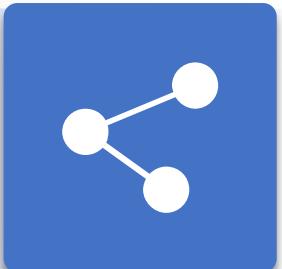




Distributed Ranging SLAM for Multiple Robots with Ultra-WideBand and Odometry Measurements

Ran Liu, Zhongyuan Deng, Zhiqiang Cao, and et al.
IROS 2022

智能网络与优化实验室





1

Introduction



介绍

大多数的多机器人 SLAM 系统是基于集中式 (Centralized) 或分布式 (Distributed) 的结构：

- 集中式：在一个服务器上处理所有收集到的测量数据，随着机器人数的增加，计算开销呈指数增长。
- 分布式：在可扩展性和计算效率上优于集中式解决方案，计算被分配在一组机器人之间进行。

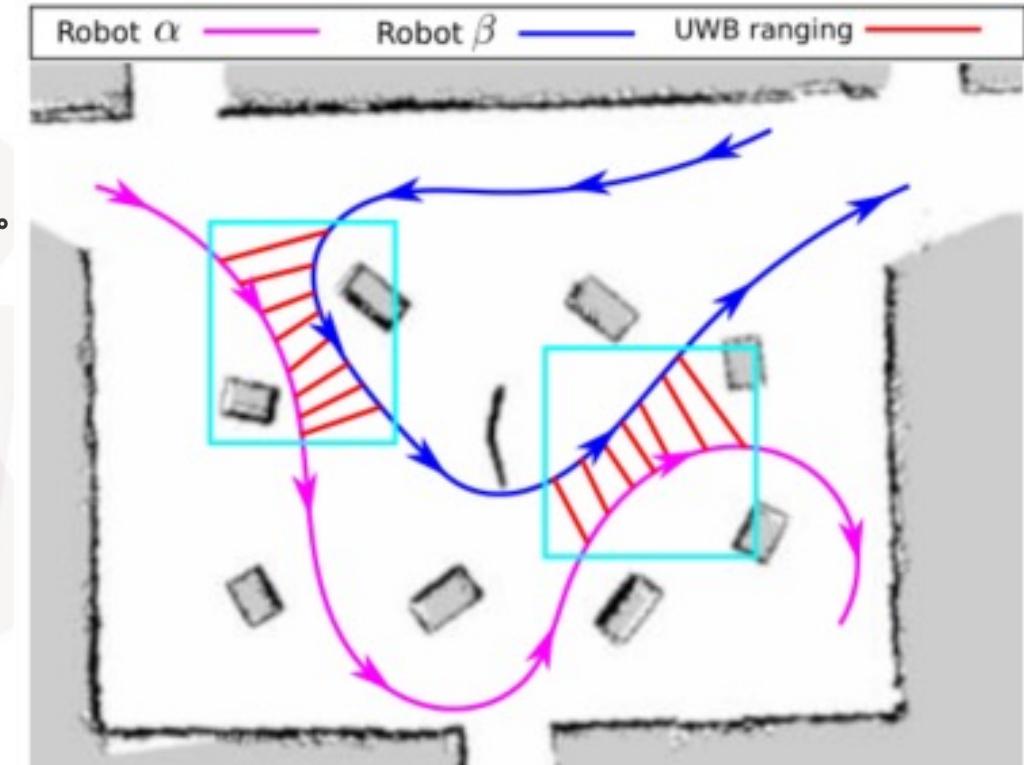
UWB 提供了精确的距离，但由于缺乏方位信息，难以应用于机器人导航。传统的基于 AoA (Angle of Arrival) 的解决方案需要建立一个繁琐的相移估计。本文专注于使用纯 UWB 测距和里程计在机器人集群的姿态估计，而提出的解决方案目的是能在 non-line-of-sight (NLOS) 条件下工作。



贡献

本文提出了一个分布式 SLAM 用于基于 UWB 测距和里程测量的机器人集群轨迹估计，每个机器人配备一个 UWB 节点用于测距。

- 使用短期 UWB 测距和里程测量估计机器人之间的姿态（位置和方向）。特别地，通过最小化两个机器人靠近时在不同位置的测距测量来实现分布式位姿估计。
- UWB 很容易受到 NLOS 传播的影响，从而导致错误的环路闭合，特别是当机器人所走路径是不可预测的。为了消除错误的闭环，本文利用分布式的成对一致性最大化（Pairwise Consistency Maximization，PCM）算法来验证闭环对的质量，并执行 outlier rejection。
- 最后，通过分布式位姿图优化（DPGO）将内环闭包与测程方法结合，估计机器人的运动轨迹。





2

Methodology



多机器人分布式测距 SLAM (Distributed Ranging SLAM for Multiple Robots)

本文的分布式 SLAM 主要包含以下三个步骤：

1. 每个机器人进行分布式位姿估计，利用短时 UWB 测距和里程测量确定其同伴机器人（peer robot）的位姿。
2. 每个机器人运行一个分布式异常抑制模块（distributed outlier rejection module），过滤出可能存在错误的估计。
3. 采用分布式 SLAM 框架融合内参和里程信息，在一个通用参考系中生成轨迹。

