点云配准综述：算法、软件加速与硬件约束

邓岂

2026 年 3 月 6 日

1. 引言

三维点云配准研究的是这样一个问题：给定两组无序三维点集 $P = \{p_i\} \subset \mathbb{R}^3$ 与 $Q = \{q_j\} \subset \mathbb{R}^3$，在对应关系未知的条件下，估计刚体变换 $T = (R, t) \in SE(3)$，使 $T(P)$ 与 Q 在某种几何误差度量下尽可能一致。若从优化角度看，这一任务等价于在旋转和平移构成的六维空间中同时处理“位姿”与“对应”两个耦合未知量；其中，对应关系的不确定性决定了问题既难以直接写成闭式解，也容易受到噪声、外点和初始化误差的影响。

1992 年，[1] 提出的点对点 ICP 与 [2] 提出的点对面 ICP 几乎同时给出了局部配准的经典框架：固定当前位姿估计对应，再在固定对应下更新位姿。这个交替结构把原本带有组合搜索性质的问题拆成了两个可重复求解的子步骤，因此很快成为三维配准系统的标准后端。此后三十余年，相关工作一方面沿着算法变体扩展鲁棒性、收敛域和适用场景，另一方面沿着软件实现与硬件架构压缩延迟和功耗。[3] 与 [4] 已经梳理了早期方法谱系；本文进一步把“算法设计”“软件加速”“硬件约束”放在同一框架下讨论。

1.1 点云配准的重要性与应用场景

点云配准之所以长期保持研究热度，原因不在于它只服务某一类任务，而在于多类系统都把它当作几何对齐的基础算子。下面按五类代表性场景说明其需求差异，并给出文献中可直接复核的设置或数量级。

移动机器人同步定位与地图构建（SLAM）。激光雷达里程计依赖逐帧扫描配准估计增量位姿，回环检测则需要把当前观测与历史局部地图重新对齐，以抑制长期漂移。[3] 将这一流水线拆成数据滤波、关联求解、外点剔除和误差最小化四个模块，并用搜索救援、电厂检测、海岸监测和自动驾驶等案例说明：不同场景变化的不是“是否需要配准”，而是允许的重叠率、噪声水平和计算预算。以 KITTI 为例，整套数据共 6 小时，传感器频率覆盖 10–100 Hz[5]；这意味着局部配准若放在在线里程计回路中，留给单帧更新的时间多半只有几十到一百毫秒。

自动驾驶高精度定位。此类任务要求把车载 LiDAR 点云与预建地图实时对齐，因此延迟和功耗常与精度同样重要。[6] 在面向点云配准的体系结构研究中指出，KD 树搜索是普遍存在的主导瓶颈；他们给出的专用处理器 Tigris 在 KD 树搜索子任务上相对 RTX 2080 Ti 获得 77.2 倍加速、7.4 倍功耗降低，折算到端到端配准性能约为 41.7% 提升、功耗约降为原来的三分之一。这一结果直接说明：当系统工作在持续在线模式时，瓶颈不只来自算法误差模型，还来自数据结构与访存方式。

工业检测与机器人拣选。Besl 与 McKay 当年提出 ICP 的直接动机，就是把传感器扫描的刚性零件与 CAD 模型对齐，从而判断加工误差 [1]。这一场景到今天仍然成立，只是约束从“能否对齐”进一步变成“能否在节拍内对齐”。[7] 面向拣选机器人设计的 SoC-FPGA ICP 加速器，在 Amazon Picking Challenge 数据上把位姿估计时间压到 0.72 s、功耗为 4.2 W，相比基于 KD 树的四核 CPU 实现快 11.7 倍；[8] 进一步给出 3.4 W 功耗下最高 17.36 倍的 CPU 加速。这里首先失效的环节多半不是位姿求解，而是最近邻搜索无法跟上抓取周期；一旦搜索延迟失控，后续最小二乘更新再稳定也无法进入控制回路。

医学图像配准与术中导航。计算机辅助手术需要把术前体数据与术中传感器点云对齐，以定位器械或病灶位置。这类场景的特点不是点数一定很大，而是误差容忍极小，且系统不能依赖复杂纹理或大规模离线训练。[3] 将医疗应用列为典型方向之一，原因就在于 ICP 的几何误差模型和局部收敛行为更容易被解释与审查；但这一前提只在初始位姿已较接近、可观测结构充分时才成立。

三维重建与多视角拼接。多视角扫描先做粗配准，再用 ICP 局部精修，原因是只靠局部最近邻难以跨越大位姿偏差。[9] 提出的 FPFH 仍是常用初始化特征之一；而 [10] 在多类数据上比较经典方法与学习方法后指出，

面对部分重叠和复杂几何，不少方法的成功率仍低于 40%。由此可见，两阶段框架并未过时，它仍是把“大范围捕获”和“局部高精度对齐”拼接起来的常见工程方案。

表 1: 第 1.1 节应用场景中的代表性定量约束与数据摘录。仅保留原文或摘要中口径清晰的数字。

场景	代表文献/数据	约束或指标	正文摘录的代表性数值	直接含义
移动机器人 SLAM	[5]	传感器频率、数据规模	KITTI 总时长 6 h；频率 10–100 Hz	在线配准多半只有毫秒到百毫秒级预算
自动驾驶定位	[6]	KD 树搜索与端到端配准性能	KD 树搜索子任务相对 RTX 2080 Ti 为 77.2 倍加速、7.4 倍功耗降低；端到端约 41.7% 提升	系统瓶颈首先落在对应搜索和访存
工业拣选	[7]	位姿估计延迟、功耗	0.72 s，4.2 W，较四核 CPU 快 11.7 倍	节拍受限时，最近邻搜索先成为短板
嵌入式映射	[8]	执行时间与能耗	3.4 W 下最高 17.36 倍 CPU 加速	低功耗部署会反过来约束算法实现方式
多视角重建	[10]	跨场景成功率	部分重叠与复杂几何下，多数方法成功率低于 40%	仅靠局部精修不足以覆盖大位姿偏差

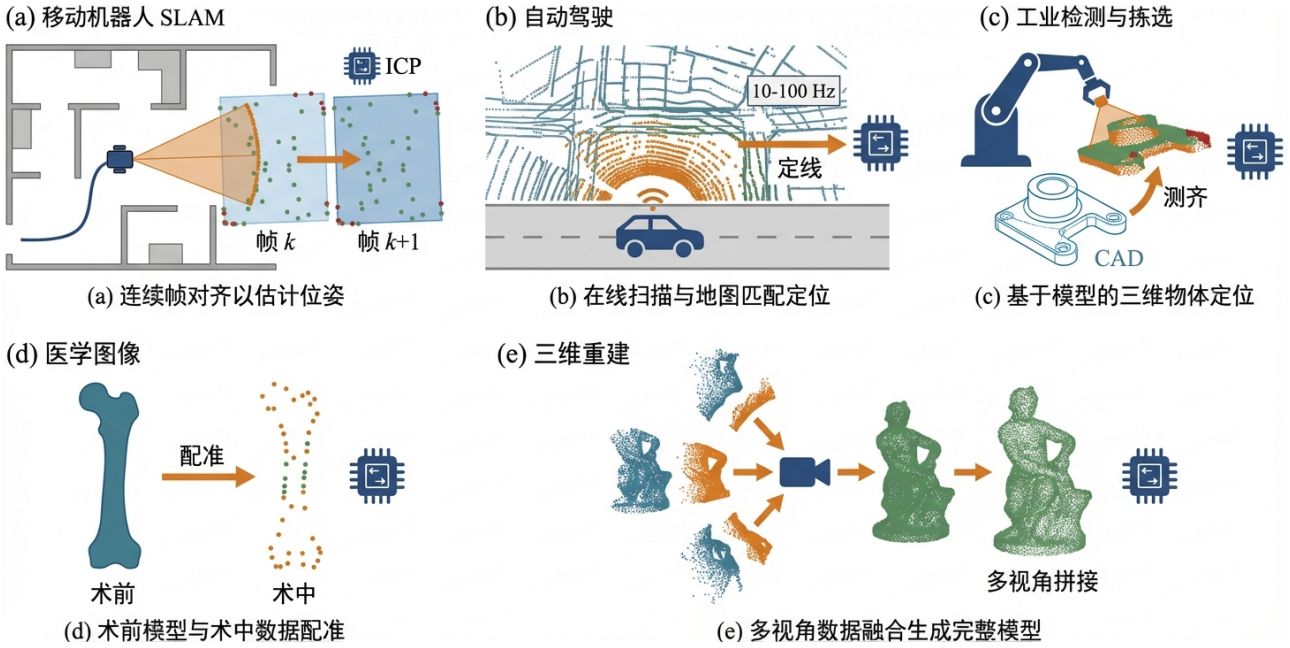


图 1: 点云配准的五类主要工程应用场景概览。(a) 移动机器人 SLAM：LiDAR 帧间配准建立局部地图；(b) 自动驾驶：实时点云与高精地图匹配（10-100 Hz）；(c) 工业检测与拣选：扫描零件与 CAD 模型配准；(d) 医学图像：术前体数据与术中传感器点云对齐；(e) 三维重建：多视角扫描拼接为完整模型。

1.2 ICP 的地位：三十年的主流局部配准基线

ICP 长期作为局部配准基线，主要因为它在“可解释性”“求解成本”和“可替换性”之间保持了较好的平衡。第一，它只要求一个几何最近邻算子，不依赖纹理或人工特征，因此在点云、曲线和网格等不同表示上都能工作 [1]。第二，当对应暂时固定时，旋转和平移可以退化为闭式最小二乘问题，[11] 与 [12] 分别给出了 SVD 与四元数解法。第三，它的主循环天然可拆成“建立对应”“抑制坏对应”“更新位姿”三部分，这使工程实现可以在不推翻整套系统的前提下逐项替换模块 [13]。

[3] 将 ICP 的完整配准流水线形式化为**数据滤波** → **关联求解** → **离群值剔除** → **误差最小化**四个可独立替换的功能模块，该框架将百余页的变体研究系统化整合，成为此后算法比较与工程实现的通用参考架构。

然而，广泛应用也带来了严重的碎片化问题。针对 ICP 的各类失败模式，研究者在不同学术社区中提出了数百种修改方案，彼此间缺乏统一视角与系统对比。针对**部分重叠**场景，[14] 提出了 TrICP 算法；针对**离群值**干扰，[15] 提出了含 ℓ_1 稀疏惩罚的 Sparse ICP；针对**全局最优**性保证，[16] 通过完整 $SE(3)$ 分支定界提供了理论保证；针对**收敛速度**优化，[17] 引入 Anderson 加速将迭代次数削减约 30–35%；深度学习方向，[18] 用注意力机制替代硬最近邻对应，[19] 以可学习 Sinkhorn 最优传输求解软对应概率矩阵。

另一方面，性能压力也推动了与算法并行演化的实现路线。[6] 把 KD 树搜索作为体系结构优化核心；[8] 直

接以暴力近邻搜索的规则访存换取流式 FPGA 效率；PICK [20] 则把 kNN 查询进一步下推到 SRAM 存内计算阵列，报告相对现有设计 4.17 倍加速和 4.42 倍能效提升。也就是说，部署平台并不只是“运行算法”的容器，它会反过来影响对应搜索、数据布局和近似策略的选择。

1.3 综述范围与文章结构

本综述沿**算法分类、软件加速、硬件加速**三条主线组织 ICP 文献，并以统一的计算瓶颈分解框架将算法选择与实现约束有机连接。

第 2 节 形式化定义配准问题，给出刚体变换的数学表示与基本目标函数，梳理标准 ICP 算法的步骤流程与收敛性质，并归纳典型失败模式与应对策略。

第 3 节 按对应建立、鲁棒性增强、收敛加速、变换估计、全局初始化与学习化六个维度系统归纳 ICP 变体，强调每类改动对应的误差来源与计算代价。

第 4 节 讨论软件层加速策略：数据结构优化、降采样/多分辨率处理、并行化与近似最近邻等方法在精度-速度权衡上的可量化影响。

第 5 节 讨论 GPU、FPGA、ASIC 与 PIM 四条硬件加速路径，并将其与最近邻搜索、矩阵构建等计算热点对应起来。

第 6 章 整理主要应用场景、常用数据集与评测协议，并给出跨方法的对比维度与复现要点。

第 7 节 总结仍未解决的核心挑战与潜在研究方向；第 8 节 给出全文结论。

1.4 与已有综述的对比

三篇代表性系统综述均未覆盖本文范围，各有侧重与局限。

[3]（*Foundations and Trends in Robotics*，104 页）从移动机器人视角梳理了 2014 年前的配准算法，涵盖搜索救援、工业检测、海岸监测、自动驾驶四类应用场景，分类框架详尽完善。然而，该综述发表于深度学习配准方法成熟之前，未包含 DCP、RPM-Net 等学习型方法，亦无任何硬件加速内容，且聚焦于移动平台软件实现，未涉及 FPGA/ASIC/PIM 等专用硬件路径。

[4]（*IEEE TVCG*，vol. 19，pp. 1199—1217）覆盖刚性与非刚性三维配准，是该领域较为全面的算法分类综述。其主要局限在于聚焦算法描述层面，不讨论计算复杂度量化、软件优化实现或任何形式的硬件加速，亦不区分不同部署约束下的算法选择策略。

[10]（*ISPRS Open Journal*）同时覆盖经典与深度学习配准方法，在室内到卫星的多源数据集上进行了定量评估，是近年较为全面的方法比较工作。该综述偏重摄影测量与遥感场景，不含硬件加速设计，对嵌入式与边缘部署约束的讨论也相对有限。

除上述三篇系统综述外，[21] 从” 无靶标（cloud-to-cloud）” 配准流程出发，对挑战与潜在研究方向做了凝练讨论，并专门点评了深度学习方法在点云配准中的兴起及其尚未解决的问题；但该工作篇幅较短，缺少跨部署约束（软件与硬件）的定量对比框架。[22] 以更宽口径回顾了三十年 3D 配准方法谱系，但未对 ICP 的加速实现与硬件路径给出面向工程部署的系统总结。

本文的工作重点在于把**算法变体** × **软件加速** × **硬件加速**放进同一条分析链路中：前两类问题决定“误差如何形成、如何被抑制”，后一类问题决定“这些设计是否能在既定时延与功耗下落地”。例如，PICK [20] 强调的是把最近邻查询的数据搬运压到存储层，HA-BFNN [8] 强调的则是以规则数据流替代树结构访问；两者处理的是同一瓶颈，但对应的算法友好性并不相同。

