，我们检查 Σ 的较小特征值是否至少是较大特征值的0.001倍，如果不是，则将其设为该值。

图1显示了激光扫描的示例和NDT结果的可视化。通过评估在每个点的概率密度建立可视化，亮的区域意味着高概率密度。下一章展示，如何使用该变换来对齐两帧激光扫描。

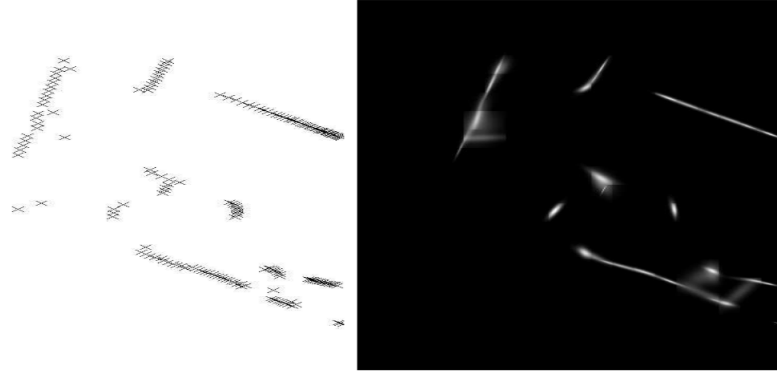


Fig. 1. An example of the NDT: The original laser scan and the resulting probability density.

4. 扫描对齐

两个机器人坐标系之间的空间映射 T 如下：

$$T: \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}, \quad (2)$$

其中 $(t_x, t_y)^t$ 和 ϕ 分别描述了两个坐标系之间的旋转和平移。扫描对齐的目标是使用在两个位置得到的激光扫描来恢复这些的参数。给定两帧扫描（第一帧和第二帧），提出的方法的大纲如下：

- 1) 创建第一帧扫描的NDT。
- 2) 初始化估计参数（用0值或者使用里程计数据）
- 3) 对于第二帧扫描中的每个样本点：根据参数将重建的2D点映射到第一次扫描的坐标系中。
- 4) 确定每个映射的点所对应的正态分布。
- 5) 通过评估每个映射的点的分布并将结果相加，来确定参数的得分。
- 6) 通过尝试优化得分，计算一个新的参数估计。这通过执行牛顿算法的一个步骤即可完成。
- 7) 回到第3步，直到满足收敛判据。

前四个步骤很简单：如上一章所描述的那样构建NDT。如上所述，里程计数据可以被用来初始化估计。使用 T 并找到对应的正态分布来完成对第二帧扫描的建图，是在NDT栅格中的一个简单查找。

现在，使用以下符号详细描述其余部分：

- $\mathbf{p} = (p_i)_{i=1..3}^t = (t_x, t_y, \phi)^t$ ：要估计的参数向量。
- \mathbf{x}_i ：在第二帧扫描坐标系下，第二帧扫描的激光扫描样本点 i 的重建2D点。
- \mathbf{x}'_i ：根据参数 \mathbf{p} 将点 \mathbf{x}_i 映射到第一帧扫描坐标系中的点，即 $\mathbf{x}'_i = T(\mathbf{x}_i, \mathbf{p})$ 。
- Σ_i, \mathbf{q}_i ：在第一帧扫描中查找的 \mathbf{x}'_i 所对应的正态分布的协方差矩阵和均值。

如果评估所有具有参数 Σ 和 \mathbf{q}_i 的点 \mathbf{x}'_i 的正态分布的和最大，则可以认为根据 \mathbf{p} 建立的地图是最优的。我们称这个和为 \mathbf{p} 的分数。它被定义为：

$$score(\mathbf{p}) = \sum \exp\left(\frac{-(\mathbf{x}'_i - \mathbf{q}_i)^t \Sigma_i^{-1} (\mathbf{x}'_i - \mathbf{q}_i)}{2}\right). \quad (3)$$

该得分在下一章被优化。

5. 使用牛顿算法优化

由于优化问题通常被描述为最小化问题，因此我们将采用这种约定的表示法。因此，在本章中，要最小化的函数是 $-score$ 。牛顿的算法迭代地找参数 $\mathbf{p} = (p_i)^t$ ，使函数 f 最小化。每次迭代求解如下等式：

$$\mathbf{H}\Delta\mathbf{p} = -\mathbf{g}$$

其中 \mathbf{g} 是输入为 $g_i = \frac{\partial f}{\partial p_i}$ 时， f 的梯度的转置， \mathbf{H} 是输入为 $H_{ij} = \frac{\partial^2 f}{\partial p_i \partial p_j}$ 时， f 的海塞矩阵。

将这个线性方程的 $\Delta\mathbf{p}$ 加到当前的估计中： $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$