假设机器人集群由 N 个机器人构成：记机器人 α ($1 \leq \alpha \leq N$) 在时间 t 的位姿为 $\mathbf{x}_\alpha^t = [\mathbf{R}_\alpha^t, \mathbf{t}_\alpha^t] \in SE(2)$ ，其中 $\mathbf{R}_\alpha^t \in SO(2)$ (对应的旋转角为 θ_α^t) 且 $\mathbf{t}_\alpha^t = [x_\alpha^t, y_\alpha^t]^T \in \mathbb{R}^2$ ；记 $\mathbf{x}_\alpha^{k,t}$ 为机器人 α 从时间 k 到时间 t 的相对位姿。记 $\mathbf{r}_\alpha^t = \{r_{\alpha,\beta}^t\}$ ($\beta \in N_\alpha^t$) 为机器人 α 在时间 t 经 UWB 测得的距离， N_α^t 为 α 的邻居， $r_{\alpha,\beta}^t$ 为 α 和 β 在时间 t 的 UWB 测距。

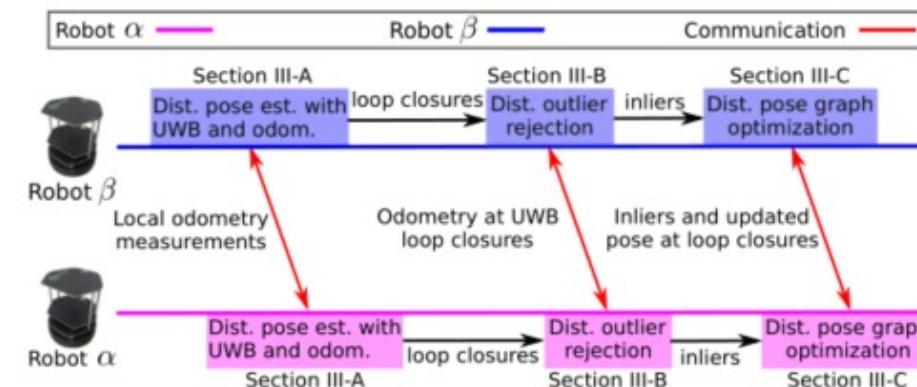


Fig. 2. Overview of the modules and their communication requirements for our distributed ranging SLAM with two robots.



使用短期 UWB 测距和里程测量的分布式位姿估计 (Distributed Pose Estimation with Short-term UWB and Odometry Measurements)

β 在 t 时刻相对 α 的相对位姿估计值为 $\dot{\mathbf{x}}_{\alpha,\beta}^t$ 可以通过最小化在滑动时间窗 τ 内不同位置采集的 UWB 测距残差来获得：

$$\dot{\mathbf{x}}_{\alpha,\beta}^t = \underset{\mathbf{x}_{\alpha,\beta}^t}{\operatorname{argmin}} \sum_{i=t-\tau}^t e(\mathbf{x}_{\alpha,\beta}^t, \mathbf{x}_\alpha^t, \mathbf{x}_\beta^t, \mathbf{x}_\alpha^i, \mathbf{x}_\beta^i, r_{\alpha,\beta}^i) = \underset{\mathbf{x}_{\alpha,\beta}^t}{\operatorname{argmin}} \sum_{i=t-\tau}^t \left(r_{\alpha,\beta}^i - \underbrace{\left\| (\mathbf{x}_\alpha^t)^{-1} \cdot \mathbf{x}_\alpha^i, \mathbf{x}_{\alpha,\beta}^t \cdot (\mathbf{x}_\beta^t)^{-1} \cdot \mathbf{x}_\beta^i \right\|}_{\text{Odom. of } \alpha \text{ and Odom. of } \beta} \right)^2$$

$\|\cdot\|$ 计算两个位置之间的欧氏距离， τ 表示位姿估计的滑动窗口大小。

在给定好的初值 \mathbf{x} 的前提下，上式可以使用 g2o、gtsam、Ceres 等工具进行计算。但是，由于前面目标方程的非凸性，没有好的初始值难以保证搜索到全局最优解。



使用短期 UWB 测距和里程测量的分布式位姿估计 (Distributed Pose Estimation with Short-term UWB and Odometry Measurements)

可以使用一种粗搜索策略来寻找一个适当的近似相对位姿，作为目标函数优化的初始值。

由于可以认为 UWB 测量是精确的， β 将被大致定位在以 α 为圆心， $r_{\alpha,\beta}^t$ 为半径的圆上（通过 UWB 测距得到），在极坐标下， β 对应的角度为 ϕ （待估计）。因此，相比在整个 2D 空间进行搜索，可以在已知 UWB 测距的前提下，在极坐标空间内对应的角度 ϕ 在 $-\pi$ 到 π 之间进行搜索。 θ （相对旋转）和 ϕ （极坐标夹角）的角搜索步长记为 δ ，搜索步数量为 $w_\phi = w_\theta = \lceil \frac{\pi}{\delta} \rceil$ 。