2. 背景与预备知识

本节形式化定义点云配准问题，给出刚体变换的数学表示与基本目标函数，梳理标准 ICP 算法的步骤流程与收敛性质，并归纳典型失败模式与应对策略，为后续章节的变体分析与加速技术讨论奠定理论基础。

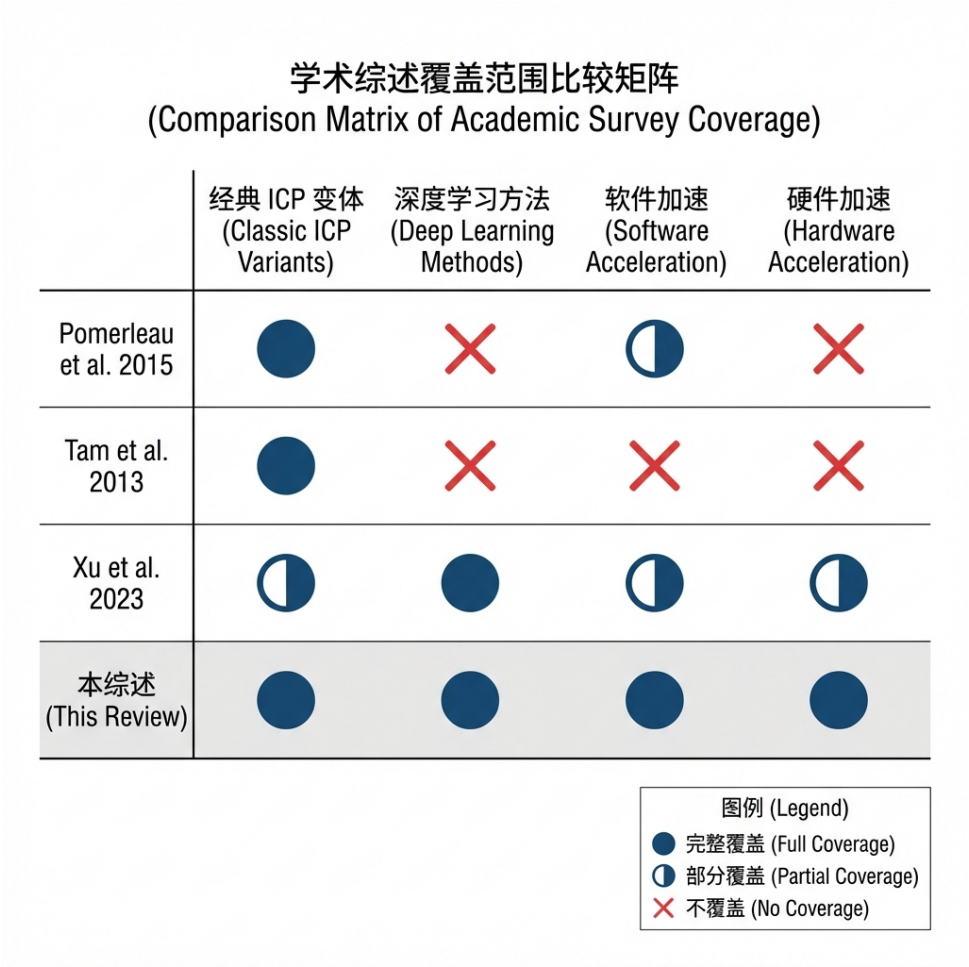


图 2: 本综述与三篇代表性系统综述在覆盖维度上的对比。横轴为四个维度（经典 ICP 变体、深度学习方法、软件加速、硬件加速），纵轴为各综述；实心圆表示完整覆盖，空心圆表示部分覆盖，叉号表示不覆盖。本综述为唯一同时覆盖四个维度的工作。

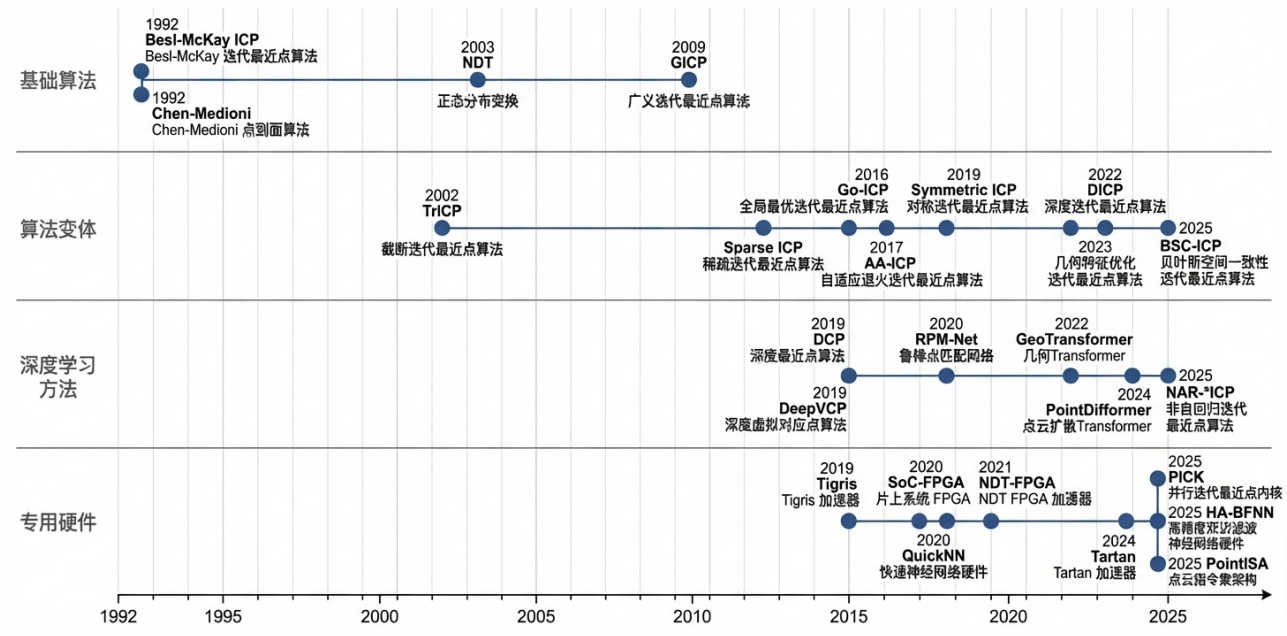


图 3: ICP 算法、加速技术与专用硬件的发展时间轴（1992—2025）。四条泳道分别为基础算法、算法变体、深度学习方法与专用硬件加速器，展示三十余年间的关键里程碑。

2.1 点云配准问题的数学形式化

2.1.1 刚体变换的数学表示

给定源点云 $P = \{p_i\}_{i=1}^{N_p} \subset \mathbb{R}^3$ 与目标点云 $Q = \{q_j\}_{j=1}^{N_q} \subset \mathbb{R}^3$，点云配准旨在估计最优刚体变换 $T = (R, t) \in SE(3)$，使变换后的 $T(P) = \{Rp_i + t\}$ 与 Q 在特定度量意义下达至最优贴合。刚体变换保持点间距离与手性不变，可统一表示为 4×4 齐次变换矩阵：

$$T = \begin{bmatrix} R & t \\ 0^\top & 1 \end{bmatrix}, \quad R \in SO(3),\, t \in \mathbb{R}^3 \tag{1}$$

其中**特殊正交群** $SO(3) = \{R \in \mathbb{R}^{3\times3} \mid R^\top R = I,\, \det R = 1\}$ 刻画三维旋转，**特殊欧氏群** $SE(3)$ 则是旋转与平移的半直积，构成六维李群结构。

$SE(3)$ 完整描述了三维空间中刚体全部可能的位置与朝向——六个自由度中三个对应平移、三个对应旋转。齐次矩阵将旋转与平移整合于统一的 4×4 矩阵表示，连续多步变换可通过矩阵乘法 $T_2 \cdot T_1$ 直接串联，无需分开处理旋转与平移分量，显著简化复合变换的表达与计算。

旋转的两种主流参数化形式各具优势与适用场景。**旋转矩阵** $R \in \mathbb{R}^{3\times3}$ 满足九个约束条件（正交性约束与行列式约束），实际自由度为三；其主要优势在于与向量运算直接兼容，且 SVD 优化过程天然保持正交约束。**单位四元数** $\mathbf{q} = [q_w, q_x, q_y, q_z]^\top$（满足 $\|\mathbf{q}\| = 1$）使用四个参数表示旋转，有效避免欧拉角的万向节锁奇异性，在姿态估计与绝对定向问题中常作为旋转变量直接优化 [12]；旋转矩阵与四元数的转换关系由下式给出：

$$R = \begin{bmatrix} 1-2(q_y^2+q_z^2) & 2(q_xq_y - q_wq_z) & 2(q_xq_z + q_wq_y) \\ 2(q_xq_y + q_wq_z) & 1-2(q_x^2+q_z^2) & 2(q_yq_z - q_wq_x) \\ 2(q_xq_z - q_wq_y) & 2(q_yq_z + q_wq_x) & 1-2(q_x^2+q_y^2) \end{bmatrix} \tag{2}$$

四元数从几何角度编码” 绕某轴旋转某角度” 的物理直觉：$q_w = \cos(\theta/2)$ 对应旋转幅度，$[q_x, q_y, q_z]^\top = \sin(\theta/2)\,\hat{u}$ 对应旋转轴 $\hat{u}$。与旋转矩阵九个元素（含六个正交性约束）相比，四元数仅需维护一个归一化约束，数值优化时更难” 漂移出” 合法旋转集合，因而成为惯性导航与姿态插值的首选参数化形式。

2.1.2 目标函数定义

三维点云配准的主流目标函数可分为三类，分别对应不同的残差几何定义与优化特性。

点对点（Point-to-Point, P2P）度量直接最小化对应点间欧氏距离的平方和 [1]：

$$\mathcal{E}_{\text{P2P}}(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|Rp_i + t - q_{\phi(i)}\|^2 \tag{3}$$

P2P 实现简洁，无需几何先验知识，适用范围广泛；其主要局限性在于光滑曲面上收敛速度较慢——源点沿目标曲面切线方向滑动同样可减小残差值，导致优化路径迂回曲折，而非径直朝向对齐方向收敛，迭代次数因此显著增加。

点对面（Point-to-Plane, P2Pl）度量将残差定义为源点到目标点切平面的有符号距离 [2]：

$$\mathcal{E}_{\text{P2Pl}}(R, t) = \sum_{i=1}^{N_p} \left(\mathbf{n}_{q_{\phi(i)}}^\top (Rp_i + t - q_{\phi(i)}) \right)^2 \tag{4}$$

其中 $\mathbf{n}_{q_j}$ 为目标点 q_j 处的单位法向量。P2Pl 允许源点沿切平面方向无惩罚滑动，等价于只保留法向方向的约束，因此在局部曲面已较稳定时更容易比 P2P 收敛更快。但它的收益建立在两个前提上：一是目标点法向估计足够稳定，二是局部曲面近似能够成立。若法向本身受噪声、稀疏采样或边界效应影响，最先出问题的就是法向投影这一步，随后法方程会沿错误法向累积偏差。[23] 的系统评测也指出，P2Pl 的表现高度依赖场景几何结构与法向估计质量。

点到分布（Point-to-Distribution, NDT）度量将目标区域建模为局部概率分布，计算源点与分布均值的马氏距离 [24][25]：

$$\mathcal{E}_{\mathrm{P2D}}(R,t)=\sum_{i=1}^{N_p}(Rp_i+t-\mu_{\phi(i)})^{\top}\Sigma_{\phi(i)}^{-1}(Rp_i+t-\mu_{\phi(i)}) \tag{5}$$

其中 $\mu_{\phi(i)}$ 与 $\Sigma_{\phi(i)}$ 分别为目标体素内点集的高斯分布参数。NDT 最初针对二维激光匹配提出，随后扩展至三维点云配准领域；其核心思想是以体素级高斯分布替代离散点对，从而在稀疏扫描或强噪声条件下仍能提供更平滑的梯度方向，但代价在于需要预先构建体素格网结构，且格网分辨率一旦选得不合适，最先失真的是局部协方差估计，随后马氏距离会把本不应合并的局部结构一起“平均化”[24][25]。以上三类度量在第 3 节 中还会继续展开。

三类残差度量并非彼此割裂独立。Generalized ICP（GICP）可被视为 P2P 与 P2Pl 之间的统一框架：该算法为每个点建立局部协方差模型，残差以两侧不确定性的加权形式表达，从而在刚性点云配准任务中兼顾收敛速度与建模鲁棒性 [26]。

2.1.3 已知对应关系时的闭式解

当对应关系 ϕ 已知时，P2P 目标函数存在解析闭式解。核心技巧在于**质心解耦**：令 $\bar{p}=\frac{1}{N_p}\sum_i p_i$、$\bar{q}=\frac{1}{N_p}\sum_i q_{\phi(i)}$ 分别为源点云与目标点云的质心，$\hat{p}_i=p_i-\bar{p}$、$\hat{q}_i=q_{\phi(i)}-\bar{q}$ 为去质心后的点坐标，则最优平移由 $t^*=\bar{q}-R^*\bar{p}$ 给出，最优旋转由协方差矩阵 W 的奇异值分解（SVD）确定：

$$W=\sum_{i=1}^{N_p}\hat{q}_i\hat{p}_i^{\top}=U\Sigma V^{\top}\ \Rightarrow\ R^*=U\cdot\mathrm{diag}(1,1,\det(UV^{\top}))\cdot V^{\top} \tag{6}$$

$\mathrm{diag}(1,1,\det(UV^{\top}))$ 修正项保证 $\det R^*=+1$（正常旋转），防止数据呈反射对称时得到镜像解。此 SVD 闭式解由 [11] 于 1987 年提出；基于单位四元数的闭式解则将问题转化为 4×4 对称矩阵的特征值问题，二者同属绝对定向（absolute orientation）问题的经典解法 [12]。两类解法的计算复杂度均为 $O(N_p)$（对固定维度的矩阵分解视为常数级运算），其关键技巧在于” 先归零、再对准”：将两组点云各自平移至质心后，旋转与平移问题实现完全解耦。

然而**实际应用中对应关系 ϕ 在多数情况下未知**，旋转、平移与对应关系三者相互耦合，使直接求解面临指数级组合搜索空间。ICP 的核心贡献就在于把这个联合问题拆成高效的交替迭代：先利用当前变换猜测对应，再在固定对应下求解最优变换。[1] 与 [13] 都强调，这种拆分之所以实用，不是因为它消除了非凸性，而是因为它把原本难以直接求解的联合优化分解为两个可反复高效求解的子问题。

