

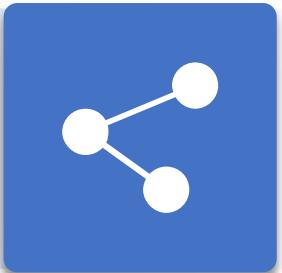


Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3D Point Cloud Map

Giseop Kim and Ayoung Kim

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018

智能网络与优化实验室





1

介绍



介绍

位置识别 (Place Recognition) 对于 SLAM 中的回环检测起到比较重要的作用。基于 LiDAR 的位置识别方案有以下几种：

- (1) 比较早期的方案有：从计算机视觉中的 3D 模型移植来的基于传统的局部关键点描述子进行位置识别。
- (2) 而目前主要的工作集中在怎么在点云的结构化信息中构建局部和全局的描述子。

作者认为方案 (2) 有两个问题需要克服：

- 描述子需要有旋转不变性
- 由于点云中的分布在距离和法向量都不均匀，因此空间描述子需要进行噪声处理

目前为了解决上面的问题，有很多基于直方图的方法。但是作者认为直方图类方法只提供了对场景的随机指标，对场景中的结构细节描述并不直观。

参考：

- <https://zhuanlan.zhihu.com/p/357868922>
- <http://xchu.net/2020/09/07/3SC/>
- https://blog.csdn.net/weixin_42113967/article/details/113817837

源码：<https://github.com/gisbi-kim/scancontext>

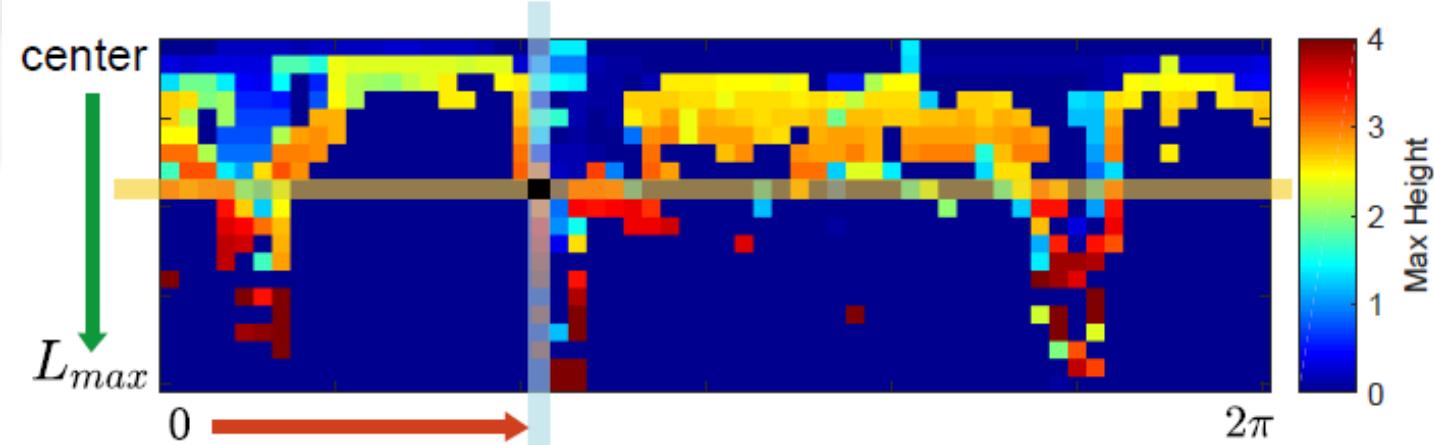


介绍

作者提出了 Scan Context , 主要包括空间描述子的定义方法和与之对应的匹配算法。论文的贡献主要有以下几点 :

- 高效的 bin 编码函数 : 和目前的点云描述子不同 , Scan Context 中提出的方法不需要计算每个 bin 中的点 , 而是有一个更高效的编码函数 , 而且这种编码对点云的密度和法向的变化不敏感。
- 保存点云的内部结构 : 矩阵每个元素的值都由属于该 bin 的点云决定。
- 有效的两阶段匹配算法 : 提供一种旋转不变的描述子 , 首先进行最近邻的搜索 , 然后分层次地把它和逐对相似性得分结合起来 , 避免了搜索整个数据库来进行闭环检测。
- 相比于其他空间描述子而言性能更好。

图中的一个像素块即一个 bin ,
整个图构成了一个矩阵。





2

论文思路



图像直方图

图像直方图 (Image Histogram) 是用以表示数字图像中亮度分布的直方图，标绘了图像中每个亮度值的像素数。这种直方图中，横坐标的左侧为纯黑、较暗的区域，而右侧为较亮、纯白的区域。因此一张较暗图片的直方图中的数据多集中于左侧和中间部分，而整体明亮、只有少量阴影的图像则相反。CV 领域常借助图像直方图来实现图像的二值化。