定义整个搜索空间为 $\bar{w} = \{-w_\phi, w_\phi\} \times \{-w_\theta, w_\theta\}$ 。由此，目标函数的优化转化为在给定相对位姿 $\hat{x}_{\alpha,\beta}^t \in \mathbf{w}$ 的前提下最小化残差，其中：

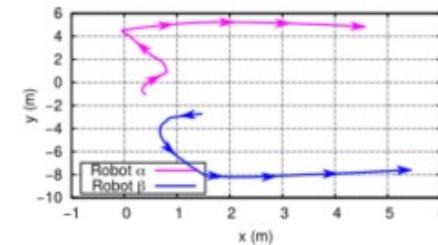
$$\mathbf{w} = \left\{ \left(\underbrace{r_{\alpha,\beta}^t \cos(\delta i_\theta)}_{\text{Position } x}, \underbrace{r_{\alpha,\beta}^t \sin(\delta i_\theta)}_{\text{Position } y}, \delta i_\theta \right), | (i_\phi, i_\theta) \in \bar{w} \right\}$$

粗搜解 $\hat{x}_{\alpha,\beta}^t$ 则可以带入原目标函数作为使用 g2o 求解的初始值。

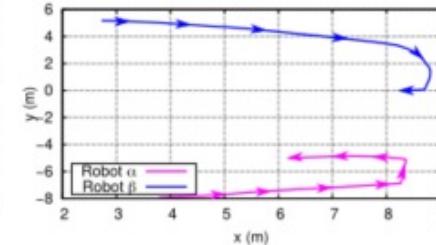


使用短期 UWB 测距和里程测量的分布式位姿估计 (Distributed Pose Estimation with Short-term UWB and Odometry Measurements)

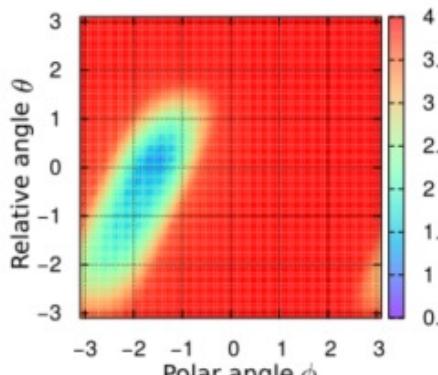
前述搜索策略归纳为右图中的算法。下图的误差热度分布也说明残差函数呈多峰分布，因此随机初始值很容易进入局部最优解。



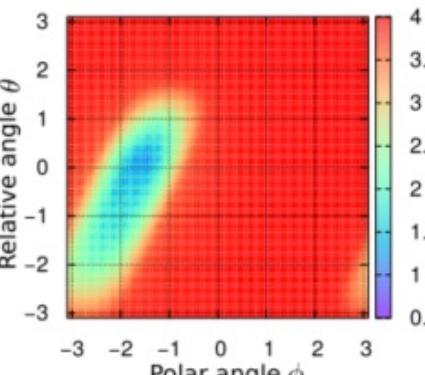
(a) Odom. trajectories for Case I



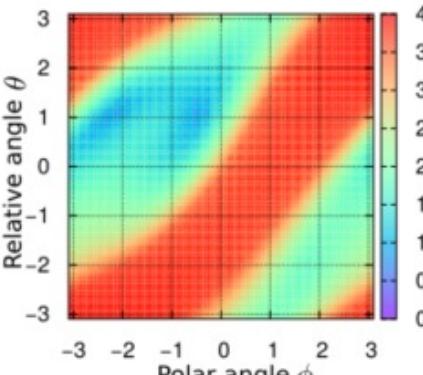
(b) Odom. trajectories for Case II



2023/5/



(c) Mean residual for Case I



(d) Mean residual for Case II

Algorithm 1: Find coarse relative pose $\tilde{\mathbf{x}}_{\alpha,\beta}^t$ between robot α and robot β with short-term UWB and odom.

```

Data: Odometry from robot  $\alpha$ :  $\mathbf{x}_\alpha^i$ , odometry from robot  $\beta$ :  $\mathbf{x}_\beta^i$ , UWB ranging between  $\alpha$  and  $\beta$ :  $\mathbf{r}_{\alpha,\beta}^i$ , where  $t - \tau \leq i \leq t$ 
Result: Coarse pose estimation  $\tilde{\mathbf{x}}_{\alpha,\beta}^t$ 
// Record the minimum residual
1  $minimumResidual \leftarrow \infty$ 
2 for  $i_\phi \leftarrow -w_\phi$  to  $w_\phi$  do
3   for  $i_\theta \leftarrow -w_\theta$  to  $w_\theta$  do
4      $\mathbf{x}'_{\alpha,\beta} = < r_{\alpha,\beta}^t \cos(\delta i_\phi), r_{\alpha,\beta}^t \sin(\delta i_\phi), \delta i_\theta >$ 
        // Initialize the residual to be 0
5      $r \leftarrow 0$ 
6     for  $i \leftarrow t - \tau$  to  $t$  do
        // Residual with Equation 1
5        $r = r + \mathbf{e}(\mathbf{x}'_{\alpha,\beta}, \mathbf{x}_\alpha^i, \mathbf{x}_\beta^i, \mathbf{x}_\alpha^i, \mathbf{x}_\beta^i, \mathbf{r}_{\alpha,\beta}^i)$ 
        // Bound residual error for fast computation
6     if  $r \geq minimumResidual$  then
7       | break
8     end
9   end
10  if  $r < minimumResidual$  then
11    |  $minimumResidual \leftarrow r$ 
12    |  $\tilde{\mathbf{x}}_{\alpha,\beta}^t = < x'_{\alpha,\beta}, y'_{\alpha,\beta}, \theta'_{\alpha,\beta} >$ 
13  end
14 end
15 end
16 end
17 end
18 return  $\tilde{\mathbf{x}}_{\alpha,\beta}^t$ 
```