表 2: 第 2.1 节三类基础目标函数的适用条件与失效位置。

度量	约束对象	适合的局部几何	先失效的步骤	后果
P2P	对应点的欧氏距离	点分布较均匀、无需法向先验	最近邻把切向滑动误当成有效改进	收敛慢，易在平滑曲面上反复“贴边走”
P2Pl	目标点法向方向距离	局部曲面近似稳定、法向可可靠估计	法向估计或法向投影失真	法方程沿错误法向累积偏差，局部更新失稳
P2D / NDT	点到局部高斯分布的马氏距离	稀疏扫描、噪声较大、希望目标更平滑	体素尺度不当导致局部统计失真	局部结构被过度平均，细节和边界约束变弱

2.2 经典 ICP 原始算法

1992 年，[1]（通用汽车研究实验室）与 [2]（USC 信号与图像处理研究所）同年独立发表了 ICP 算法，分别面向工业零件精密检测与多视角深度图像融合场景。

2.2.1 算法步骤

给定初始变换 $T^{(0)}$，第 k 次迭代依次执行以下三个步骤。

Step 1（最近点对应）对当前变换后的源点 $p_i^{(k)}=R^{(k)}p_i+t^{(k)}$，在目标点云中搜索欧氏最近邻：

$$\phi^{(k)}(i)=\arg\min_{j\in\{1,\dots,N_q\}}\|p_i^{(k)}-q_j\|_2 \tag{7}$$

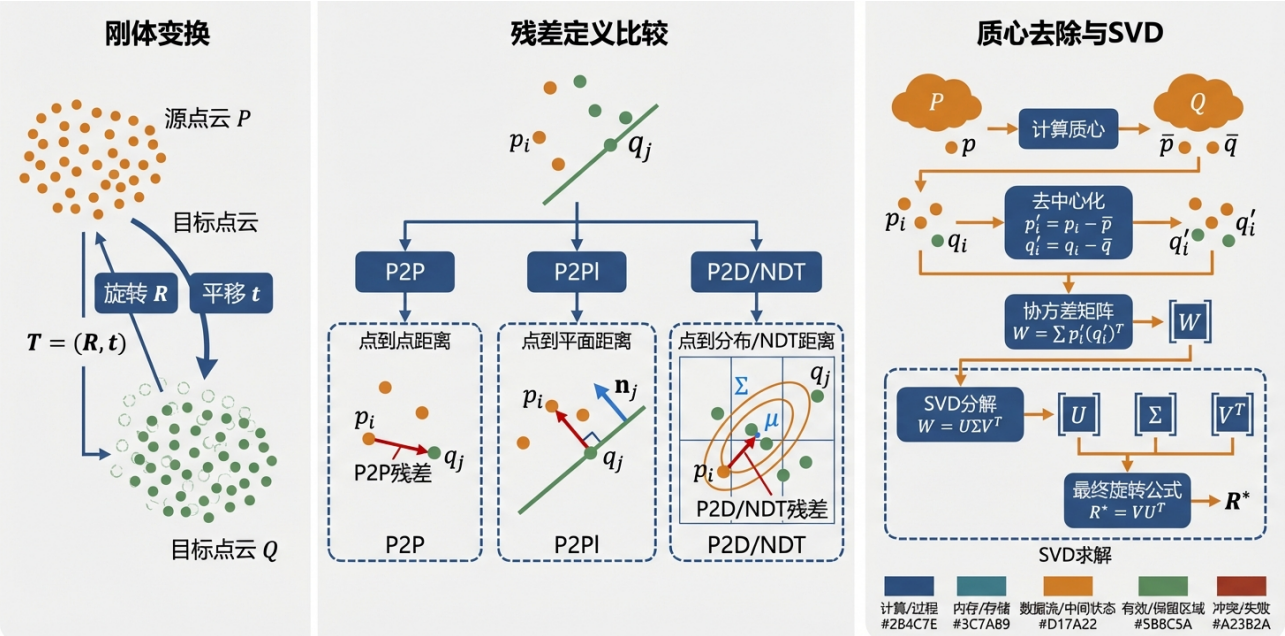


图 4: （左）刚体变换 $T=(R,t)$ 将源点云（红色）变换至目标点云（蓝色）坐标系；（中）三类主流目标函数残差几何定义：P2P 为点到点欧氏距离、P2Pl 为点到切平面有符号距离、P2D 为点到体素高斯分布的马氏距离（图角落给出颜色图例）；（右）SVD 质心解耦：去质心后协方差矩阵 W 的奇异值分解直接给出最优旋转 R^*。

Step 2（变换求解）利用式 6的 SVD 闭式解，对当前对应点对 $\{(p_i,\, q_{\phi^{(k)}(i)})\}$ 求解最优变换 $(R^{(k+1)}, t^{(k+1)})$。Chen 与 Medioni 的变体在此步骤以 P2Pl 目标函数替代 P2P，利用线性化（小角度近似）求解法向约束下的最优变换。

Step 3（收敛判断）若变换增量 $\|T^{(k+1)} - T^{(k)}\|_F < \varepsilon_T$ 或目标函数变化量 $|\mathcal{E}^{(k+1)} - \mathcal{E}^{(k)}| < \varepsilon_E$，算法停止；否则令 $k \leftarrow k+1$ 并返回 Step 1。

三步构成” 猜测对应 $\rightarrow$ 求解变换 $\rightarrow$ 检验收敛” 的闭环迭代结构：Step 1 在当前近似对齐状态下猜测点对对应关系，Step 2 在固定对应关系下求解最小二乘意义的最优刚体变换，Step 3 判断本轮迭代改进是否足够显著；一旦连续两轮变换变化量低于阈值，算法进入稳定区间，输出当前配准结果 [13]。

2.2.2 计算复杂度

Step 1（最近点对应）是整个 ICP 流水线的计算瓶颈。朴素实现的单步最近邻搜索复杂度为 $O(N_p \cdot N_q)$；以 KD 树预处理目标点云后，平均复杂度降至 $O(N_p \log N_q)$，但最坏情形（高维空间或退化分布）仍可回退到 $O(N_p \cdot N_q)$。[6] 对点云配准流水线的实测也表明，KD 树查询是最主要的时间消耗来源之一。这里首先暴露短板的不是 SVD 这类小规模线性代数，而是树结构遍历带来的不规则访存；因此后续很多软件与硬件优化都优先针对 Step 1，而不是 Step 2。

表 3: ICP 各步骤计算复杂度（$N_p \approx N_q = N$）

步骤	朴素实现	KD 树加速
最近邻搜索（Step 1）	$O(N_p N_q)$	$O(N_p \log N_q)$
变换求解（Step 2）	$O(N_p)$	$O(N_p)$
每轮迭代	$O(N_p N_q)$	$O(N_p \log N_q)$

2.2.3 单调收敛性定理

[1] 严格证明了 ICP 在 P2P 度量下的单调收敛性，证明过程基于两个核心引理。

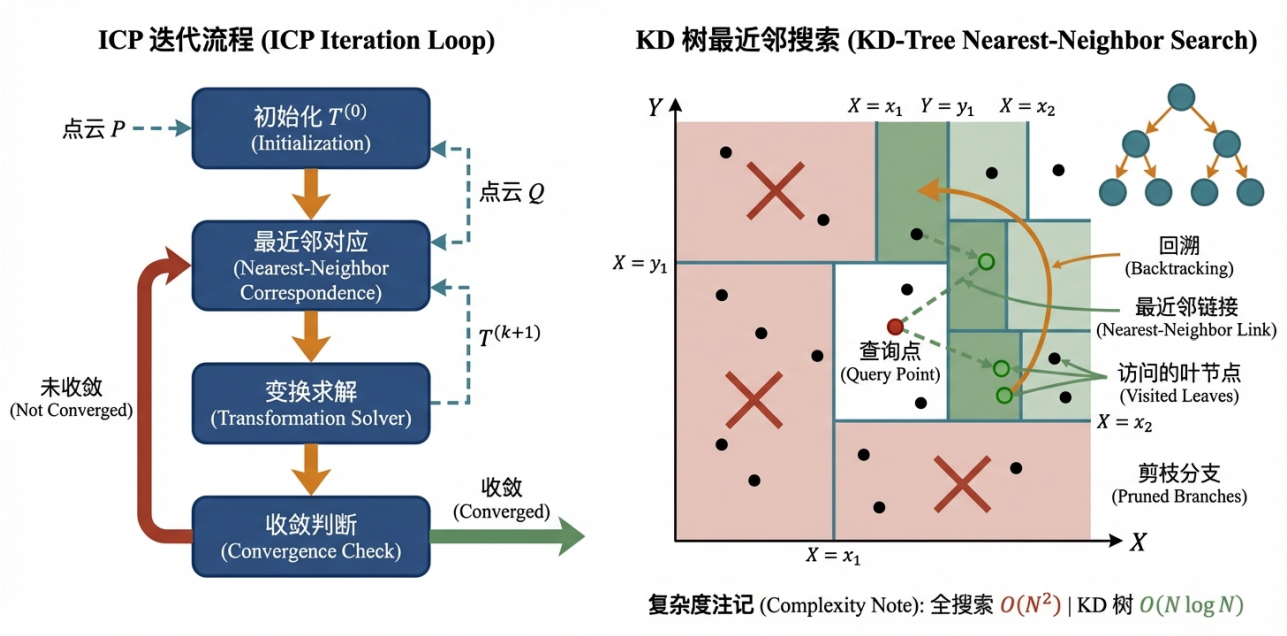


图 5: （左）ICP 算法迭代循环：初始化 $T^{(0)}$ 后交替执行最近邻对应（Step 1）、SVD 变换求解（Step 2）、收敛判断（Step 3），直至满足终止条件；（右）KD 树二维示意：空间递归分割为轴对齐子盒子，查询点（红星）仅需搜索相邻少数叶节点（橙色高亮）而非全体目标点，将搜索复杂度从 $O(N^2)$ 降至 $O(N \log N)$。

定理 1. 定理 *1*（*P2P-ICP* 的单调收敛） 令 $\mathcal{E}(T, \phi)$ 为 *P2P ICP* 的目标函数。若每次迭代首先在固定 $T^{(k)}$ 下利用最近邻准则更新对应关系 $\phi^{(k+1)}$，随后在固定 $\phi^{(k+1)}$ 下利用闭式解更新变换 $T^{(k+1)}$，则目标函数序列 $\{\mathcal{E}^{(k)}\}$ 单调非增并收敛 *[1]*。

引理 1（对应步不增） 固定变换 $T^{(k)}$，重新计算最近邻对应关系时有：

$$\mathcal{E}(T^{(k)}, \phi^{(k+1)}) \;\le\; \mathcal{E}(T^{(k)}, \phi^{(k)})$$

最近邻的定义保证 $\|p_i^{(k)} - q_{\phi^{(k+1)}(i)}\| \le \|p_i^{(k)} - q_{\phi^{(k)}(i)}\|$，逐点不等式对全局目标函数依然成立。

引理 2（变换步不增） 固定对应关系 $\phi^{(k+1)}$，SVD 给出该对应下的全局最优变换，因此：

$$\mathcal{E}(T^{(k+1)}, \phi^{(k+1)}) \;\le\; \mathcal{E}(T^{(k)}, \phi^{(k+1)})$$

两引理合并得到全局单调性：

$$\mathcal{E}^{(k+1)} \;\le\; \mathcal{E}(T^{(k)}, \phi^{(k+1)}) \;\le\; \mathcal{E}^{(k)} \tag{8}$$

序列 $\{\mathcal{E}^{(k)}\}$ 单调非增且有下界 0，由单调有界定理可知其收敛。

该证明仅依赖朴素的单调性观察：每步迭代均做出局部最优决策——最近邻对应是当前变换状态下的最优配对方式，SVD 是当前对应关系下的最优变换方式；两个” 不会变差” 的决策叠加，保证整体目标函数单调下降。目标函数具有自然下界 0，单调且有下界的序列因此收敛。

2.2.4 收敛盆地与局部最优

ICP 仅保证收敛至局部极小值，不保证全局最优性。这是标准 ICP 最重要的理论局限之一 [1]，[13]。收敛结果高度依赖初始变换 $T^{(0)}$ 的质量：当 $T^{(0)}$ 落在正确对齐的收敛盆内时，ICP 可能收敛到期望解；否则容易停在与真值相差显著的局部极小值。对称形状、重复几何结构以及部分重叠都会让这种盆地结构变得更碎，从而使“最近邻 + 最小二乘”这套局部机制在一开始就偏离正确方向 [23]。

针对此理论局限，[16] 提出了 Go-ICP，在分支定界（Branch-and-Bound）框架下搜索完整的 $SE(3)$ 空间：对旋转和平移分别建立不确定包围盒，利用误差上下界做剪枝，并把局部 ICP 作为子程序嵌入搜索过程。根据

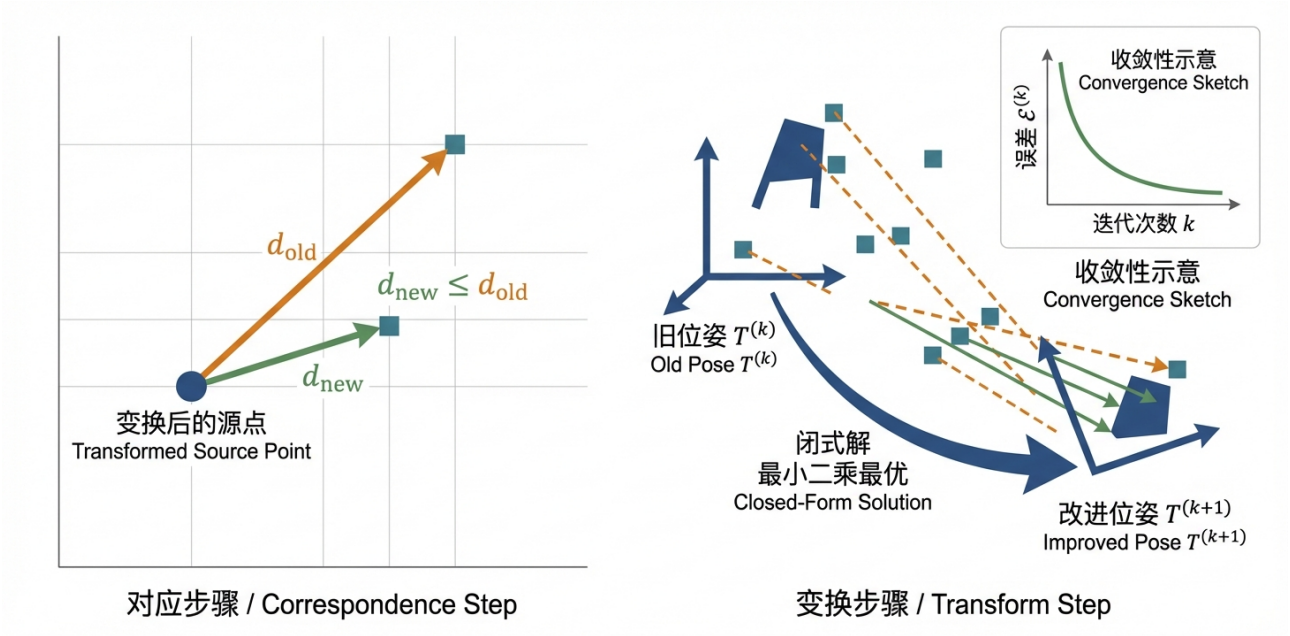


图 6: ICP 单调收敛证明的两步直觉。（左）引理 1（对应步不增）：固定变换 $T^{(k)}$ 后重选最近邻，每个源点的对应距离只会减小或不变（灰色旧对应 $\rightarrow$ 蓝色新对应），全局目标函数因此下降。（右）引理 2（变换步不增）：固定对应后，SVD 给出该对应下的全局最优变换（蓝色最优位置 $<$ 橙色旧位置），目标函数再次下降。右上角 inset 给出 $\mathcal{E}^{(k)}$ 随迭代的单调下降示意。