直方图的意义如下：

- 直方图是图像中像素强度分布的图形表达方式。
- 它统计了每一个强度值所具有的像素个数。

直方图是对数据的统计集合，并将统计结果分布于一系列预定义的 bins 中。这里的数据不仅仅指的是灰度值，且统计数据可能是任何有效描述图像的特征。



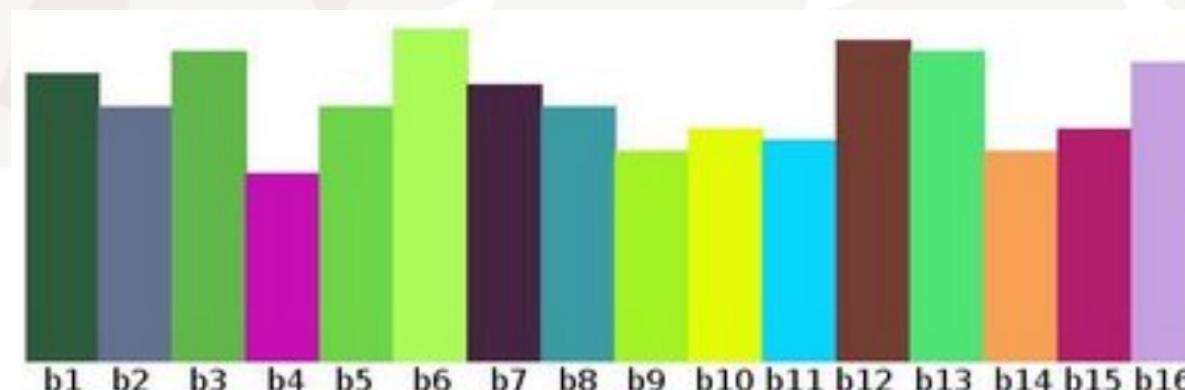
图像直方图

假设有一个矩阵包含一张图像的信息（灰度值 0 - 255），既然已知数字的范围包含 256 个值，于是可以按一定规律将这个范围分割成子区域（也就是 bins）。如：

$$[0,255] = [0,15] \cup [16,31] \cup \dots \cup [240,255]$$

$$\text{range} = \text{bin}_1 \cup \text{bin}_2 \cup \dots \cup \text{bin}_{n=16}$$

然后再统计每一个 $\text{bin}(i)$ 的像素数目。采用这一方法来统计上面的数字矩阵，可以得到下图（其中 x 轴表示 bin，y 轴表示各个 bin 中的像素个数）。