如果 \mathbf{H} 是正定的， $f(\mathbf{p})$ 最初将沿 \mathbf{p} 的方向减小。如果 \mathbf{H} 不是正定的， \mathbf{H} 将被 $\mathbf{H}' = \mathbf{H} + \lambda\mathbf{I}$ 代替，选择 λ 确保 \mathbf{H}' 是正定的，关于最小化算法本身的实现细节可以在例如[4]中找到。

该算法现在被应用到函数 $-score$ 中。通过集合等式3的所有和项的偏导数来构建梯度以及海塞（矩阵）。为了使符号更短，并避免混淆参数号 i 和激光扫描样本（点） i 的索引，将不再使用样本点号的索引 i 。另外，我们有：

$$\mathbf{q} = \mathbf{x}'_i - \mathbf{q}_i \quad (8)$$

可以很容易地验证， \mathbf{q} 关于 \mathbf{p} 的偏导数等于 \mathbf{x}'_i 的偏导数。然后 $-score$ 的和 s 为：

$$s = -exp(\frac{-\mathbf{q}^t \Sigma^{-1} \mathbf{q}}{2}) \quad (9)$$

对于这样和，梯度的输入为（使用链式法则）：

$$\tilde{g}_i = -\frac{\partial s}{\partial p_i} = -\frac{\partial s}{\partial q} \frac{\partial q}{\partial p_i} = \mathbf{q}^t \Sigma^{-1} \frac{\partial \mathbf{q}}{\partial p_i} \cdot exp(\frac{-\mathbf{q}^t \Sigma^{-1} \mathbf{q}}{2}) \quad (10)$$

\mathbf{q} 关于 p_i 的偏导数由 T 的雅可比矩阵 \mathbf{J}_T 给出（参见等式2）：

$$\mathbf{J}_T = \begin{pmatrix} 1 & 0 & -x \cdot \sin\phi - y \cdot \cos\phi \\ 0 & 1 & x \cdot \cos\phi - y \sin \cdot \phi \end{pmatrix} \quad (11)$$

在海塞（矩阵） \mathbf{H} 中的一个和的输入为：

$$\tilde{H}_{ij} = -\frac{\partial^2 s}{\partial p_i \partial p_j} = -exp(\frac{-\mathbf{q}^t \Sigma^{-1} \mathbf{q}}{2}) ((-\mathbf{q}^t \Sigma^{-1} \frac{\partial \mathbf{q}}{\partial p_i})(-\mathbf{q}^t \Sigma^{-1} \frac{\partial \mathbf{q}}{\partial p_j}) + (-\mathbf{q}^t \Sigma^{-1} \frac{\partial^2 \mathbf{q}}{\partial p_i \partial p_j}) + (-\frac{\partial \mathbf{q}^t}{\partial p_j} \Sigma^{-1} \frac{\partial \mathbf{q}}{\partial p_i})) \quad (12)$$

\mathbf{q} 的二阶导数为（见等式11）：

$$\frac{\partial^2 \mathbf{q}}{\partial p_i \partial p_j} = \begin{cases} \begin{pmatrix} -x \cdot \cos\phi + y \cdot \sin\phi \\ -x \cdot \sin\phi - y \cdot \cos\phi \end{pmatrix} & i = j = 3 \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & otherwise \end{cases} \quad (13)$$

从这些等式中可以看到，建立梯度和海塞（矩阵）的计算成本较低。每个点只有一个对指数函数的调用以及少量的乘法运算。三角函数仅取决于 ϕ （当前角度估计），因此每次迭代仅需调用一次。接下来的两章将使用此算法来对SLAM进行位置跟踪。

6. 位置跟踪

这一章描述了扫描匹配算法如何被应用到，从给定的时间 $t = t_{start}$ 跟踪当前的位置。然后下一章将其扩展到SLAM中。

全局参考坐标系由在此时刻的局部机器人坐标下定义。以下将各个激光扫描称为关键帧。对此关键帧进行跟踪。在时刻 t_k ，算法执行如下步骤：

- 1) 定义 δ 为 t_{k-1} 到 t_k 时间内的移动（例如来自里程计）。
- 2) 根据 δ 建立在 t_{k-1} 时刻估计的位置的地图。
- 3) 使用当前的扫描，关键帧的NDT以及新的位置估计执行优化算法。

4) 检查该关键帧离当前的扫描是否足够“近”。如果是，迭代。否则，将最新的成功匹配的扫描作为关键帧。

扫描是否仍然足够接近的决定是基于简单的经验判据，该判据涉及关键帧和当前帧之间的平移和角度距离以及结果分数。为了帮助位置跟踪，算法必须能够实时运行：在一个1.4GHz的机器上建立一个扫描的NDT需要大约10ms。对于扫描之间很小的移动，优化算法通常需要大约1-5次迭代（很少超过10次）。一次迭代大概需要2ms，所以实时没有问题。