该文实验总结，在 Stanford 数据与合成数据上，即使初始化随机扰动较大，Go-ICP 仍能稳定得到可靠结果；对包含部分重叠和外点的设置，结合修剪策略后可达到 100% 配准成功率。代价也很明确：它通过扩大搜索范围换取全局性，因此运行时间比局部 ICP 高出一个甚至多个数量级，更适合作为困难帧初始化或精度上界参考，而不是高频在线前端。

2.3 ICP 的核心挑战

尽管 ICP 算法已发展三十余年，以下五类挑战在实际工程部署中依然存在，且相互关联、彼此制约：

（1）局部极小值问题。ICP 目标函数具有非凸特性，最近邻对应关系的离散跳变产生大量局部极小值，使算法结果强烈依赖初始变换 $T^{(0)}$ 的质量。在含高度对称结构或周期性纹理的场景中，即使初始误差仅有数度旋转偏差，也可能导致算法收敛至完全错误的解。

（2）对初始位姿的敏感性。局部 ICP 的收敛盆本来就有限 [23]。当初始扰动过大时，最先被破坏的是最近邻对应：源点会先落到错误表面上，随后最小二乘更新沿着错误对应继续收敛，最终得到一个数值上稳定但几何上错误的解。因此，粗配准或全局初始化不是附加组件，而是把问题送入局部收敛区间的前置条件；相关方法见第 3.6 节 。

（3）噪声与外点鲁棒性。动态目标、遮挡和传感器噪声会生成错误对应。标准 ICP 对所有对应等权处理，因此坏对应一旦进入法方程，就会直接拉偏位姿更新。[14] 讨论部分重叠与外点时已经说明，外点比例上升后，首先失效的是“最近的就是对的”这一前提，随后误差最小化阶段会把这些伪对应当成真约束。更细的鲁棒化机制见第 3.2 节 。

（4）部分重叠（Partial Overlap）。两次扫描常常只有部分区域重合，未重叠区域中的点天然找不到正确匹配，其“最近邻”只是距离最近的伪对应。这类误差不是随机噪声，而是带方向性的系统偏差：未重叠区域越大，位姿更新就越容易被拖向错误区域。[14] 针对这一问题提出截断思想，实质上就是先减少未重叠区域对目标函数的支配，再谈局部优化。

（5）计算效率瓶颈。最近邻搜索常是配准流水线中最耗时的部分 [6]。在需要持续在线运行的系统里，最先超预算的多半也是这一步，而不是位姿更新本身。软件层面的 KD 树优化、近似最近邻与 GPU 并行化只能部分缓解这一问题；当访存不规则、点数持续增长或功耗预算过低时，系统会进一步转向专用加速器，如 Tigris、

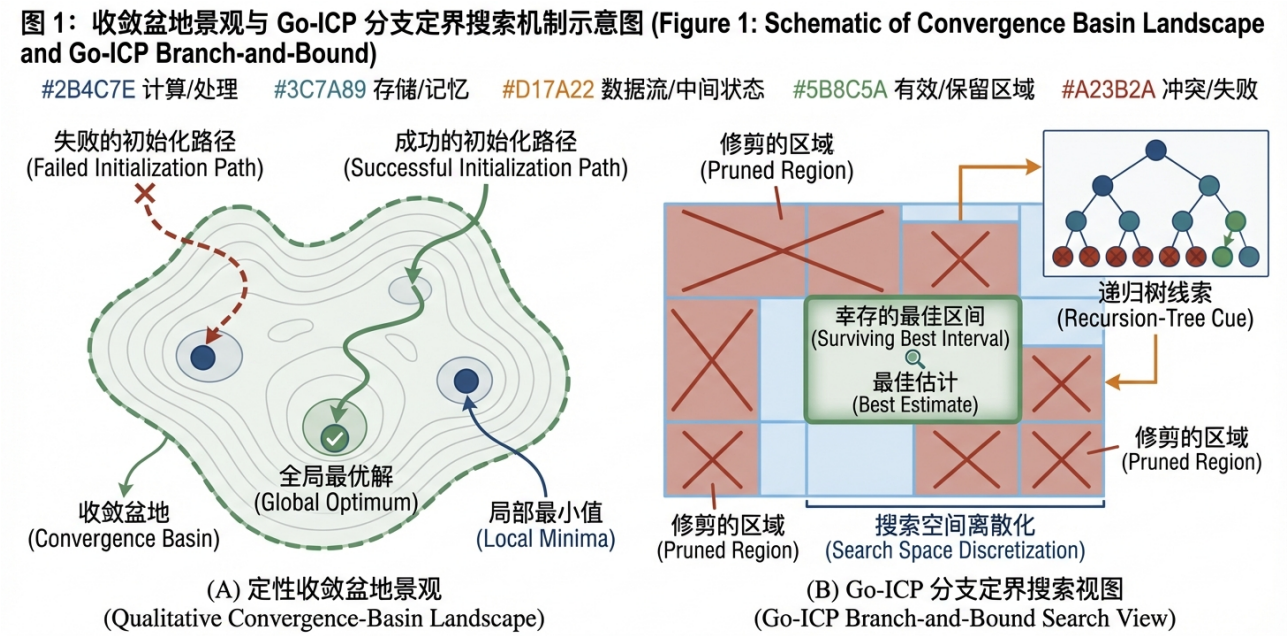


图 7: （左）标准 P2P ICP 的收敛盆地（basin of convergence）：以旋转偏差 $\Delta\theta$ 为横轴，目标函数值 $\mathcal{E}$ 为纵轴，灰色区域为正确收敛盆，圆形标记为多个局部极小值；初始变换落在盆内（蓝箭头）则收敛至全局最优，落在盆外（红箭头）则陷入局部极小。（右）Go-ICP 的 $SE(3)$ 分支定界搜索示意：将旋转空间划分为立方体包围盒并递归剪枝，灰色盒子为已剪枝区域（下界高于当前全局上界），绿色标记为最终最优区间。

HA-BFNN 和 PICK。对应的软件与硬件路径分别见第 4 节 与第 5 节 。

表 4: 第 2.3 节五类核心挑战的“触发条件—失效位置—后果”对应关系。

挑战	触发条件	最先失效的步骤	直接后果	主要应对方向
局部极小值	初始位姿偏差大、结构对称或重复	最近邻落入错误盆地	收敛到错误局部解	第 3.6 节的全局初始化
初始位姿敏感	IMU 漂移、回环前位姿不准、大角度旋转	对应建立先错	后续最小二乘在错误对应上稳定收敛	粗配准、多分辨率、两阶段框架
外点鲁棒性	动态目标、遮挡、离群噪声	坏对应进入法方程	位姿更新被系统性拉偏	第 3.2 节的鲁棒核、截断和剪枝
部分重叠	未重叠区域占比高	非重叠点被当成近邻	目标函数被伪对应主导	截断、重叠估计、前置过滤
计算效率	点数大、频率高、功耗受限	最近邻搜索与访存	无法进入实时回路	第 4 节与第 5 节

3. ICP 算法变体

第 2 章将 ICP 归结为一个可模块化替换的迭代流水线：建立对应 → 外点抑制 → 位姿更新 → 终止准则 [3]。这套拆解之所以有用，正是因为工程里常常要把“算法环节”对齐到“传感器节拍”：Pomerleau 的综述在多个移动机器人案例里都提到，为了把计算卡进实时预算，实践中常会把每帧参与匹配的点数压到“几千量级”（例如随机采样 5000 点再做匹配与优化）[3]。如果把这个直觉换成更直观的系统约束：在移动机器人常见的 10 Hz 里程计频率下，配准模块通常只能容许 5–10 次迭代的预算；而以 Velodyne HDL-64E 为例，点云吞吐可达约 1.3M 点/秒，最近邻检索、外点处理和线性化求解的常数因子就会变得非常“值钱”。

经典 ICP 形式还隐含了几条看似朴素、但一旦失效便导致配准失败的强假设：其一，最近邻足以近似对应关系 [1]——在 Besl 与 McKay 的早期实验里，点集规模可以只有十来个点（例如 8 点对 11 点，约 6 次迭代收敛，耗时 <1 s），但在更接近真实模型的设置下，单个对象也会很快上到数千点（例如 2546 点、24 个初始旋转，

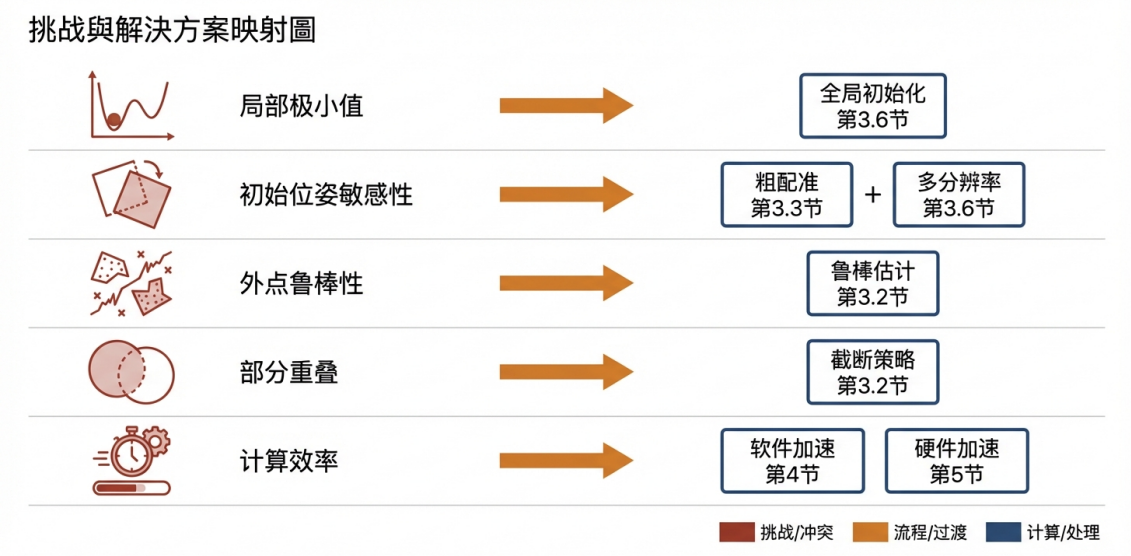


图 8: ICP 五类核心挑战及各章节的应对策略概览。每行为一类挑战，右侧箭头指向对应的解决方向：局部极小值 → 第 3.6 节；初始位姿敏感性 → 第 3.3 节 与第 3.6 节；外点鲁棒性 → 第 3.2 节；部分重叠 → 第 3.2 节；计算效率 → 第 4 节 与第 5 节。

每个初值约 6 次迭代；报告的 RMS 约 0.59），此时“对应错一点”就会被迭代不断放大。其二，残差用二次损失刻画就足够 [2]——而 Chen 与 Medioni 把误差写成源点到目标切平面的有符号距离，本质上只在法向方向施加约束；他们的多视角建模实验里，相邻视角常按约 45° 的间隔采集（侧面约 8 视角，并补充 6–8 个顶/底视角），这类设置恰好能让“切平面近似”在局部成立。其三，位姿更新可以靠一次线性代数步骤解决（Kabsch/Horn 型闭式解）[12]：这类解法往往要求至少 3 对不共线点对才能避免退化，并且把误差模型（是否等方差、是否需要加权）直接写进求解器。其四，只要不断迭代就会在局部盆地里稳稳收敛 [13]——但 Rusinkiewicz 与 Levoy 的速度向实验也明确指出：只在“初值已经很好”的前提下，组合投影匹配与点到平面度量才能把两帧深度图的对齐压到几十毫秒量级（他们在 550 MHz 处理器上报告约 30 ms），而当初始误差大到旋转偏差超过约 20–30° 时，局部 ICP 很容易被局部极小值困住。

这些假设一旦被噪声、外点、部分重叠或结构退化打破，标准 ICP 就会出现慢收敛甚至收敛到错误解等失败模式：同一套实现在“室内稠密、特征丰富”的场景中表现稳定，而在低重叠、重复结构或几何退化（长走廊、隧道、开阔地）的场景下则明显不可靠。