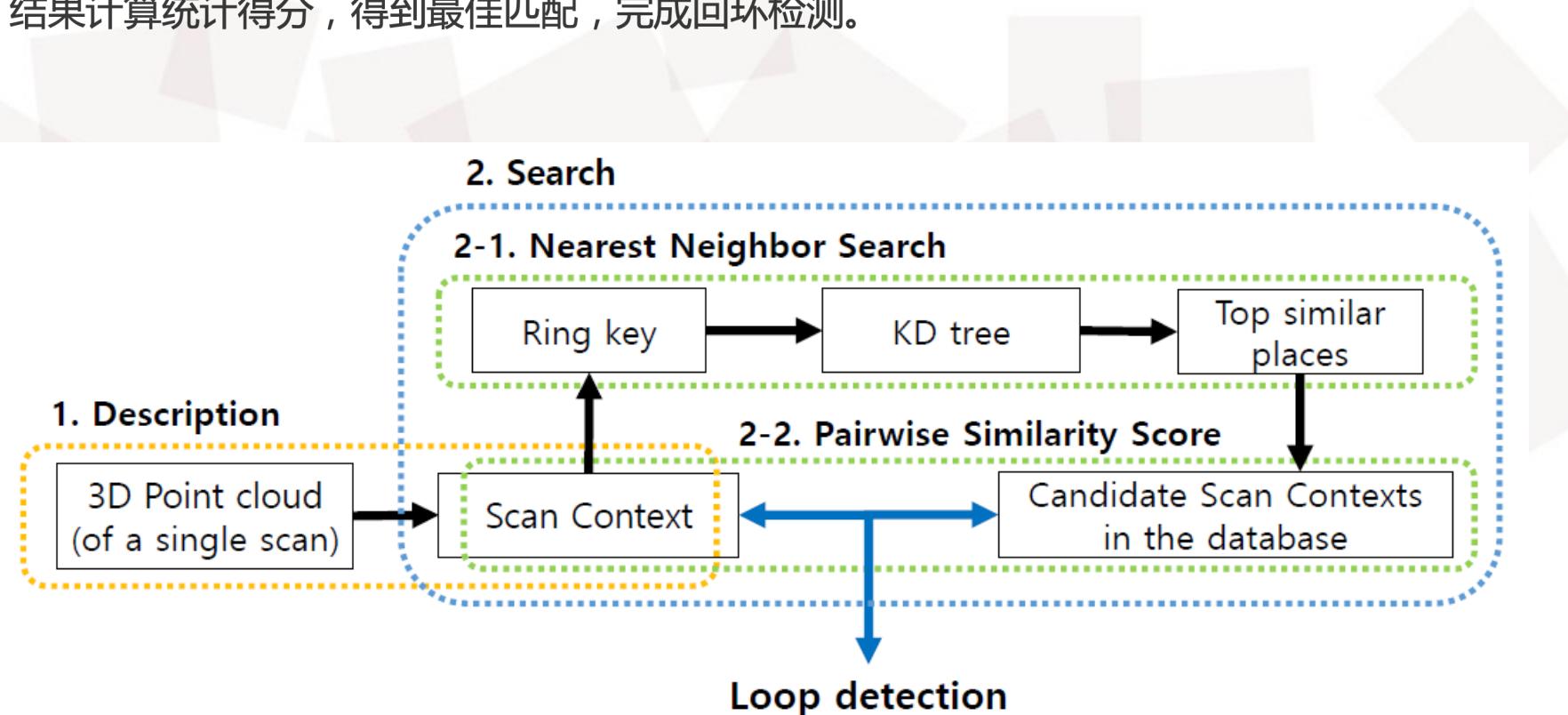
直方图的一些术语和细节：

- dims : 需要统计的特征数目。
- bins : 每个特征空间子区段的数目，可译为 “直条” 或 “组距”，在上例中， $\text{bins} = 16$ 。
- range : 每个特征空间的取值范围。在上例中， $\text{range} = [0, 255]$ 。



算法流程

读取单帧 3D 点云数据，建立 Scan Context，在 Mapping 过程中生成的 KeyFrame 中查找，使用 Ring-key 在 KD-tree 下查找最近邻结果，结果计算统计得分，得到最佳匹配，完成回环检测。



2 论文思路 (Scan Context)