7. 应用到SLAM

我们将地图定义为关键帧在各自全局位姿下的集合。这一章描述了当机器人到达未知区域，如何相对于地图定位并扩展及优化地图。

A. 相对于多个扫描的定位

对于地图中的每一帧扫描 i ，它关联了一个角度 ϕ_i （或者一个旋转矩阵 \mathbf{R}_i ）和一个平移向量 $(t_x, t_y)_i^t = \mathbf{T}_i$ 。它们描述了扫描 i 在全局坐标系下的位姿。当前机器人的位姿旋转矩阵 \mathbf{R} 和平移向量 \mathbf{T} 表示：

$$T' : \begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{R}_i^t (\mathbf{R} \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{T} - \mathbf{T}_i) \quad (14)$$

仅需要很小的变化就可以使第5章的算法适用这种情况。扫描 i 的2D点的建图现在通过应用 T' 来计算。而且， T' 的雅可比(矩阵)和二阶导数现在变得稍微复杂一点。建图的雅可比(矩阵)如下：

$$\mathbf{J}_{T'} = \mathbf{R}_i^t \mathbf{J}_T \quad (15)$$

T' 的二阶导数如下：

$$\frac{\partial^2 \mathbf{x}'}{\partial p_i \partial p_j} = \begin{cases} \mathbf{R}_i^t \begin{pmatrix} -x \cdot \cos \phi + y \cdot \sin \phi \\ -x \cdot \sin \phi - y \cdot \cos \phi \end{pmatrix} & i = j = 3 \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & otherwise \end{cases} \quad (16)$$

通过汇总所有重叠的扫描可以求得用于优化算法的梯度和海塞（矩阵）。但是我们找到了另一种选择，它更快且能产生同样好效果：对于在机器人位置获得的扫描的每一个样本点，确定评估概率密度的结果最大的扫描。只有该扫描被用于此样本和当前的迭代，这样，建立优化算法的梯度和海塞（矩阵）所需的操作，与重叠关键帧的数量无关，除了找到提到的最大值。

B. 加入新的关键帧并优化地图

在每一时刻，地图由各自全局坐标系位姿下的关键帧组成。如果当前扫描与地图的重叠太小，那么通过最近的成功匹配的扫描来扩展地图。然后，将每个重叠的扫描分别匹配到新的关键帧，产生两次扫描之间的相对位姿。维护一个图，其中包含成对匹配结果的信息。

在图中，每一个关键帧由一个节点来表示。一个节点保存了在全局坐标系中关键帧位姿的估计值。两个节点之间的边表示相应的扫描已成对匹配，并保持两个扫描之间的相对位姿。

在一个新的关键帧加入后，通过优化基于所有关键帧的参数定义的误差函数，来精化地图。成对配准的结果用于为每个匹配对定义二次误差模型，如下所示：两次扫描的全局参数还定义两次扫描之间的相对姿态。令 Δp 为全局参数定义的相对位姿与成对匹配结果定义的相对位姿之间的差。然后，我们对 Δp 的函数，两次扫描的分数进行建模，使用二次模型：

$$score'(\Delta p) = score + \frac{1}{2}(\Delta p)^t \mathbf{H}(\Delta p) \quad (17)$$

因此， $score$ 是最终稿的得分，当成对的匹配收敛， \mathbf{H} 是由此获得的海塞（矩阵）。该模型可通过对 $score$ 在 $\Delta p = \mathbf{0}$ 附件进行二阶泰勒展开得到。注意，缺失了线性项，因为我们扩展了一个极端点。现在，基于所有的边对分数求和并优化 g 。

如果关键帧的数量变大，则无法在实时条件下执行此最小化操作（空闲参数的数量为 $3N-3$ ，其中 N 为关键帧的数量）。我们因此只优化地图的子图。通过收集所有关键帧来构建该子图，可以通过遍历不超过三个边来从新关键帧的节点到达此关键帧。现在，我们仅针对参数（属于此子图中包含的关键帧）对上面的误差函数进行优化。当然，如果产生了闭环，我们必须在所有关键帧上进行优化。

8. 结果

我们现在提供的结果（以及第6节中的示例）是在不使用里程计的情况下得到的。这应该证明了该方法的鲁棒性。当然，如Thrun在[15]中所提到的，只要任何2D结构存在于世界上，这都是可能的。