本章的任务是系统梳理研究者如何逐环节放宽上述假设，并形成可复用的工程组合：第 3.1 节 从度量与约束形式出发组织对应策略；第 3.2 节 讨论从截断到鲁棒核、再到图论剪枝的外点处理；第 3.3 节 聚焦迭代过程的加速与稳定化（如 AA-ICP/FRICP 等）[17]——例如 AA-ICP 在 Freiburg RGB-D 等真实数据基准上给出的统计是：中位数约 35% 的整体加速，且约 97% 的测试序列最终误差更优（论文用 $\varepsilon = 0.001$、历史深度 m 通常取 5–10，并限制混合系数避免发散）。第 3.4 节 回顾变换估计的参数化与不确定性建模（如 GICP、Stein ICP）[26]：GICP 的一个典型实现细节是用约 20-NN 估计各点的局部协方差，并在实验里把迭代上限设为 50 或 250 轮以对齐不同的速度/精度取舍。第 3.5 节 讨论几何退化与可定位性分析对 ICP 可观测性的影响 [27]：在他们的水下声纳 SLAM 实验中，退化感知机制会主动拒绝不可靠的闭环（例如某组数据里直接拒绝了 25 个错误闭环），把“约束不足方向”从图优化里剔除，避免轨迹被拖向错误极小值。第 3.6 节 总结全局初始化与两阶段框架如何为局部 ICP 提供可用起点（FGR、Go-ICP、TEASER++ 等）[28]：FGR 在无初值、低重叠（最低约 21%）的 UWA 基准上报告 0.05 阈值下约 84% 的 recall，同时把单次配准平均耗时压到约 0.22 s，并给出相对 CZK 约 50×、相对 ICP 约 2.8× 的速度优势。第 3.7 节 介绍深度学习方法在特征、对应与端到端估计上的替代与融合趋势 [18]——DCP 在 ModelNet40 的常用设置下把旋转误差从传统 ICP 的灾难性水平（例如 MSE(R) 约 895）拉回到可用区间（DCP-v2 的 MSE(R) 约 1.31、RMSE(R) 约 1.14°），推理时间也可到约 8 ms/对点云量级。第 3.8 节 则以优化视角统一这些变体的求解器结构与可认证性边界。

贯穿全章的主线并非“提出新损失函数”本身，而是围绕可观测性、鲁棒性与可计算性三者的取舍：在可观

测性不足（第 3.5 节 ）或初始误差过大（第 3.6 节 ）时，任何局部加速（第 3.3 节 ）都难以挽救；而当对应质量（第 3.1 节 ）与外点处理（第 3.2 节 ）足够可靠时，求解器设计（第 3.8 节 ）往往成为决定实时性与可扩展性的关键因素。

3.1 对应关系建立策略 (Correspondence Strategy)

对应关系建立是 ICP 每次迭代的起点，对应的可靠性几乎直接决定了位姿更新的稳定性与收敛域。[3] 用一篇长篇综述（pp. 1–110）把配准流程拆成**数据滤波 → 关联求解 → 外点剔除 → 误差最小化**四个可独立替换的模块；在其移动机器人案例里，为了卡进实时预算，甚至会直接把每帧参与匹配的点压到“几千量级”（例如随机采样 5000 点再做匹配与优化）[3]。在这套框架下，“关联求解”（association solver）逐渐固化出六类常见策略。它们的差异大体可从两个维度理解：一是**用何种度量刻画源点与目标的“接近程度”**（欧氏距离、切平面距离、概率密度、几何特征相似度、语义一致性等）；二是**对应如何被建立并施加约束**（单向最近邻、双向互惠一致、软对应概率矩阵等）。

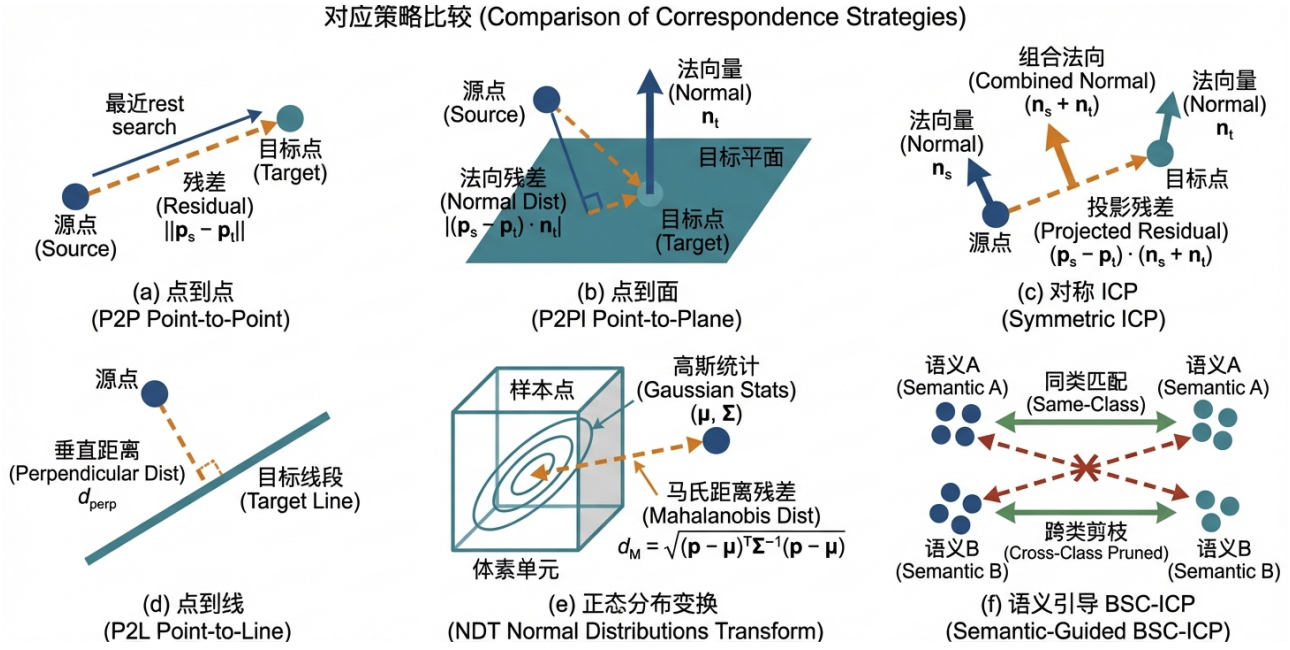


图 9: ICP 对应关系建立的六类策略几何示意。(a) 点到点（P2P）：源点（红星）与目标最近邻（蓝点）的连线为残差；(b) 点到面（P2Pl）：源点到目标切平面的有符号距离为残差；(c) 对称 ICP：双侧法向量平均方向上的投影差；(d) 点到线（P2L）：源点到目标线段的垂直距离（2D LiDAR 场景）；(e) NDT：目标空间划分为体素，每体素拟合高斯分布，残差为马氏距离；(f) 语义引导（BSC-ICP）：仅在相同语义类别内搜索最近邻，双向互惠约束过滤歧义配对。

3.1.1 点到点、点到面与对称 ICP

标准 ICP 的两种原始形式都基于单向最近邻。**点到点（P2P）**[1] 以欧氏最近邻为对应，最小化点间距离平方和：

$$\mathcal{E}_{\text{P2P}} = \frac{1}{N} \sum_{i=1}^{N} \|R p_i + t - q_{j^*}\|^2 \tag{9}$$

P2P 形式简洁：每轮仅需一次最近邻查询，再求解一个 $SE(3)$ 刚体最小二乘（对应 6 个自由度的更新）。[1] 的奠基论文通过一系列规模较小但口径明确的实验，系统展示了 ICP 的计算行为：点集示例里，8 个数据点对 11 个模型点，只跑 6 次迭代，整次 ICP 用时不到 1 s；曲线示例把一条 3D 样条用 64 个点的折线近似，并给每个点加高斯噪声，作者用 12 个初始旋转与 6 个初始平移（共 72 个初值）做全局尝试，最终在约 6 min 找到正确对齐；在“扫描对模型”的表面示例里，mask 模型用 2546 个点近似，作者用 24 个初始旋转、每个初值跑 6 次迭

代，约 10 min 得到 RMS 0.59 的对齐结果 [1]。上述实验的价值在于揭示了 ICP 的计算结构：最近邻查询与初值枚举往往比后端求解消耗更多资源，后续大量加速与鲁棒化工作从本质上都是在降低这两个环节的计算常数。

点到面（P2Pl）[2] 将残差替换为源点到目标切平面的有符号距离，即只在法向方向施加约束，从而允许点在切平面内“滑动”而不受惩罚。其原始动机是“多视角范围图建模”：在 ICRA 1991 的实现里，他们以 Mozart bust 的范围图为例展示配准流程，并在 wood blob、plaster tooth 的建模实验中采用 8 个侧视角，顶/底再补 6–8 个视角，相邻侧视角间隔约 45°；这些视角数与旋转步长决定了相邻帧的重叠比例，也解释了 P2Pl 为什么更适合做局部细化：只要重叠足够，法向方向那一条约束就能把两帧“拧”到一起 [29]。在更晚的系统对比中，[23] 在 `libpointmatcher` 里给出了两条可复现的基线链（原文表 5），并在“Challenging Laser Registration”的 6 个真实场景（Apartment、Stairs、ETH、Gazebo、Wood、Plain）上对比 P2P 与 P2Pl。两条链共享的设置包括：先用 `MinDist` 去掉 1 m 内近距离点；对待配准点云（reading）用 `RandomSampling` 随机保留 5% 点；对应搜索用 KD-tree（近似常数 $\epsilon = 3.16$）；外点用 `TrimmedDist` 做比例截断；终止条件为最多 150 次迭代或增量低于 1 cm / 0.001 rad。两者的差异主要在参考点云（reference）侧：P2P 同样随机保留 5% 点，并保留最近 75% 的对应；P2Pl 则用 `SamplingSurfaceNormal` 将下采样与法向估计合并（约 7× 下采样，阈值 7 点），并保留最近 70% 的对应。以单核 2.2 GHz Core i7 的单次配准耗时计，作者报告 P2P 的中位用时为 1.45 s、P2Pl 为 2.58 s；据此指出，P2Pl 额外的法向估计开销并未被迭代次数减少完全抵消 [23]。两者的目标函数推导与收敛性分析已在第 2.1.2 节 中给出，本节不再重复。

P2Pl 仅利用目标侧法向量，忽略了源点自身的法向信息。[30] 于 2019 年提出**对称 ICP（Symmetric ICP）**，将残差定义为对应点对双侧法向量之和方向上的投影差：

$$\mathcal{E}_{\text{sym}} = \sum_{i=1}^{N} \Big[(Rp_i + t - q_i) \cdot (n_{p_i} + n_{q_i}) \Big]^2 \tag{10}$$

对称目标函数线性化后可沿用与 P2Pl 相同的闭式求解器，计算开销几乎不增加。它的关键不在“多了一个法向”，而在零残差集合被扩展了：点对落在二次曲面（常曲率片）上时，对称目标依然可以做到零残差，允许配准在曲面上更自由地“贴着走”，从而把可收敛的初值范围往外推 [30]。作者在 Bunny 的初值盆实验里把这件事做成了可复现的数字口径：选取 `bun000` 与 `bun090` 两帧扫描（两者 IOU 约 23%），将初始误差离散成“旋转角 × 平移幅值（按模型尺寸归一化）”的二维网格，并在每个网格点上采样 1000 个随机初始变换；随后分别统计在 20/100/500 次迭代内成功收敛的比例热图（原文图 5）[30]。

更关键的是，这种“收敛域更宽”并不是一句空话。[30] 在 Bunny 扫描数据上把初值难度离散成二维网格（初始旋转角度、初始平移幅值，平移幅值按模型尺寸归一化），并对每个网格点采样 1000 个随机初始变换；随后分别统计在 20 次、100 次、500 次迭代内收敛的成功比例（原文图 5）。对称目标在这些热力图上的高成功率区域明显外扩，直观解释了它为什么更不容易被初值误差锁死在错误盆地：二阶曲面上残差为零，使得优化更接近“贴着曲面走”，而不是被单侧切平面强行拉回。

3.1.2 点到线对应（2D LiDAR 变体）

当传感器为 2D 激光雷达（SICK LMS、Hokuyo 等）时，环境可建模为平面线段集合而非三维点集。**点到线（P2L）**将残差定义为源点到最近目标线段的垂直距离：

$$e_i = d(p_i,\ l_{j^*}), \quad l_{j^*} = \arg\min_{l_j} d(p_i, l_j) \tag{11}$$

[23] 在 `libpointmatcher` 框架中对 P2L 进行了系统实现与对比，指出其在走廊、房间等高度结构化的 2D 场景中，收敛速度可与 P2Pl 相当，且不依赖三维法向量估计，因而更适合算力受限的平台。其主要局限在于线段提取对传感器噪声与遮挡更敏感；在植被覆盖或非结构化室外地形中，线段质量下降后，P2L 常会退回到与 P2P 类似的行为，性能优势随之减弱。

3.1.3 点到分布对应（Normal Distributions Transform）

正态分布变换（NDT）由 [24] 在 IROS 2003 提出，从根本上摆脱了” 显式点对点对应” 的范式。其核心思路是将目标空间划分为均匀体素格网，对每个体素 c 内的局部点集拟合高斯分布 $\mathcal{N}(\mu_c, \Sigma_c)$，源点的匹配代价定

义为其在概率密度场中的负对数似然之和：

$$\mathcal{E}_{\mathrm{NDT}} = -\sum_{i} \exp\left(-\frac{(p_i' - \mu_{c(i)})^{\top} \Sigma_{c(i)}^{-1} (p_i' - \mu_{c(i)})}{2}\right) \tag{12}$$

其中 $p_i' = Rp_i + t$，$c(i)$ 为变换后源点所在体素。NDT 有两个突出优势。其一，目标函数对 T 分段连续可微，可用 Newton 类方法直接优化，**无需维护显式的点对点对应列表**；体素索引可通过网格地址直接定位候选分布，从而把“对应搜索”转化为“查表 + 局部评估”。其二，对点云稀疏性更鲁棒：协方差矩阵将局部几何建模为连续分布，能够在一定程度上缓解最近邻在稀疏区域的歧义配对。为降低体素边界导致的梯度不连续，[24] 采用四个互相偏移的重叠格网以平滑代价景观。

NDT 的工程说服力来自它在“没有里程计”的真实数据上仍能跑得动。[24] 记录了一段室内行走数据：机器人 20 分钟行进约 83 m、共采集 28430 帧 2D 激光扫描（SICK，180° 视场，1° 角分辨率），实验中为模拟更高速度只取每 5 帧 1 帧；在 1.4 GHz 机器的 Java 实现里，单帧 NDT 构建约 10 ms、一次 Newton 迭代约 2 ms，离线处理全序列用时 58 s（约 97 scans/s）。这些数字背后对应的是它把“对应搜索”从最近邻查询替换成“体素索引 + 解析梯度/Hessian”，从而能用更少的内存随机访问换来更稳定的吞吐。

NDT 与 GICP [26] 的本质差异在于不确定性建模的粒度：GICP 为**每个点**赋予协方差矩阵（设为各向同性时退化为 P2P，设为平面方向时退化为 P2Pl），NDT 则对**整个体素**估计一个协方差；两者共同构成了” 以何种分辨率建模目标几何不确定性” 这一设计轴的两端。