分布式异常抑制 (Distributed Outlier Rejection)

位姿估计的精度极大依赖于 UWB 测距的精度以及机器人的路径。实际应用中，这些因素是不可控的，尤其是存在 NIOS 的未知环境下。因此，本文使用了现有的异常抑制算法 PCM (Pairwise Consistency Maximization) 来处理可能存在异常的位姿估计（即回环）。

给定 α 和 β 间的两个回环 $\bar{\mathbf{x}}_{\alpha,\beta}^k$ 和 $\bar{\mathbf{x}}_{\alpha,\beta}^i$ ，通过下式来确定两个回环的成对一致性 (pairwise consistency) :

$$\left\| \mathbf{x}_{\alpha}^{k,i} \oplus \bar{\mathbf{x}}_{\alpha,\beta}^i \oplus \mathbf{x}_{\beta}^{i,k} \ominus \bar{\mathbf{x}}_{\alpha,\beta}^k \right\|_{\Sigma} \leq \chi_{\epsilon}^2$$

$\|\cdot\|_{\Sigma}$ 表示马氏距离 (Mahalanobis distance)， \oplus 和 \ominus 表示两个位姿的 composition 和 inversion 操作[1]。如图为一致性检测的示意图。在理想条件下，所有的位姿估计不存在噪声，由两个里程 (图中绿线) 和两个回环 (图中青线) 组成的位姿循环能被识别。上式的目的就是计算沿这个循环累积的误差是否与噪声协方差 Σ 一致。 χ_{ϵ}^2 为卡方分布置信度 ϵ 对应的分位点。

[1] R. Smith, M. Self, and P . Cheeseman, "Estimating uncertain spatial relationships in robotics," in 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, USA, March 31-April 3 1987, pp. 850–850.



分布式异常抑制 (Distributed Outlier Rejection)

$$\left\| \mathbf{x}_\alpha^{k,i} \oplus \bar{\mathbf{x}}_{\alpha,\beta}^i \oplus \mathbf{x}_\beta^{i,k} \ominus \bar{\mathbf{x}}_{\alpha,\beta}^k \right\|_\Sigma \leq \chi_\epsilon^2$$

在理想条件下，所有的位姿估计不存在噪声，由两个里程（图中绿线）和两个回环（图中青线）组成的位姿循环能被识别。上式的目的就是计算沿这个循环累积的误差是否与噪声协方差 Σ 一致。 χ_ϵ^2 为卡方分布置信度 ϵ 对应的分位点。

经过上式的一致性检测，PCM 构建一个邻接矩阵来记录回环的成对一致性。机器人从矩阵中计算最大团，得到最大的 pairwise consistent 回环集合 (inliers)。为了减少实时操作的计算成本，使用 PCM[1] 的一个高效版本，它用启发式搜索策略增量更新 pairwise consistent 回环，以找到最大团。这些 inliers 将会被传入位姿图进行优化。

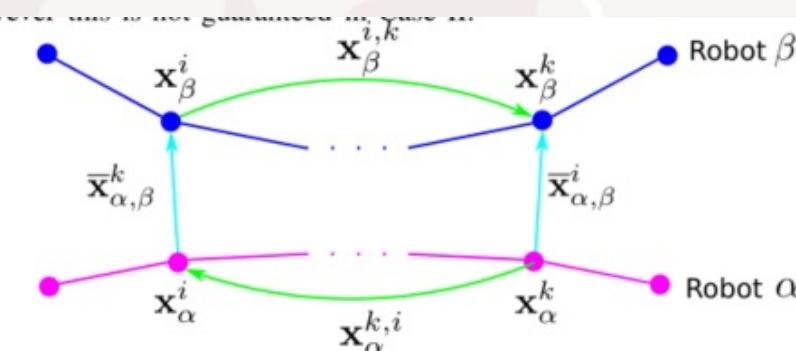


Fig. 4. Illustration of the distributed PCM consistency check performed by robot α . Robot α estimates the pose of robot β at time k and time i using distributed pose estimation described in Section III-A. The estimated pose and odometry are passed for consistency check.