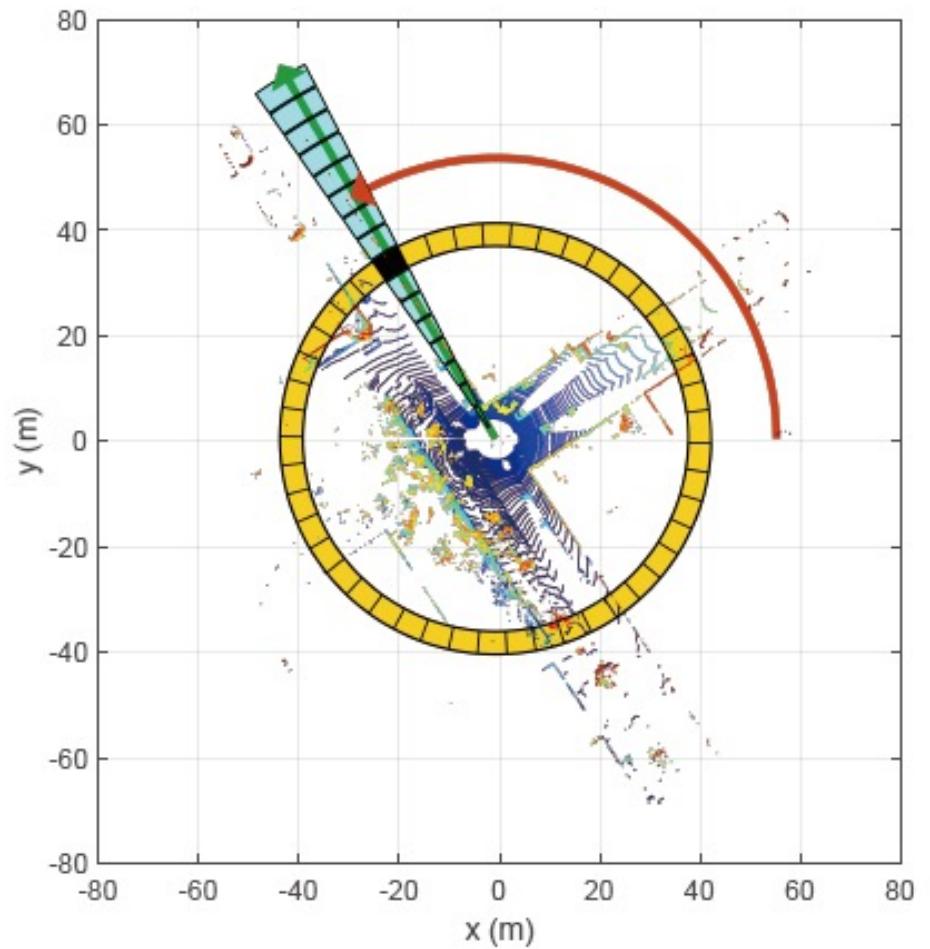
Scan Context 描述子

Scan Context 描述子如图所示，

- ring 为黄色圆环
- sector 为青色部分
- bin 为黑色区域

ring 类似于极坐标中的角度，sector 类似于极坐标中的长度，3D 物理空间被 ring 和 sector 划分成 2D 的“区块”空间，每个区块由 ring 和 sector 唯一确定。同时，将所需相关信息存放在每个区块中，比如：区块中的点云数量、区块中的点云 max height、区块中的点云 mean height 等。

Scan Context 为右图下部分的矩阵，保存了点云的绝对几何结构；每一行代表一个 ring，每一列代表一个 sector，矩阵每个元素的值为对应的 bin 中点云的信息 (max height)。蓝色像素代表 bin 为空或被遮挡。



(a) Bin division along azimuthal and radial directions



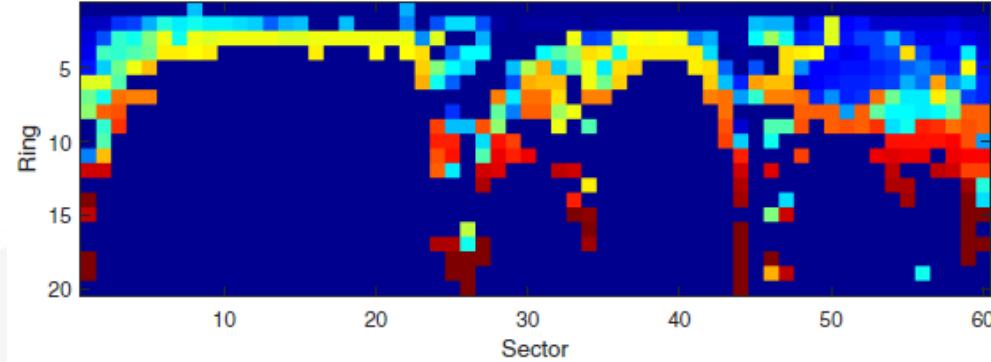


Scan Context 描述子

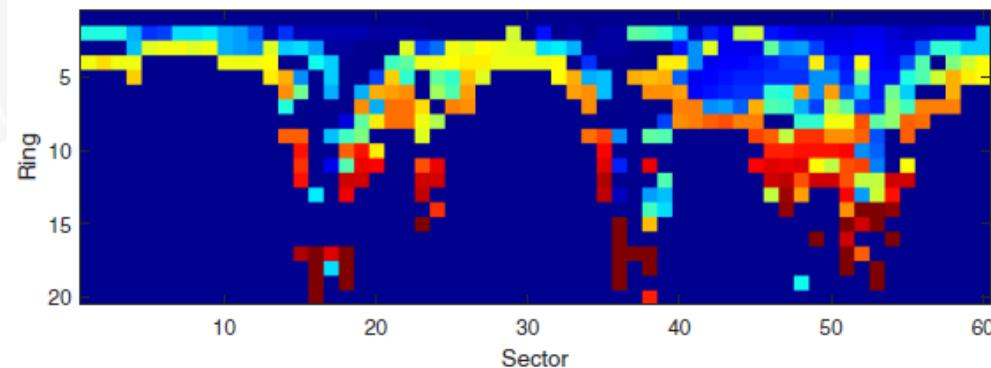
Scan Context 的表示如图 : sector 的数量为 $N_s = 60$, 表示 $0\sim 360^\circ (0\sim 2\pi)$, 每个 sector 的分辨率为 6° ; ring 的数量为 $N_r = 20$, 表示距离 $0\sim L_{max}$ (L_{max} 表示 LiDAR 的最大传感距离)。

图中 (a) 和 (b) 为同一场景下的两帧数据所形成的 Scan Context , 可以看出 , 两个矩阵图像除去水平方向的位移偏差外 , 是高度相似的。即所提到的 “旋转不变性” 。

也就是说 , 即便是每次采集数据的角度偏差很多 , 采集的点云数据也可以通过此种描述子找到对应的场景。直观能想到的 , 只要每次将其中的一个矩阵图像水平移动一个单位 , 然后计算两个矩阵的相似度 , 最终一定能找到对应的回环结果。



(a) The query scan context (3280th scan, KITTI00)



(b) The detected scan context (2345th scan, KITTI00)