[26] 的实验设置也能看出它更接近“点级概率模型 + 仍保持 ICP 的工程骨架”：对应仍用欧氏最近邻（便于 KD-tree），但在更新步里把两侧局部平面结构都写进协方差里，试图从“点到面”走向“面到面”。论文在对比中把标准 ICP 的迭代上限设为 250，而点到面与 GICP 只跑到 50 次迭代（收敛更快，且更容易在这个预算内拉开差别）；真实数据还包含 Velodyne 车载扫描的配对示例，描述为两帧扫描约 30 m 间隔、量测范围约 70–100 m 的室外场景 [26]。这些细节并不直接等价于“误差提升多少”，但它们指向 GICP 的两类收益：同样的迭代预算下更快进入稳定区；对最大匹配距离阈值 d_{max} 的敏感性下降，使参数更好调。

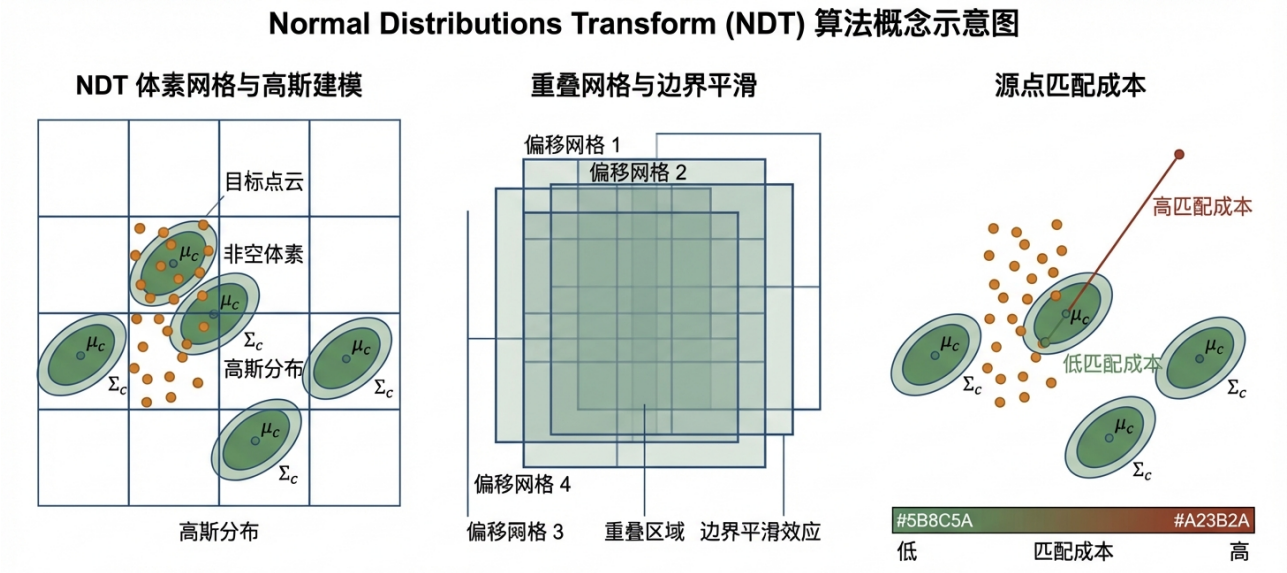


图1：NDT 算法的关键组件。(左) 目标点云的体素化和高斯建模。(中) 使用多个偏移网格来平滑边界效应。(右) 源点相对于目标高斯分布的匹配成本评估。

图 10: NDT 对应建立原理。（左）目标点云被均匀体素格网划分（灰色网格），每个非空体素内点集拟合为二维高斯分布（蓝色椭圆），椭圆形状反映局部几何的各向异性；（中）Biber & Straßer 采用的四重偏移重叠格网（四种颜色），消除体素边界处的梯度跳变；（右）源点（红星）在体素格网中的匹配代价：落在高斯椭圆内部的点代价低（匹配概率高），远离中心的点代价高，优化目标是最大化所有源点的总匹配概率。

3.1.4 特征加权对应（Geometric Feature Optimization）

标准最近邻以纯欧氏距离确定对应，在点密度不均或曲率变化剧烈的区域容易引入大量误配。典型例子是平坦区域：点在切向方向天然“模糊”，“距离近”并不等价于“几何约束强”。**GFOICP**（Geometric Feature Optimized ICP）由 [31]（IEEE TGRS 2023）提出，在对应建立全流程中引入**法向量角、曲率、点间距**三类几何特征，通过采样、匹配、过滤三层机制将“几何可靠性”融入目标函数。

采样层：以三类特征的交叉熵统计量筛选几何稳定的注册点，剔除平坦区域的冗余采样，保留曲率/法向突变的高信息量区域。**动态匹配层**：为避免“初期阈值过严导致找不到对应”，作者把距离阈值做成迭代自适应：第 I 次迭代的阈值 $d_{TH}^{(I)}$ 取上一轮中对应距离的最大值，用它来过滤本轮过远的候选对应 [31]。**特征过滤层**：以 Sigmoid 函数把几何特征相似度映射为软权重 w_i，再将 w_i 带入目标函数：

$$\mathcal{E}_{\mathrm{GFOICP}} = \sum_i w_i \left\| R p_i + t - q_j^* \right\|^2 \tag{13}$$

权重函数让“看起来更像一回事”的对应更主导更新：几何特征相似度高的点对权重大，平坦或模糊区域的点对即使距离近也会被压低。论文在 Bunny 的噪声实验里给了一个很直观的量化切片：固定 $k = 8$ 时，把阈值系数 δ 从 0.05 增到 0.2，会把源/目标的注册点数从 821/1513 拉到 7707/11078，对应运行时间从 0.541 s 增到 0.840 s，但旋转/平移误差（表头标注为 $\times 10^{-3}$）可从 2.153/0.196 降到 0.099/0.011（原文表 II）[31]。这类结果的含义很明确：GFOICP 的收益并不靠“更复杂的求解器”，而是靠把算力投入到更有效的约束点上；代价也同样清晰，三类特征估计需要 k 近邻，在极稀疏点云或对实时性极敏感的系统里会成为预处理瓶颈。

3.1.5 语义引导对应（Semantic Correspondence）

仅依赖几何距离的对应，在动态场景（行人、车辆）或高度重复结构（走廊、隧道）中往往难以区分“近”与“对”。例如，行人点云的几何最近邻可能落到路面上；走廊墙面的最近邻也可能指向几何上相似、但语义上并不对应的位置。**BSC-ICP**（Bivariate Semantic Correntropy ICP）由 [32]（*Fundamental Research* 2025）提出，在统一框架内融合语义标签、双向距离约束与最大相关熵准则（Maximum Correntropy Criterion, MCC）。

对应建立可概括为三步。第一步，点 p_i 仅在目标点云中与其语义类别相同的子集内搜索最近邻，将搜索空间从全量 N_q 收缩到同类子集，减少跨类误配。第二步，在正向匹配 $p_i \to q_{c(i)}$ 之后进行反向搜索 $q_j \to p_{d(j)}$，仅保留双向互惠一致的对应对，用互惠约束压制歧义配对。第三步，以 MCC 核加权双向联合目标函数：

$$\max_{R,t} \sum_i \exp\left(-\frac{\|Rx_i + t - y_{c(i)}\|^2 + \phi(s_i - s_{c(i)})^2}{2\sigma^2}\right) + \sum_j \exp\left(-\frac{\|Rx_{d(j)} + t - y_j\|^2 + \phi(s_{d(j)} - s_j)^2}{2\sigma^2}\right) \tag{14}$$

其中 ϕ 为语义惩罚系数，σ 为带宽参数控制核的宽度。MCC 核对大残差的权重指数衰减，因此无需手动设定硬性外点阈值，噪声点与离群点常会因残差过大而被自然压低权重。[32] 在语义数据集与行业场景数据上报告了更高的成功率以及更小的 RTE/RRE。其限制也很直接：语义引导对应依赖上游点云分割结果，分割误差会传导为配准失败或偏置。

[32] 给出了两组能直接对上“语义约束到底值不值”的表格：在 Semantic-KITTI 上，他们报告 BSC-ICP 的 recall 为 95.7%，RTE/RRE 为 0.07 m / 0.24°，单次配准耗时 498.5 ms；对照的 ICP(P2P) recall 只有 14.3%（472.2 ms），ICP(P2Pl) 为 33.5%（461.7 ms），FGR 为 39.4%（506.1 ms）（原文表 1）。在自建煤矿数据上，BSC-ICP 的 recall 为 93.1%，RTE/RRE 为 0.02 m / 0.19°，耗时 501.5 ms；FGR 的 recall 为 78.5%，ICP(P2Pl) 为 57.9%（原文表 2）[32]。这些数字有个容易被忽略的点：BSC-ICP 的时间几乎和传统 ICP 在一个量级，差别主要体现在“成功率”上，语义把那些几何上很近、但物理上不该配到一起的对应（例如车对路面、人对背景）挡在搜索阶段之外，避免了后端优化在错误对应上越迭代越偏。

3.1.6 对应唯一性、双向一致性与过滤策略

上述各策略均面临一个共同问题：即使找到了” 最近” 对应，在对称结构、重复图案或部分重叠场景中该对应仍可能存在歧义。三类过滤机制可进一步提升对应质量。

双向一致性(Bidirectional/Reciprocal Consistency)要求对应 (p_i, q_j) 满足互惠条件:$q_j = \arg\min_q \|p_i - q\|$ 且 $p_i = \arg\min_p \|q_j - p\|$，即双向最近邻均指向对方。这一约束又称"Picky ICP"[3]，在对称形状和重复几何场景中可显著降低歧义配对率；代价是每次迭代需执行额外的最近邻搜索，计算开销会增加。

距离阈值过滤用硬阈值 $d_{\max}$ 直接丢弃距离过大的对应。[23] 系统对比了硬距离阈值与比例截断策略(TrimmedDist：保留最近 ρ 比例的对应)，指出 TrimmedDist 对尺度变化更稳健，但需要预先给出重叠率的先验。**法向一致性过滤**主要用于抑制跨平面误配：当对应点对两侧（reading 与 reference）估计的表面法向夹角超过 $\theta_{\max}$ 时，直接丢弃该对应；[23] 在其 7-floor mapping 的应用链中取 $\theta_{\max} = 45°$，用来避免跨楼层的错误匹配。工程实现里也常在 30°–45° 之间取值。

法线空间均匀采样（Normal-Space Sampling）针对采样策略的一个常见问题：随机采样虽高效，但采样点可能集中在法线方向相近的区域，从而对部分旋转自由度约束不足。[13] 提出在法向方向空间中做均匀抽取，使各旋转分量都有相对均衡的约束来源。该策略在含大片平坦区域的近似平面网格（如铭刻表面、光滑金属零件）上尤其有效，常能以更少迭代更快收敛 [13]。

这篇工作还把“采样是否只是工程技巧”说得很具体：他们在 Wave、Fractal landscape、Incised plane 三个合成场景上做了受控对比（每个场景约 100k 点），每轮迭代固定抽取 2000 个点做匹配与更新；并在 550 MHz Pentium III Xeon 的 C++ 实现上展示，通过组合投影匹配、点到面误差与法线空间采样等变体，两帧 range image 的配准可以压到几十毫秒量级 [13]。这里的要点不是“某个参数取多少”，而是它把速度瓶颈拆成了可控的几个部件：采样率决定每轮算多少，匹配方式决定访存模式，误差度量决定要不要额外算法向与 Jacobian。

表 5: ICP 对应关系建立策略对比

策略类型	代表方法	对应度量	优点	局限
点到点	Besl-McKay ICP	$\|Rp_i + t - q_j\|^2$	实现简单，无需预计算	光滑曲面收敛慢
点到面	Chen-Medioni ICP	$(n_{q_j}^\top (Rp_i + t - q_j))^2$	收敛更快 [23]	依赖目标法向量质量
对称 ICP	Rusinkiewicz 2019	双侧法向平均方向投影	更宽收敛域，同等计算量	需双侧法向量估计
点到线	2D LiDAR ICP	$d(p_i, l_j)^2$	无需三维法向量	局限于结构化 2D 场景
点到分布	NDT	$-\exp(-\Delta^\top \Sigma^{-1} \Delta/2)$	无显式点对列表，适合稀疏点云	体素分辨率敏感
特征加权	GFOICP	几何特征 Sigmoid 权重	高信息区域强化约束	特征估计额外开销
语义引导	BSC-ICP	MCC + 语义类别一致性	动态场景鲁棒	依赖点云分割精度

对应策略的选择与场景特性密切耦合: 结构化室内 2D 环境首选 P2L, 稀疏车载 LiDAR 场景优先考虑 NDT 或对称 ICP，含动态目标的开放场景需 BSC-ICP 的语义过滤。外点处理与截断对应的系统分析见第 3.2 节 。

3.2 外点处理与鲁棒性 (Outlier Handling & Robustness)

外点（outlier）是点云配准中最常见的干扰来源：它们可能来自传感器噪声与遮挡、两帧之间的非重叠区域、动态目标（行人/车辆），或由几何重复导致的错误对应。在标准 ICP 的最小二乘目标中，少量大残差就可能主导梯度方向，使优化偏离正确解并落入错误的局部极小值。本节按“外点如何被识别与抑制”的机制梳理六类代表性鲁棒化方案，从截断估计与连续降权，到引入物理先验与图论一致性剪枝。