[1] Y . Tian, Y . Chang, F. H. Arias, C. Nieto-Granda, J. P . How, and L. Carlone, “*Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems*,” IEEE Transactions on Robotics, pp. 1–17, 2022.



分布式位姿图优化 (Distributed Pose Graph Optimization)

分布式 SLAM 中，每个机器人基于局部信息进行位姿图优化。机器人交换各自的位姿估计，迭代求解全局优化问题。使用 [1] 中的方法，每个机器人运行一个 DPGO 模块，基于里程测量和 inlier 回环进行轨迹优化。DPGO 采用黎曼梯度下降法进行局部优化。它以分布式的方式解决位姿图优化的秩限制松弛问题。在更新过程中，算法只需要交换连接回环的位姿，避免交换所有位姿信息，从而节省了大量的通信资源。

[1] Y . Tian, K. Khosoussi, D. M. Rosen, and J. P . How, " *Distributed certifiably correct pose-graph optimization*," IEEE Transactions on Robotics, vol. 37, no. 6, pp. 2137–2156, 2021.



3

Experiment



实验设置

- UWB 采样速率 : 50Hz
- 实验系统 : ROS
- Hokuyo LiDAR
- 评估 : MSE , 平移误差单位 : m ; 旋转误差单位 : °



实验结果

1. 时间窗 τ 大小的影响

τ	NLS optimization [27]			Coarse search			Proposed approach (coarse search+opt.)		
	Trans. error	Rot. error	Time	Trans. error	Rot. error	Time	Trans. error	Rot. error	Time
$\tau = 10$	8.19 ± 6.85	70.71 ± 53.85	2ms	5.03 ± 5.19	54.56 ± 50.74	4ms	4.54 ± 5.24	49.34 ± 50.61	6ms
$\tau = 25$	7.47 ± 6.89	63.48 ± 56.47	4ms	2.89 ± 3.38	31.82 ± 41.97	5ms	2.42 ± 3.37	27.08 ± 41.97	9ms
$\tau = 50$	6.12 ± 6.43	54.11 ± 56.25	6ms	1.88 ± 2.21	15.58 ± 23.39	7ms	1.38 ± 2.01	12.66 ± 24.15	13ms
$\tau = 100$	4.53 ± 5.74	37.16 ± 48.49	10ms	1.31 ± 1.45	9.23 ± 11.07	9ms	0.83 ± 1.17	6.74 ± 11.49	19ms
$\tau = 200$	3.97 ± 5.49	32.56 ± 46.67	15ms	1.25 ± 1.49	7.90 ± 9.67	14ms	0.82 ± 1.24	6.29 ± 11.09	29ms