Scan Context 描述子

Scan Context 描述子的形成：

设 P_{ij} 表示第 i 个 ring 和第 j 个 sector 所对应的 bin 中的点的集合，则整个点云构成为

$$P = \bigcup_{i \in [N_r], j \in [N_s]} P_{ij}, \text{ where } [N] = \{1, 2, \dots, N\}$$

将点云划分为不同的 bin 后，为每个 bin 指定一个值 $\phi: P_{ij} \rightarrow \mathbb{R}$

$$\phi(P_{ij}) = \max_{p \in P_{ij}} z(p)$$

$z(p)$ 表示取出点 p 的 z 轴坐标，若某个 bin 为空即 $P_{ij} = \Phi$ ，则 $\phi(P_{ij}) = 0$ 。通过这种方式，一个 Scan Context 就可以表示为 $N_r \times N_s$ 的矩阵 I

$$I = (a_{ij}) \in \mathbb{R}^{N_r \times N_s}, \text{ where } a_{ij} = \phi(P_{ij})$$



Scan Context 的相似性得分

前面提到 Scan Context 描述子的“旋转不变性”，可以通过平移矩阵来计算两个矩阵的相似性来检测回环，此部分则对相似性检测进行说明。

记 I^q 和 I^c 分别表示查询点云和候选点云的 Scan Context， c_j^q 和 c_j^c 为 Scan Context 的一列。通过逐列比较来获得二者的相似度，相同索引的列向量之间计算 cosine 距离，然后所有 cosine 距离求和，并除以列数来进行归一化。距离函数如下：

$$d(I^q, I^c) = \frac{1}{N_s} \sum_{j=1}^{N_s} \left(1 - \frac{c_j^q \cdot c_j^c}{\|c_j^q\| \|c_j^c\|} \right)$$

由于激光雷达视角的变化，当机器人重回一个地点时（反向或存在一个角度），即使是同一位置，候选帧的 Scan Context 的列也可能会发生移动，而行的顺序通常不变，那么上式的距离计算则存在偏差。



Scan Context 的相似性得分

为了解决上述偏差问题，本文计算所有可能存在列移动的 Scan Context，并找到最小距离。通过下式来找到达到最好对齐的列移动的数量以及此时的最小距离：

$$D(I^q, I^c) = \min_{n \in [N_s]} d(I^q, I_n^c)$$

$$n^* = \operatorname{argmin}_{n \in [N_s]} d(I^q, I_n^c)$$

其中 I_n^c 表示将原 I^c 延 N_s 方向移动 n 列。

从理论上，使用当前帧 Scan Context 遍历历史所有 Scan Context 一定能找到对应的结果，即相似度的全局极值。但从效率上，这并不可取，还需要更快的方式。于是引入了 Ring Key。



Ring Key 描述子

Ring Key 是一种旋转不变描述子，具体表示为一个向量 k ，向量中每一个元素 $\psi(r_i)$ 为第 i 个 ring 的编码值， r_i 的排布为距离原点从近到远：

$$k = (\psi(r_1), \dots, \psi(r_{N_r})), \text{ where } \psi: r_i \rightarrow \mathbb{R}$$

$$\psi(r_i) = \frac{\|r_i\|_0}{N_s}$$

式中， $\|r_i\|_0$ 表示第 r_i 个 ring 中的非零数量（即 L0 范数）。

右图为向量 k 的排布。

Ring Key 相当于对数据进行了压缩和降维，而由于 Ring Key 是旋转不变描述子，所以两个 Scan Context 做匹配的时候可以直接对比 Ring Key 的相似度，而不用直接使用计算距离的方式，至此有了高效的匹配方式。