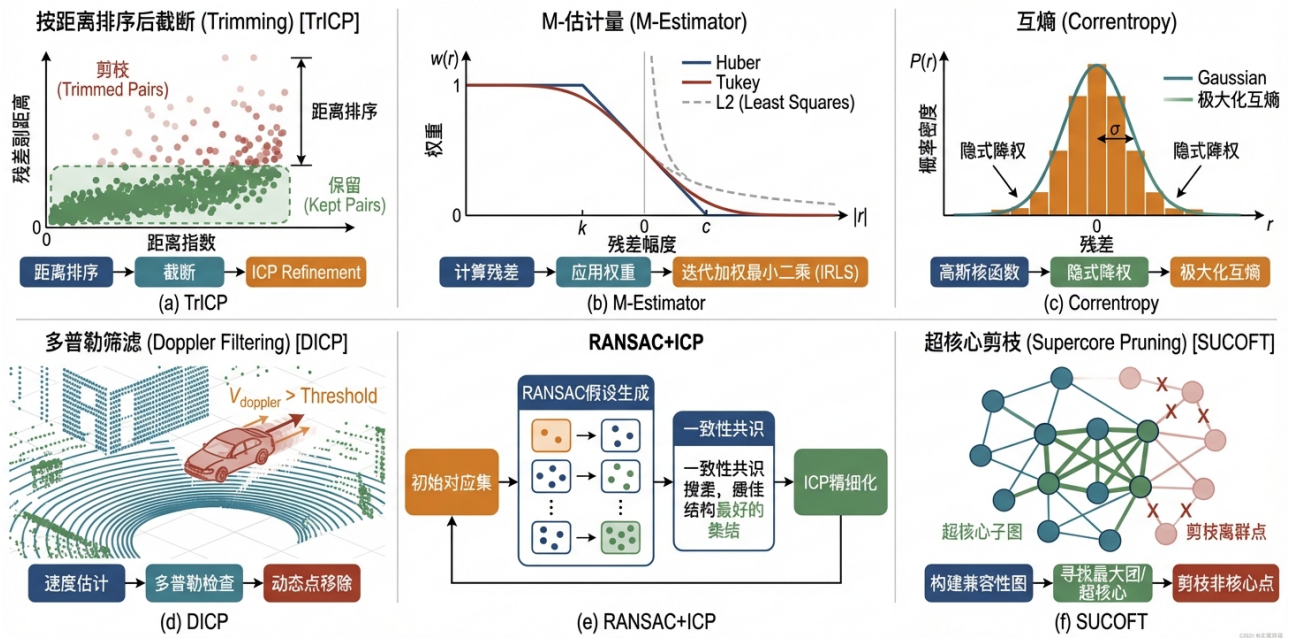


图 11: 六类外点处理策略概览示意。(a) TrICP：截断距离最大的对应；(b) M-估计量：对大残差连续降权；(c) 互熵：以核函数隐式抑制离群残差；(d) DICP：引入多普勒先验剔除动态点；(e) RANSAC+ICP：先粗假设筛内点再局部精修；(f) SUCOFT：在兼容性图上保留一致性强的对应子集。

3.2.1 截断 ICP：TrICP

截断迭代最近点算法（Trimmed ICP，TrICP）由 [14] 于 2002 年提出，核心思想是将最小截断二乘（Least Trimmed Squares，LTS）估计器系统地应用于 ICP 的各个步骤。给定目标重叠率参数 $\rho \in (0, 1]$，算法在每次迭代中仅保留距离最小的 $\lfloor \rho n \rfloor$ 个点对，其目标函数为

$$\mathcal{E}_{\text{TrICP}} = \sum_{i=1}^{\lfloor \rho n \rfloor} d_{(i)}^2 \tag{15}$$

表 6: 第 3.1 节代表性“可复现设置 + 定量结果”汇总（仅摘录文中明确报数且口径清晰的结果）。

文献	场景/数据集	指标口径	结果（数值）	关键设定（便于复现）
[1]	点集 / 曲线 / 曲面（论文实验小例子）	迭代次数、初值枚举规模、端到端用时与 RMS	点集：8 点对 11 点，6 次迭代，<1 s；曲线：折线 64 点 + 高斯噪声，12 旋转 × 6 平移初值，约 6 min；mask 曲面：模型 2546 点，24 个初始旋转，每个初值 6 次迭代，约 10 min 得到 RMS 0.59	以“先枚举初值，再跑局部 ICP”的方式做全局化尝试；
[23]	“Challenging Laser Registration”6 场景（Apartment、Stairs、ETH、Gazebo、Wood、Plain）	单次配准总耗时（达到终止条件）	P2P：中位 1.45 s；P2Pl：中位 2.58 s（单核 2.2 GHz Core i7）	`MinDist`（去 1 m 内点）+ `RandomSampling`（reading 保留 5%；P2P 参考端也保留 5%；P2Pl 参考端 `SamplingSurfaceNormal` 约 7×、阈值 7 点）+ KD-tree（$\epsilon = 3.16$）+ `TrimmedDist`（P2P 75%，P2Pl 70%）+ 终止条件（≤150 次或 $\Delta t < 1$ cm、$\Delta r < 0.001$
[2]	多视角范围图建模（ICRA 1991 版实验）[29]	视角数、旋转步长（影响重叠与可配准性）	侧面 8 视角 + 顶/底 6–8 视角；相邻侧视角旋转间隔约 45°	点到面距离最小化（避免显式
[30]	Bunny 扫描对（`bun000` vs `bun090`，IOU 约 23%）	成功率：在给定迭代预算内成功收敛的比例	每个“初始旋转角 × 初始平移幅值（按模型尺寸归一化）”网格点采样 1000 个随机初值；分别统计在 20/100/500 次迭代内的成功率热图（原文图 5）	固定对应搜索 + 三种目标（P2P/P2Pl/对称）；
[24]	室内真实 2D 激光序列	处理吞吐（scans/s）与端到端耗时	28430 帧、20 min、约 83 m；离线处理 58 s（约 97 scans/s），NDT 构建约 10 ms/scan，Newton 单次迭代约 2 ms（1.4 GHz，Java）	SICK 180°，1° 分辨率；只取每 5 帧 1 帧；无里程计初始化（用上一
[31]	Bunny（噪声实验；原文表 II）	旋转误差 ER(rad)、平移误差 Et(m)（表头标注为 $\times 10^{-3}$）与时间	例：$k = 8, \delta = 0.05$：ER 2.153、Et 0.196、0.541 s、注册点 821/1513；$k = 8, \delta = 0.2$：ER 0.099、Et 0.011、0.840 s、注册点 7707/11078	通过 k 邻域估计法向/曲率；交叉熵 几何特征相似度 Sigmoid 过滤
[32]	Semantic-KITTI + 煤矿场景	RTE(m)/RRE(deg)/Recall/Time(ms)	Semantic-KITTI：Ours 0.07/0.24/95.7%/498.5 ms；ICP(P2P) 0.04/0.11/14.3%/472.2 ms；FGR 0.93/0.96/39.4%/506.1 ms。煤矿：Ours 0.02/0.19/93.1%/501.5 ms；FGR 0.06/0.28/78.5%/491.6 ms	语义同类搜索 + 双向互惠 + 在论文中给出经验范围并有消
[26]	模拟扫描 + Velodyne 实测扫描对	迭代预算与场景尺度（实验设置层面）	标准 ICP 最大 250 次迭代；点到面与 GICP 最大 50 次迭代；真实示例包含两帧扫描约 30 m 间隔、量测范围约 70–100 m 的室外点云对	对应仍用欧氏最近邻（KD-tre 友好）；更新步使用双侧局部平 d_{max} 参数敏感性降低

其中 $d_{(1)} \le d_{(2)} \le \cdots$ 是排序后的点对距离。与标准 ICP 相比，TrICP 在 $\rho = 1$ 时退化为原始算法；当 $\rho < 1$ 时，被截除的点对不贡献梯度，优化仅利用保留下来的内点对。[14] 证明了该算法单调收敛于局部极小值，并在部分重叠、含外点的设置下验证了截断估计对配准稳定性的提升。

从求解过程看，TrICP 的改动不在位姿更新器，而在于对应集的筛选机制：每次迭代在建立最近邻对应后，先按残差升序排列，仅保留最小的 $\lfloor \rho n \rfloor$ 对，再以该内点子集执行标准刚体配准。从优化角度看，这等价于将外点抑制建模为一个离散选择问题。其优势在于物理含义明确：当两帧重叠率约为 70% 时，设定 $\rho = 0.7$ 即可将非重叠区域的点对系统性地排除在目标函数之外。其局限同样来自这一离散化本身：截断边界处的硬跳变使目标函数在边界附近不连续，两个残差相近但分处边界两侧的点对权重可能分别为 1 和 0，对优化轨迹的平滑性造成影响。

这篇论文给了两组“难度梯度”非常清晰的报数。一组是 3D Frog 数据：两个点集各约 3000 点；作者把标准 ICP 视作 $\rho = 1$ 的特例，对比了 ICP（45 次迭代，MSE=5.83，耗时 7 s）与 TrICP（设定重叠率 70%，88 次迭代，MSE=0.10，耗时 2 s）（原文表 1，1.6 GHz PC）[14]。另一组是 SQUID 鱼轮廓库的 1100 个 2D 形状：通过控制旋转角（1°/5°/10°/15°/20°）与重叠率（100%→60%）构造部分重叠场景，并用“估计旋转与真值旋转的平均绝对误差（deg）”衡量稳健性。以最难的 20° 旋转、60% 重叠为例，TrICP 的误差为 1.7949°，而 ICRP 为 3.0254°（原文表 2、表 3）[14]。这组数据对应的是一个明确边界：当外点主要来自非重叠区域时，TrICP 能把更新重新限制在重叠区域内；但一旦错误对应混入被保留的那一段，硬截断本身并不会再区分“保留下来的内点”和“恰好没被截掉的伪内点”。

TrICP 的参数 ρ 直接对应先验重叠率估计，物理含义清晰；但 ρ 固定意味着截断边界不随迭代动态调整，若重叠区域几何多变，可能在某些帧出现截断过多或截断不足。

3.2.2 M-估计量与稀疏 ICP

M-估计量将标准最小二乘目标替换为对大残差施加次二次惩罚的鲁棒核函数，从而隐式地压低外点权重：

$$\mathcal{E}_{\text{robust}} = \sum_{i=1}^{n} \rho_M\big(\|Rp_i + t - q_{j(i)}\|\big) \tag{16}$$

常用核函数包括 Huber 核 $\rho_H(r) = r^2/2$ ($|r| \le \delta$) 和 Tukey 双权核 $\rho_T(r) = c^2[1 - (1 - r^2/c^2)^3]/6$ ($|r| \le c$，否则为常数 $c^2/6$)。Tukey 核对超过截断半径 c 的残差贡献零梯度，等效于完全忽略极端外点；与 TrICP 的硬截断不同，这种忽略是通过梯度逐渐衰减到零实现的。

M-估计量与 TrICP 的根本区别在于：前者以连续降权取代离散截断，不对对应集做二元取舍，而是通过权重函数连续地调制每个对应的贡献。将鲁棒核改写为等价的迭代加权最小二乘形式：

$$\mathcal{E}_{\text{robust}} \approx \sum_{i} w_i(r_i)\, r_i^2,$$

其中 $w_i(r_i) = \rho'_M(r_i)/(2r_i)$。残差较小的点对保留接近 1 的权重，残差较大的点对权重随之降低，但对应点仍然参与目标函数的构造。从数值上看，这一连续降权过程比 TrICP 的截断更平滑，对梯度类优化器更为友好；但核函数尺度参数需要仔细选择，带宽过大则退化为普通最小二乘，过小则可能将真实内点的贡献一并压低。

[33] 提出的 FRICP（Fast and Robust ICP）系统地将渐进非凸（Graduated Non-Convexity，GNC）框架引入 ICP：从接近凸的目标函数出发，逐步收紧鲁棒核，使优化轨迹更不易被早期外点“带偏”，从而改善部分重叠与外点条件下的配准稳定性。[33] 在五组部分重叠模型对（含 Bunny、Dragon 等）上以 Super4PCS 先做粗对齐，并在点集上叠加不同比例的随机外点；以 Bunny 为例，其鲁棒点到点方案的平均/中位 RMSE（$\times 10^{-3}$）为 0.85/0.69、耗时 0.34 s，而 Sparse ICP 为 0.94/0.71、耗时 24.06 s，误差同量级但运行时间相差约两个数量级 [33]。

[15] 从另一角度出发，将配准目标显式改写为 ℓ_p 范数（$p \in [0,1]$）最小化问题：

$$\mathcal{E}_{\text{Sparse}} = \sum_{i=1}^{n} \|Rp_i + t - q_{j(i)}\|_2^p, \quad p \in [0,1] \tag{17}$$

当 $p \to 0$ 时目标趋近于计数内点数，自然获得稀疏性；$p = 1$ 退化为 ℓ_1。求解采用 ADMM（交替方向乘子法），保留了 ICP 交替估计的基本框架 [15]。

与 M-估计量相比，Sparse ICP 在外点建模的层次上更为深入。M-估计量在最小二乘框架内通过调整权重实现软鲁棒化，目标函数本质上仍属于光滑优化范畴；Sparse ICP 则直接在目标层面引入 ℓ_p 稀疏惩罚，将“仅有少数对应承担大残差”这一先验以范数约束的形式编码进去。当 $p < 1$ 时目标函数明显非凸，不再适合基于加权更新的闭式求解，因此论文采用 ADMM 将刚体变换更新与稀疏残差更新交替求解。二者的本质区别在于：M-估计量在目标函数的形式上保留了可微结构，Sparse ICP 则通过 ℓ_p 范数将外点稀疏性明确写入优化问题本身。

[15] 在“owl”虚拟扫描的对齐实验里把这个思想量化得很直接：他们用相对于真值的 RMSE（记为 e）评估配准质量，并对比了“阈值剔除到底该取多大”这一最常见的工程困境。原文图 4 的报数里，粗初值为 $e = 4.0\times10^{-1}$；传统 ℓ_2 ICP（$p = 2$）配合距离阈值剔除时，$d_{th} = 5\%$ 仍为 4.1×10^{-1}，$d_{th} = 10\%$ 可到 2.9×10^{-2}，但 $d_{th} = 20\%$ 又回升到 7.5×10^{-2}（d_{th} 以包围盒对角线的百分比定义）；ℓ_1（$p = 1$）做到 1.6×10^{-2}；而 ℓ_p（$p = 0.4$）进一步降到 4.8×10^{-4}[15]。上述实验结果揭示了一个具有实践意义的结论：阈值剔除方法的效果对场景尺度与初值质量高度敏感；ℓ_p 稀疏范数将硬截断转化为连续的自适应降权，从而消除了阈值参数对场景的依赖性。作者还在附录中指出其 shrink 更新通常 2–3 次迭代即可收敛，这也是它能保持可用速度的重要原因之一 [15]。

3.2.3 互熵 ICP (Correntropy-based ICP)

互熵（Correntropy）来源于信息论，其定义为

$$C_\sigma(X,Y) = \mathbb{E}\left[k_\sigma(X-Y)\right] = \frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{\|Rp_i + t - q_{j(i)}\|^2}{2\sigma^2}\right) \tag{18}$$

其中 σ 为高斯核带宽。最大化互熵等价于最小化一个以高斯衰减为权重的加权 ℓ_2 损失：残差越大，高斯权重越小，外点贡献随之被自然压低，无需手动设置硬截断阈值 [34]。与 Tukey 核相比，互熵核的权重随残差增大单调下降但不硬截为零，更偏向在统计鲁棒性与数值稳定性之间取折中。