实验结果

2. 角度搜索步长 δ 的影响

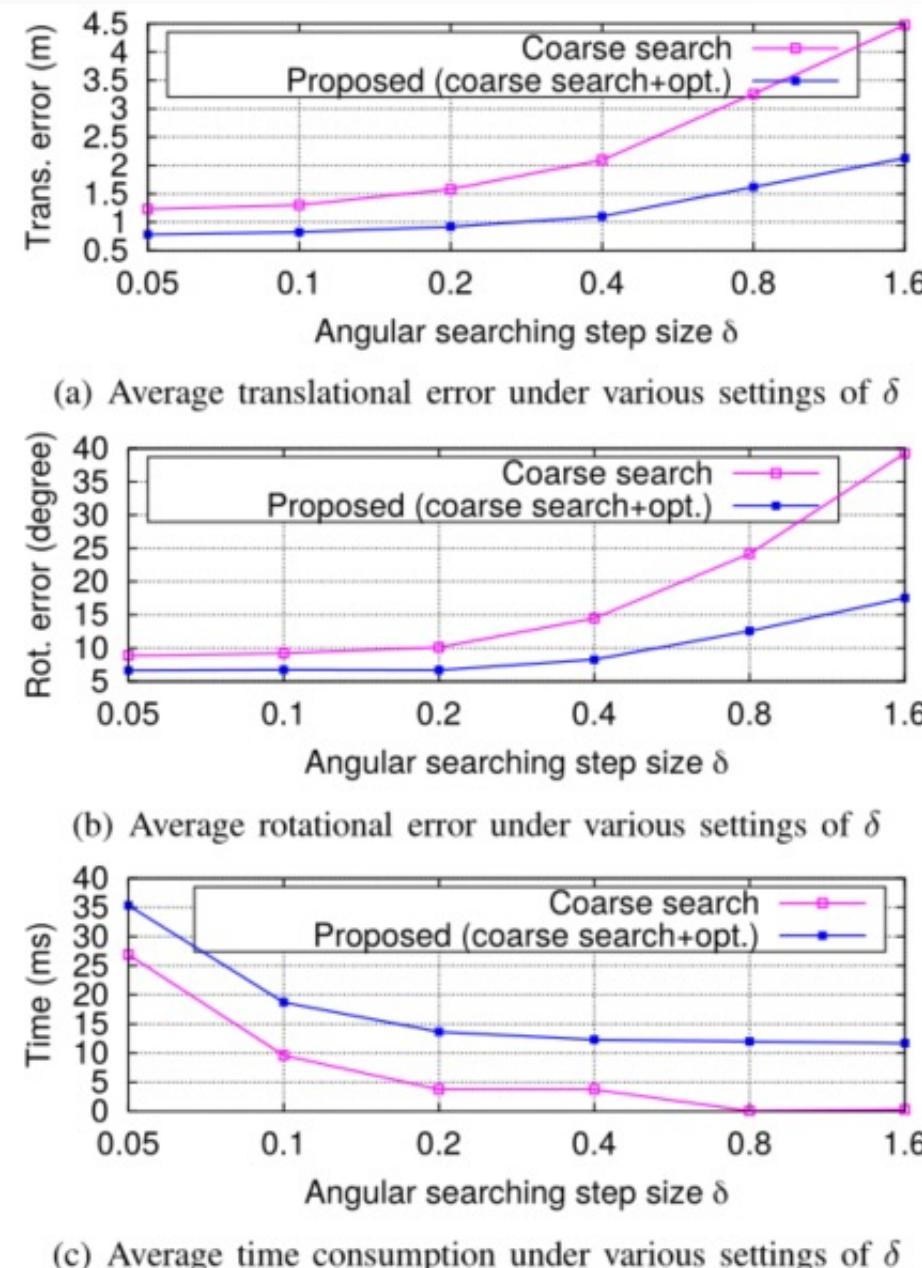


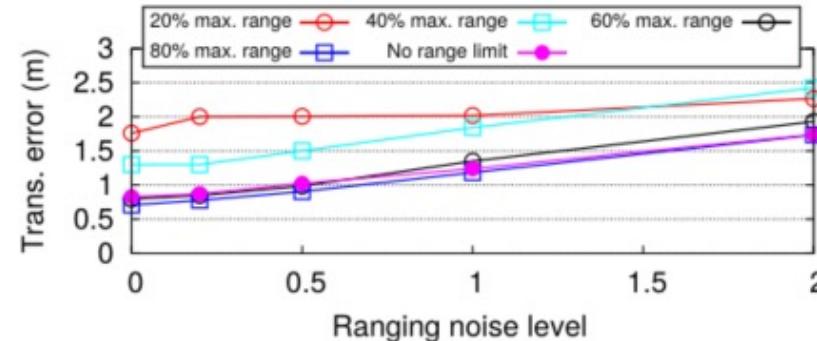
Fig. 6. Relative pose estimation accuracy and time consumption under different angular searching step sizes δ .