2023/5/3

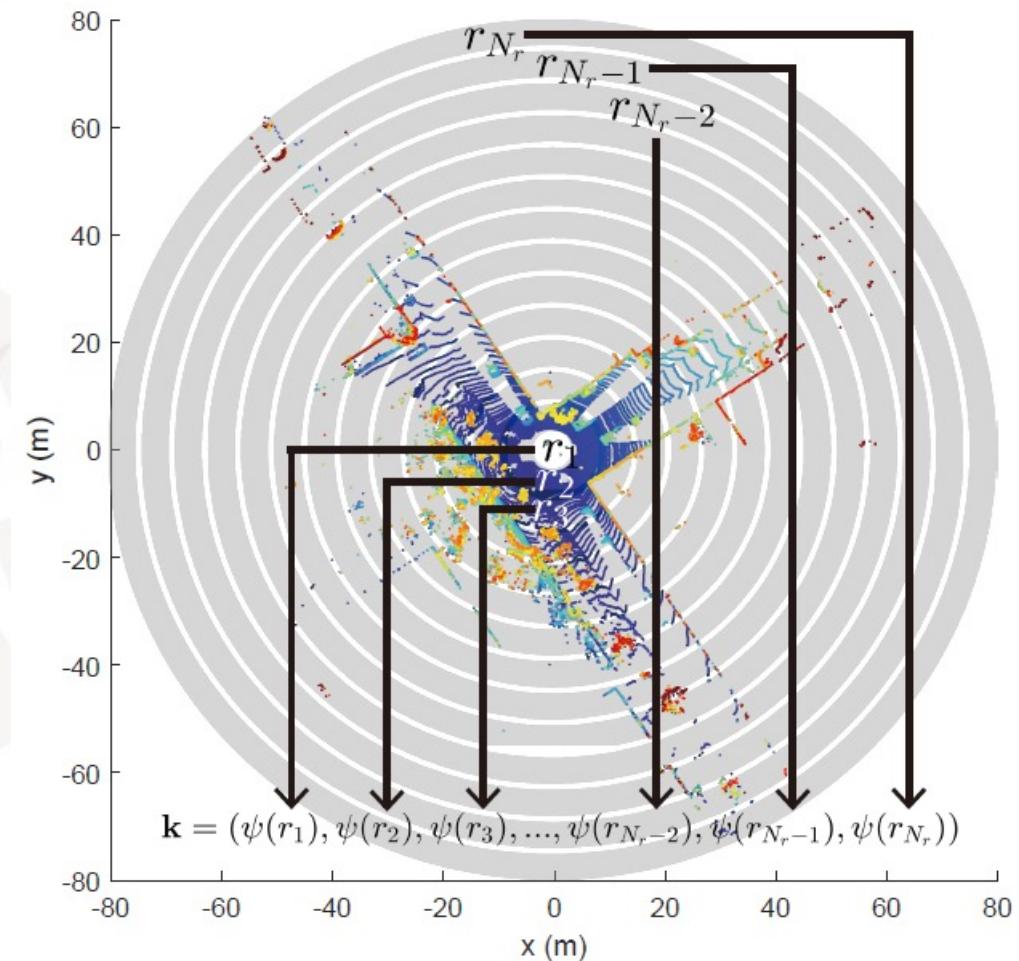


Fig. 4. The ring key generation for the fast search.



Ring Key 描述子 (两阶段层级优化)

向量 k 作为 key 来构建 KD-tree , 同时 , 查询帧的 Ring Key 被用来找相似的 key 以及它们对应的帧索引。通过比较这些候选帧的 Scan Context 与查询帧的 Scan Context 的距离 , 来找到最近的候选帧 , 再通过一个接受阈值 , 找到成为闭环的那帧点云 :

$$c^* = \underset{c_k \in \mathcal{C}}{\operatorname{argmin}} D(I^q, I^{c_k}), \text{s.t. } D < \tau$$