在图2中建立的地图是通过控制机器人走出实验室，顺着走廊，逆着走廊，然后返回实验室得到的。

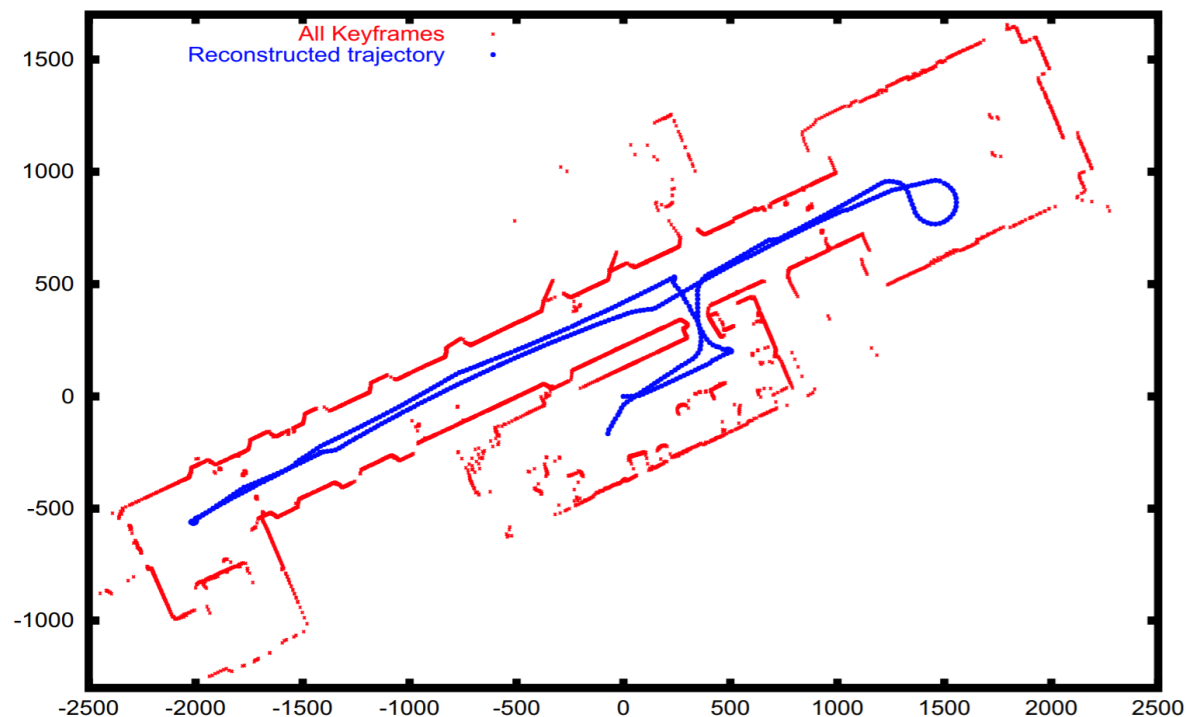


Fig. 2: A map, that was built using our scan matcher. Lengths are given in cm. Shown are the set of keyframes and the estimated trajectory (see video). More videos can be found on the authors' homepage [11].

因此，这种情况既需要扩展地图，又需要相对于地图的定位。机器人在20分钟运动了大约83米，采集了28430帧激光扫描。使用SICK激光获取分辨率为 1° ，角度为 180° 的扫描。仿真的速度大约 35cm/s ，并且每秒大约23帧扫描。我们使用组合策略来创建该地图。第4章的位置跟踪器被应用到每一个扫描中，而我们通过线性地推断最新时刻的结果来初始化参数【即以匀速模型预测下一时刻位姿】。每第十帧扫描，就会执行一次第七章的过程。

图2展示了建图的结果。其最终显示的地图由33个关键帧组成。仔细观察也可以发现，我们的扫描匹配算法可以容忍环境中的小变化，例如开门或关门。在 1.4GHz 的机器上，离线处理所有的帧需要58秒，即每秒（处理）97帧扫描。通过将我们当前的实现从Java移植到更快的语言，也许可以提高（处理）速度。

9. 结论以及将来的工作

我们已经提出了一个距离扫描的新表达，即正态分布变换（NDT）。可使用该变换来推导匹配另一个扫描的解析表达式。我们也展示了如何将我们的扫描匹配器应用于位置跟踪问题和SLAM问题。我们方法的主要优点是：

- 点或特征之间无需建立明确的对应关系。因为这是在大部分方法中最容易出错的部分，没有对应，我们（的方法）更稳定。
- 可以解析地计算出所有导数，这样快且正确。

当然，也存在问题：一切都能够通过局部正态分布进行充分地建模吗？

迄今为止，我们在室内环境的测试表现很好，这从来没有问题。我们计划在结构较少的环境中，最好是在户外，进行进一步测试。我们还打算系统地比较我们的方法与Lu和Milios方法的收敛半径。

10. 致谢

Peter Biber由Land BadenWuerttemberg赠款资助，对此深表谢意。我们还要感谢Andreas Zell教授和Achim Lilienthal，（我们才有可能）使用机器人和数据采集软件。我们还要感谢Sven Fleck进行了许多富有成果的讨论，并为视频提供了帮助。

11. 参考文献