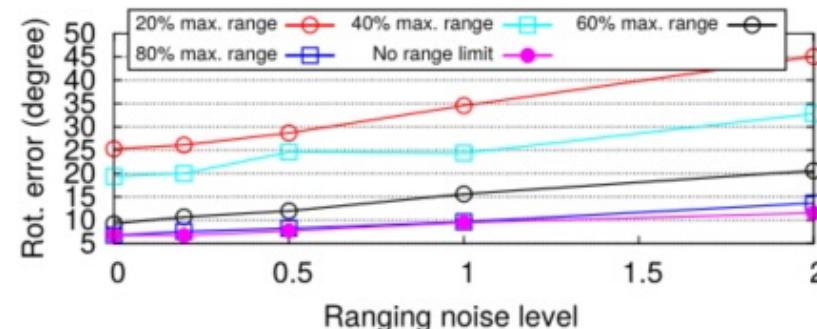
实验结果

3. UWB 和里程计设置的影响

设置 $\delta = 0.1$, $\tau = 100$, 对比限制不同 UWB 最大测距范围下的精度, 同时向 UWB 数据中加入高斯噪声。



(a) Average translational error under various ranging setting and ranging noise



(b) Average rotational error under various ranging setting and ranging noise



实验结果

4. PCM 异常抑制效果

设置 $\delta = 0.1$ 。

τ	Distributed PCM consistency check				Distributed pose graph optimization	
	ϵ in PCM	No. of inliers	Trans. error	Rot. error	Trans. error	Rot. error
$\tau=10$	Without PCM	999	4.54 ± 5.24	49.34 ± 50.61	2.13 ± 1.12	8.72 ± 5.27
	$\epsilon=0.01$	391	1.57 ± 1.31	12.15 ± 9.98	0.78 ± 0.45	6.08 ± 3.60
	$\epsilon=0.05$	269	1.37 ± 1.07	9.59 ± 8.35	0.79 ± 0.43	6.52 ± 4.07
	$\epsilon=0.1$	212	1.12 ± 0.82	8.09 ± 6.22	0.61 ± 0.36	5.19 ± 3.18
	$\epsilon=0.2$	152	0.97 ± 0.82	6.47 ± 4.54	0.71 ± 0.43	5.69 ± 3.56
	$\epsilon=0.5$	65	0.57 ± 0.42	5.68 ± 3.40	0.53 ± 0.36	5.18 ± 3.38
	$\epsilon=0.8$	25	0.73 ± 0.72	7.97 ± 6.02	0.69 ± 0.38	6.60 ± 3.96
$\tau=50$	Without PCM	999	1.38 ± 2.01	12.66 ± 24.15	0.74 ± 0.46	5.42 ± 4.05
	$\epsilon=0.01$	904	0.95 ± 0.91	6.93 ± 6.75	0.63 ± 0.42	4.78 ± 3.50
	$\epsilon=0.05$	863	0.90 ± 0.83	6.14 ± 5.16	0.60 ± 0.37	4.29 ± 3.07
	$\epsilon=0.1$	782	0.76 ± 0.59	5.78 ± 4.30	0.52 ± 0.33	4.43 ± 3.00
	$\epsilon=0.2$	663	0.64 ± 0.42	5.25 ± 3.71	0.49 ± 0.32	4.18 ± 2.83
	$\epsilon=0.5$	394	0.51 ± 0.31	4.52 ± 2.99	0.45 ± 0.30	4.01 ± 2.81
	$\epsilon=0.8$	131	0.53 ± 0.36	4.35 ± 3.78	0.52 ± 0.26	4.23 ± 2.75
$\tau=100$	Without PCM	999	0.83 ± 1.17	6.74 ± 11.49	0.60 ± 0.41	5.05 ± 3.56
	$\epsilon=0.01$	987	0.75 ± 0.76	5.88 ± 4.57	0.59 ± 0.40	4.96 ± 3.44
	$\epsilon=0.05$	974	0.74 ± 0.74	5.71 ± 4.15	0.57 ± 0.37	4.88 ± 3.34
	$\epsilon=0.1$	929	0.69 ± 0.63	5.42 ± 3.71	0.53 ± 0.34	4.71 ± 3.22
	$\epsilon=0.2$	736	0.64 ± 0.49	5.19 ± 3.48	0.53 ± 0.34	4.69 ± 3.29
	$\epsilon=0.5$	702	0.51 ± 0.34	5.47 ± 3.42	0.48 ± 0.32	4.92 ± 3.32
	$\epsilon=0.8$	252	0.54 ± 0.36	5.08 ± 3.10	0.49 ± 0.31	4.89 ± 3.19



4

References



参考

- AoA 算法 : https://blog.csdn.net/qq_37930244/article/details/104933168



5

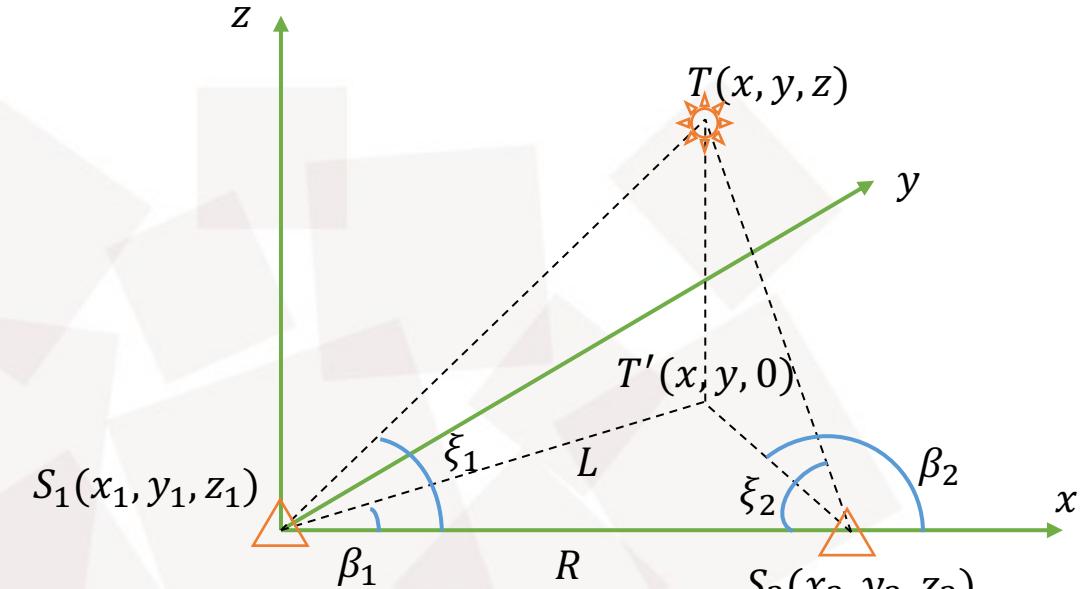
Appendix



AoA (Angle of Arrival)

如图，设有基站 $S_1(x_1, y_1, z_1)$ 和 $S_2(x_2, y_2, z_2)$ ，目标 $T(x, y, z)$ 。关键角可以由基站坐标和目标坐标给出：

$$\begin{cases} \tan \beta_1 = \frac{y - y_1}{x - x_1}, & \text{相对于 } S_1 \text{ 的方位角} \\ \tan \beta_2 = \frac{y - y_2}{x - x_2}, & \text{相对于 } S_2 \text{ 的方位角} \\ \tan \xi_1 = \frac{z - z_1}{\sqrt{(x - x_1)^2 + (y - y_1)^2}}, & \text{相对于 } S_1 \text{ 的俯仰角} \\ \tan \xi_2 = \frac{z - z_2}{\sqrt{(x - x_2)^2 + (y - y_2)^2}}, & \text{相对于 } S_2 \text{ 的俯仰角} \end{cases}$$



$$\begin{aligned} \beta_1 &= \angle T'S_1 x^+ \\ \beta_2 &= \angle T'S_2 x^+ \\ \xi_1 &= \angle TS_1 T' \\ \xi_2 &= \angle TS_2 T' \\ x^+ &\text{ 表示 } x \text{ 轴正方向} \end{aligned}$$