其中 c_k 为从 KD-tree 中抽取的候选帧索引集合 \mathcal{C} 中的一个索引 , τ 为给定的接受阈值 , c^* 即为闭环处的帧索引。



3

实验结果



实验设置

本文算法和全局描述子 M2DP、Z-projection、ESF 在数据集 KITTI、NCLT、Complex Urban LiDAR 上进行了比较：

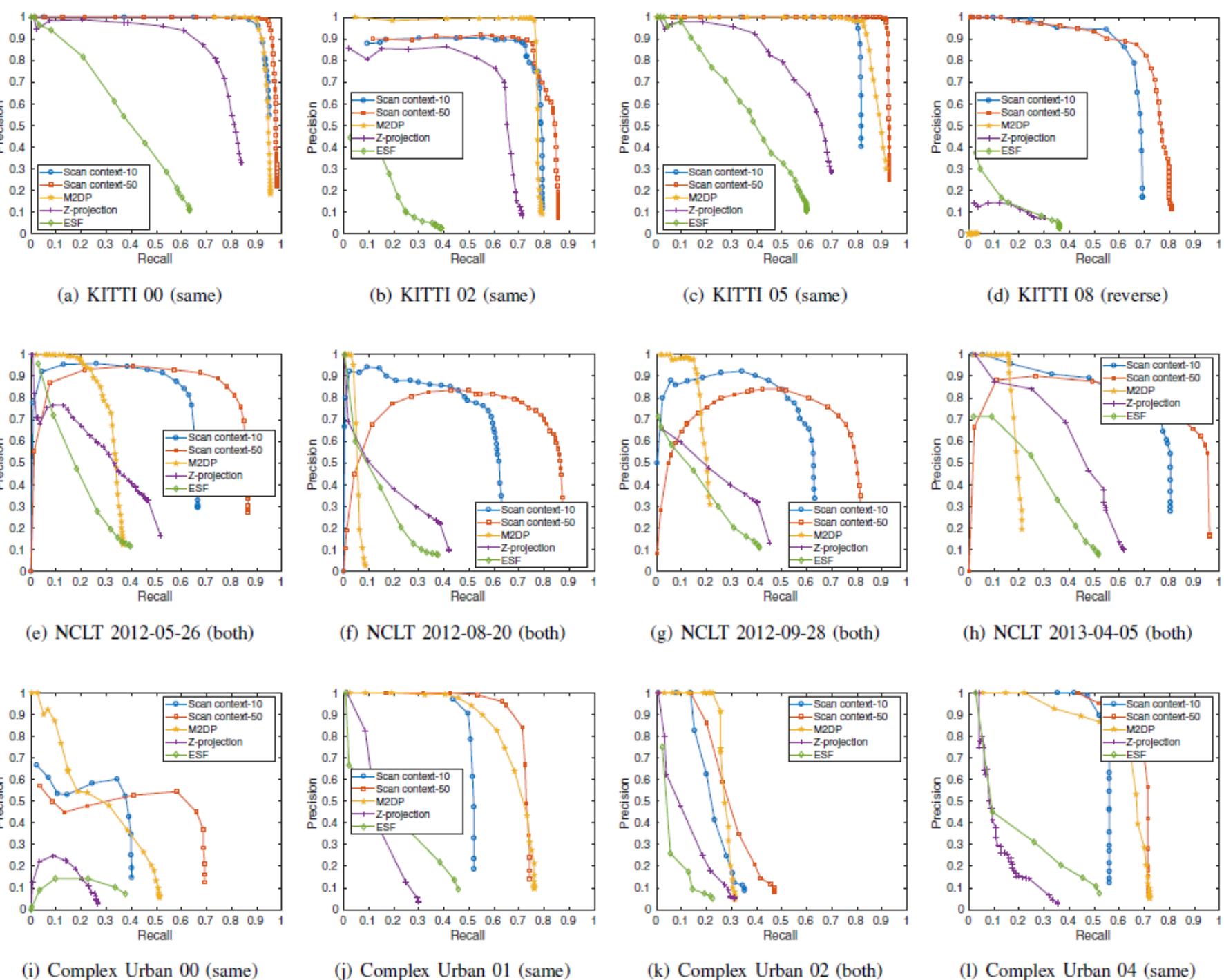
- Route Dir. On revisit 表示当出现回环时 LiDAR 的方向，Same 表示闭环的两帧为同向，Reverse 为反向。
- Scan Context 的比较有两种，即从 KD-tree 中得到的候选帧中选择 10 帧和 50 帧两种方式比较，记为 scan context-10 和 scan context-50。
- 设置 $N_s = 60, N_r = 20, L_{max} = 80m$ 。

TABLE I
SELECTED DATASET LISTS USED IN VALIDATION

	KITTI				NCLT				Complex Urban LiDAR			
Sequence Index	00	02	05	08	20120526	20120820	20120928	20130405	00	01	02	04
Total Length (m)	3714	4268	2223	3225	6345	6018	5579	4530	12020	11830	3020	6542
# of Nodes	4541	4661	2761	4071	3164	3001	2781	2259	3630	3266	862	2140
# of True Loops	790	309	493	332	810	526	635	275	361	383	125	150
Route Dir. on revisit	Same	Same	Same	Reverse	Both	Both	Both	Both	Same	Same	Both	Same

实验结果

通过 precision-recall 来体现实验结果。





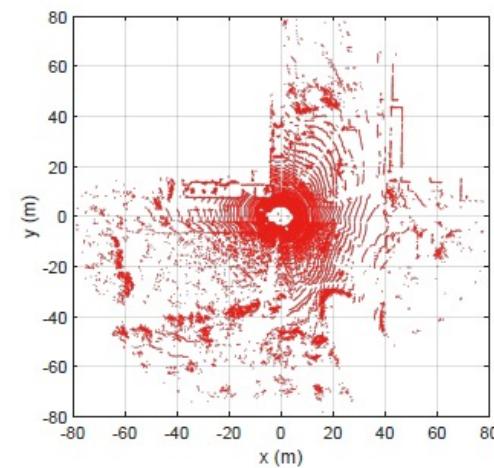
实验结果

- 基于直方图的 ESF、Z-projection 在所选数据集上都表现出比较差的性能。
- scan context-50 的准确性强于 scan context-50，说明 Scan Context 的性能取决于从 KD-tree 中选择的候选帧的数量，这是因为 Ring Key 比 Scan Context 有更少的信息，因此，小数量的候选帧可能容易受到很多相似结构的影响。
- 当应用于室外城市环境时，本文算法优于其他算法。
- 室内环境使得本文算法性能受限，因为室内在垂直高度上变化不显著。
- 本文算法通过视角的对齐展现出很好的旋转不变性。

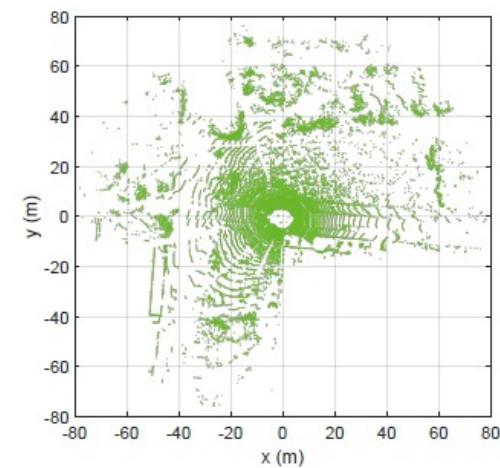
实验拓展

此方法还可以给 ICP 提供一个好的初始化。

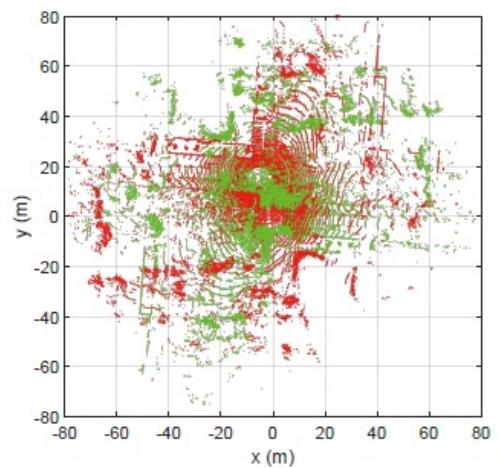
- 右图说明了提供/不提供初始化值时 ICP 的实验结果（实验耗时和 RMSE 误差）。
- 下图 (c) 和 (d) 说明了使用 Scan Context 初始化前后 ICP 的匹配结果。



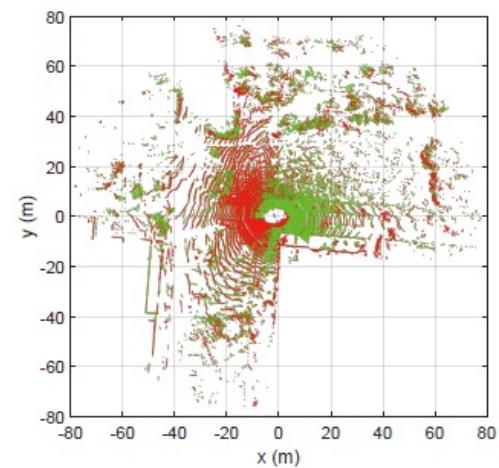
(a) Query point cloud



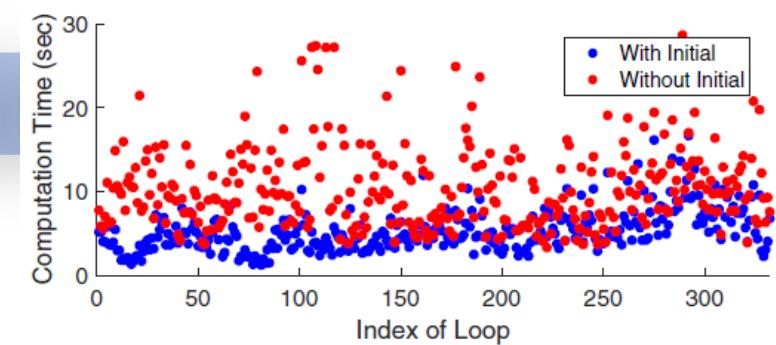
(b) Detected point cloud



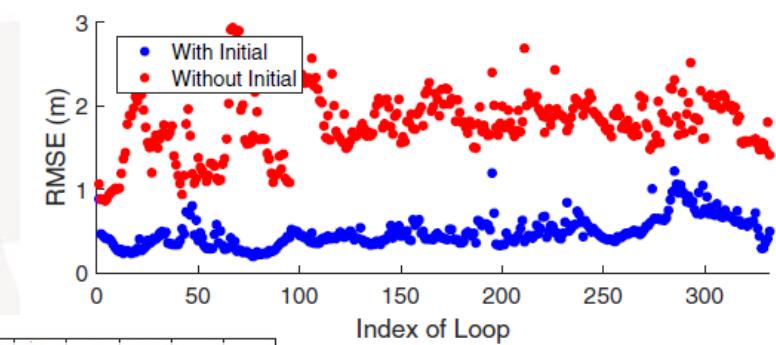
(c) Registration without initial



(d) Registration with Scan Context



(a) Computation Time (s)



(b) RMSE (m)



謝謝

Thank You

THANKS