AoA (Angle of Arrival)

由这四个角结合基站坐标即可反推出目标位置，推导如下：

Step 1：两基站之间的距离

$$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Step 2：在 $\Delta S_1 T' S_2$ 中，由正弦定理得

$$\frac{\sin(\beta_2 - \beta_1)}{L} = \frac{\sin(\pi - \beta_2)}{R} = \frac{\sin \beta_2}{R} \Rightarrow R = \frac{L \sin \beta_2}{\sin(\beta_2 - \beta_1)}$$

Step 3：在 $\Delta T T' S_1$ 中， T 到 S_1 的距离为

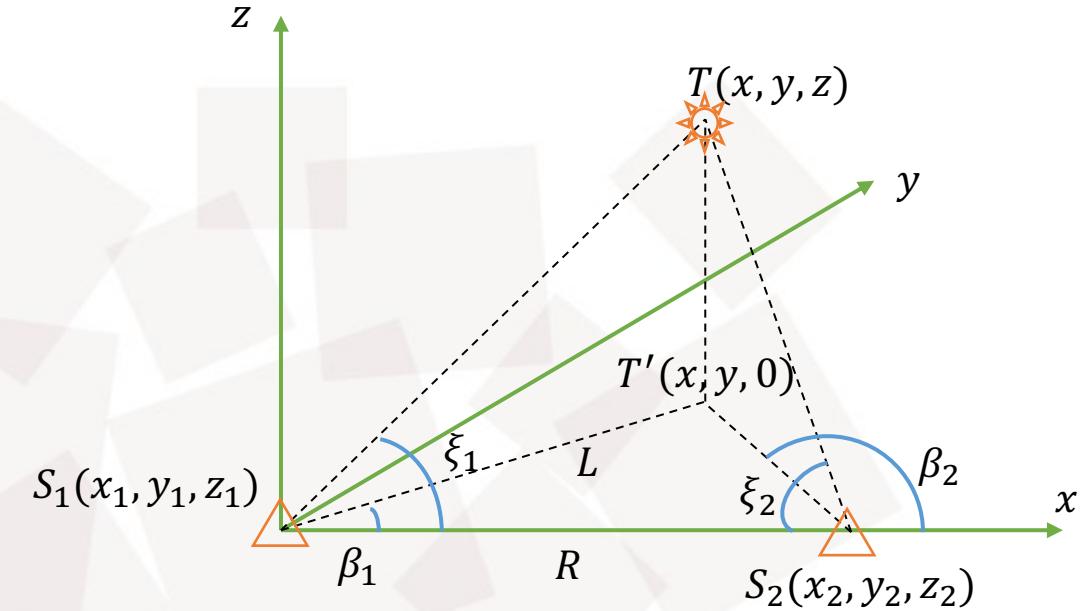
$$R' = \frac{R}{\cos \xi_1}$$

Step 4：可得

$$x = R' \cos \xi_1 \cos \beta_1 + x_1$$

$$y = R' \cos \xi_1 \sin \beta_1 + y_1$$

$$z = R' \sin \xi_1 + z_1$$





位姿的 Composition 和 Inversion 操作

记机器人 α 和 β 的位姿为 $\mathbf{x}_\alpha = [x_\alpha, y_\alpha, \theta_\alpha]^T$ 和 $\mathbf{x}_\beta = [x_\beta, y_\beta, \theta_\beta]^T$ ，则两个机器人位姿的 composition 操作为：

$$\mathbf{x}_\alpha \oplus \mathbf{x}_\beta = \begin{bmatrix} x_\beta \cos \theta_\alpha - y_\beta \sin \theta_\alpha + x_\alpha \\ x_\beta \sin \theta_\alpha + y_\beta \cos \theta_\alpha + y_\alpha \\ \theta_\alpha + \theta_\beta \end{bmatrix}$$

单个机器人位姿的 inversion 操作为：

$$\ominus \mathbf{x}_\alpha = \begin{bmatrix} -x_\alpha \cos \theta_\alpha - y_\alpha \sin \theta_\alpha \\ x_\alpha \sin \theta_\alpha - y_\alpha \cos \theta_\alpha \\ -\theta_\alpha \end{bmatrix}$$

两个机器人位姿的 inversion 操作为：

$$\mathbf{x}_\alpha \ominus \mathbf{x}_\beta = \mathbf{x}_\alpha \oplus (\ominus \mathbf{x}_\beta)$$



謝謝

Thank You

THANKS