在权重衰减方式上，互熵核与其他鲁棒核存在本质差异。Huber 核对小残差施加二次惩罚、对大残差施加线性惩罚；Tukey 核在阈值之外权重归零，等效于完全忽略极端外点；互熵核则对应高斯型权重，随残差增大呈指数衰减但不硬截为零，在统计鲁棒性与数值稳定性之间取得折中。这一性质在含长尾噪声的场景中具有优势，可避免将边缘内点误判为外点而彻底丢弃。代价是带宽参数 σ 具有更为统计化的含义：它不仅是几何距离阈值，而是刻画“多大偏差仍属于同一分布内的正常波动”的尺度参数。

[34] 在此基础上进一步提出带尺度参数的 SCICP（Scale Correntropy ICP），在相似变换框架下将各向同性缩放因子一并纳入优化，使方法可覆盖“存在尺度偏差”的配准情形（例如跨传感器标定误差或重建尺度漂移）[34]。

SCICP 的优势主要体现在“外点很脏、又有尺度差”的组合场景里。[34] 在 2D CE-Shape-1 数据库的 Apple/Pocket/Ray 三组点集上对比了 Scale ICP、CPD 与 SCICP，并分别报告尺度误差 ε_s、旋转误差 ε_R 与平移误差 $\varepsilon_{\vec{t}}$：例如 Apple 上，SCICP 的 $\varepsilon_s = 0.0020$、$\varepsilon_R = 9.0351\times10^{-4}$、$\varepsilon_{\vec{t}} = 0.0817$，而 Scale ICP 为 0.0687/0.3031/80.0321，CPD 为 0.1061/0.0889/32.0936（原文表 1）；耗时方面，Apple 上 Scale ICP 为 0.0022 s，SCICP 为 0.0083 s，CPD 为 0.0462 s（原文表 2）[34]。在 3D 仿真（Happy/Bunny/Dragon）里，SCICP 同样在 $\varepsilon_s, \varepsilon_R, \varepsilon_{\vec{t}}$ 上整体小于 Scale ICP，例如 Dragon 的 $\varepsilon_{\vec{t}}$ 从 0.0196 降到 6.8352×10^{-5}（原文表 3）[34]。这里的关键不是“核函数更复杂”，而是尺度项一旦先被坏对应拉偏，后面的旋转和平移就会跟着偏；互熵核先压低大残差的权重，等于先稳住尺度，再让三者的耦合更新继续进行。

3.2.4 Doppler ICP (DICP)

DICP 是针对 FMCW LiDAR 等能够逐点测量瞬时径向速度的传感器设计的 ICP 变体。传统 ICP 在走廊、隧道等几何结构重复的环境中极易产生退化（沿对称轴的平移不可观），因为纯几何目标函数缺乏沿此方向的约束。DICP 引入了多普勒残差项：

$$\mathcal{E}_{\text{DICP}} = \underbrace{\sum_i \|Rp_i + t - q_{j(i)}\|^2}_{\text{几何残差}} + \lambda \underbrace{\sum_i (v_i - \hat{v}_i(\boldsymbol{\omega}, \mathbf{v}))^2}_{\text{Doppler 残差}} \tag{19}$$

其中 v_i 为点 p_i 的实测径向速度，$\hat{v}_i$ 为由当前估计的角速度 $\boldsymbol{\omega}$ 和线速度 $\mathbf{v}$ 预测的径向速度，λ 为两项的相对权重。多普勒速度从独立的物理量出发约束运动估计，有效打破了退化对称性 [35]。

与前述基于残差降权或截断的外点抑制策略不同，DICP 的核心思路在于引入独立物理量作为先验，在对应建立阶段主动识别动态目标并将其排除，而非在优化阶段通过权重调整加以补偿。TrICP、M-估计量与互熵 ICP 均在几何残差层面处理外点；DICP 则利用多普勒速度测量直接识别动态点的运动异常，在对应集构建阶段即予以剔除。在走廊、隧道等几何结构沿某方向近似一维的场景中，纯几何 ICP 无法区分真实运动与等价对应假设；多普勒速度提供了独立于几何形状的运动约束，从而打破了退化对称性。

DICP 的另一贡献是利用多普勒测量识别动态目标：若某点的测量径向速度与当前运动估计的预测值差异显著，该点很可能来自行驶车辆或行人，算法将其从对应集中剔除。[35] 在 Aeva Aeries I FMCW LiDAR 的 5 段真实序列与 CARLA 的 2 段仿真序列上，与 Open3D 的经典 P2Pl ICP 做对比：在 Baker-Barry Tunnel 等“特征贫乏 + 重复结构”场景中，基线 ICP 的平移 RPE 处于米级（>1 m），而 DICP 可降至厘米级（<0.1 m），同时路径误差从 525.35 m 降至 1.23 m；在含大量动态车辆的 Brisbane Lagoon Freeway，基线路径误差高达 4337.18 m，而 DICP 为 4.16 m。作者还报告 Doppler 约束可显著减少迭代次数，例如 Baker-Barry Tunnel 的平均迭代次数由 30.8 次降至 7.6 次（Robin Williams Tunnel：44.3 次降至 13.1 次），体现了“物理量约束 + 动态点剔除”对退化与动态外点的双重抑制 [35]。

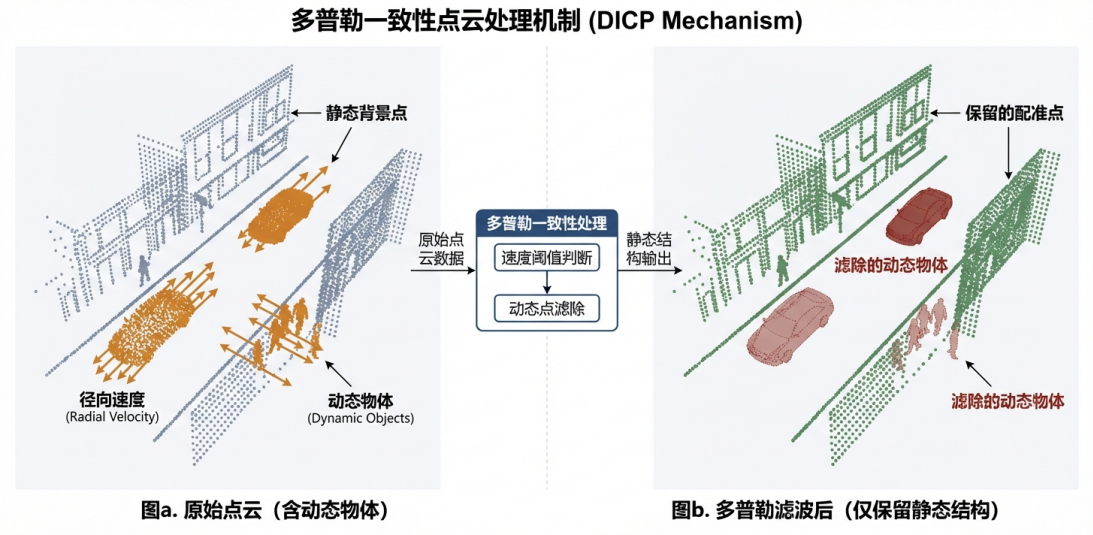


图 12: DICP 动态目标筛滤示意。左：原始点云，行驶车辆（红色）的多普勒测速与静态背景（蓝色）明显不同。右：基于多普勒阈值剔除动态点后，仅静态背景参与 ICP 优化，消除了动态污染对配准结果的影响。

3.2.5 RANSAC-ICP 混合策略

RANSAC（Random Sample Consensus）与 ICP 的结合属于“先把内点挑出来，再让局部优化做精”的两阶段思路：先用 RANSAC 从重外点对应集中采样生成位姿假设，以共识集大小或鲁棒代价评估假设质量；再用 ICP 在该假设附近做局部精修。这个套路的价值在于，它把 ICP 最怕的两件事拆开处理：初值差的问题让前端负责把解推到可收敛区域；局部精度则交给 ICP 的几何最小二乘去榨干。与 RANSAC 同属“外点极高时仍能先给出可用解”的代表还有 TEASER/TEASER++：论文不仅在摘要中强调已知尺度时可容忍超过 99% 外点，还在 3DMatch 上给出逐场景成功率与时间对比（原文表 II）：TEASER++ 在 Kitchen/Home/Hotel 等 8 个场景的正确配准率为 83.1%–98.6%，平均单次运行时间 0.059 s；RANSAC-1K 仅 74.5%–94.2%（0.008 s），RANSAC-10K 为 79.3%–97.2%（0.074 s）[36]。工程上常见的组合是“TEASER++ 给全局初值 + ICP 做最后一公里”，用 TEASER++ 的鲁棒性换一个更稳的起点，再把最后的精度交给局部优化。

RANSAC 的主要代价是当外点比例升高时，为获得足够高置信度所需的假设数量会迅速增大，这一点在系统评测里非常直观。[37] 在 U3M/BMR/U3OR/BoD5 四个数据集上评测了 14 种 RANSAC 风格估计器，并专门在 U3M 上注入可控干扰：高斯噪声标准差从 0.5 pr 到 3.0 pr（步长 0.5 pr）；均匀/随机降采样保留比例从 80% 一直降到 10%；孔洞数量从 6 增到 26（每个孔洞通过 KNN 删除邻域点合成，邻域规模取 $2\% \times |\mathbf{P}^t|$）（原文图 7）。他们还展示了对应集本身的难度差异可以很大：示例对应集的内点率从 56.41%（117 对）到 13.17%（129 对）不等（原文图 6）[37]。这些“外部条件”的数字能解释很多现象：当内点率和空间分布一旦走坏，采样策略、局部参考系重复性、以及假设评估指标的计算代价，就会一起决定 RANSAC 是否还能在可用时间内筛出足够好的初值。

RANSAC 与本章前述外点处理方法的分工可从算法层次加以理解：TrICP、M-估计量、互熵 ICP 等均以“当前位姿估计已接近真值”为前提，旨在避免异常对应对局部优化的干扰；RANSAC 则工作在更前一个层次，解决的是“候选对应集是否包含足够的内点以生成可用初值”这一问题。当候选对应集质量极差时，任何基于局部收敛的 ICP 变体均难以获得正确解。这一设计的代价在于：随着外点比例升高，为保证以目标置信度找到一个纯内点假设，所需的采样次数呈指数增长，高外点率下的运行时间代价显著。

3.2.6 SUCOFT：超核心最大化

SUCOFT（Supercore Maximization with Flexible Thresholding）从图论视角处理外点：给定一组候选对应，在兼容性图（compatibility graph）上以“局部刚体一致性”定义边关系，满足一致性的对应对之间连边；算法核心是在该图上求解“最大超核心”（maximum K-supercore）：

$$K\text{-supercore} = \{v \in \mathcal{V} : |\mathcal{N}(v) \cap S| \geq K\} \tag{20}$$

其中 K-超核心是最大子图 S，满足其中每个节点至少与 K 个其他节点相连。[38] 证明最大超核心必然包含最大团（maximum clique），因此在存在噪声与“缺边”的情况下，超核心剪枝往往比最大团更不容易把真实内点过早剔除。在此基础上，SUCOFT 以灵活阈值（flexible thresholding）做后处理进一步精炼对应集，作者报告该精炼在多数设置下仅需 2–3 次迭代即可收敛 [38]。

SUCOFT 与 RANSAC 的方法论差异体现在搜索策略的不同：RANSAC 从参数空间出发，通过随机采样生成变换假设并以共识集大小加以验证；SUCOFT 则在对应空间中，先通过图论一致性分析识别相互几何相容的对应子集，再由此推导变换估计。当外点比例较高时，RANSAC 所需的采样次数急剧增加，而图论剪枝对参数空间的搜索依赖相对较低；但 SUCOFT 的前提是候选对应集中存在足够的高质量内点以构成连通的兼容性子图，否则剪枝操作本身亦难以得到可靠结果。

在多个基准测试中，[38] 在 ETH LiDAR、WHU、Stanford Bunny/Armadillo、3DMatch 与 3DLoMatch 上系统评测了已知尺度与未知尺度两类问题，并报告 SUCOFT 在两类设定下都可容忍超过 99% 的外点。其消融结果显示，SUCOM 在初始外点率为 20%–98% 时可将剩余外点率压到 0%，即便在 99% 的极端外点下，剩余外点率也不超过 10%；在此基础上再由 ROFT 快速收敛，从而把图论一致性剪枝转化为“可落地”的外点抑制链路 [38]。在 3DLoMatch 的已知尺度设置下，SUCOFT 的 registration recall 报告为 43.14%，也反映了其在低重叠室内场景中的实用性 [38]。

3.2.7 各方法综合对比

表 7: 第 3.2 节外点鲁棒化方法对比：代表论文、核心机制、典型外点形态与局限。

方法	代表论文	核心机制	更适合的外点形态	主要局限
TrICP	[14]	距离排序后截断（LTS）	部分重叠导致的非重叠外点	需先验/估计重叠率；截断边界不自适应
M-估计量 / GNC（FRICP）	[33]	连续降权 + 渐进非凸	噪声外点与轻度误配	核带宽与退火策略影响收敛；仍是局部法
Sparse ICP	[15]	ℓ_p 稀疏惩罚 + 变量分裂	对应集中含显著离群残差	非凸；ADMM/近端迭代的收敛与速度依赖超参
互熵（SCICP）	[34]	核函数隐式降权	重尾噪声与长尾离群	带宽选择敏感；不当设置会过度抑制有效约束
DICP	[35]	多普勒先验筛滤 + 额外残差	动态目标引入的系统性外点	依赖具备多普勒测量的传感器；权重融合需标定
RANSAC+ICP	[37]	采样假设 + 共识集筛选 + 局部精修	特征匹配阶段产生的大量误配	高外点时假设数爆炸；非确定性且依赖随机种子
SUCOFT	[38]	兼容性图一致性剪枝（超核心）	对应集外点占优、但内点满足几何一致性	图构建开销大；需可靠的初始候选对应集

表 8: 第 3.2 节代表性“可复现设置 + 定量结果”汇总（仅摘录文中明确报数且口径清晰的结果）。

文献	场景/数据集	指标口径	结果（数值）	关键设定/前提
[14]	Frog（3D，约 3000 点/点集）+ SQUID（2D，1100 形状）	Frog：MSE/时间/迭代；SQUID：旋转误差（deg）	Frog：ICP（45 次，MSE=5.83，7 s）vs TrICP（重叠率 70%，88 次，MSE=0.10，2 s，1.6 GHz PC）；SQUID 最难设置（20°，60% 重叠）：TrICP 1.7949°，ICRP 3.0254°	LTS 截断只保留最小 $\lfloor \rho n \rfloor$ 对应；SQUID 旋转角 1–20°、重叠率 60–100% 的系统性扫描
[33]	5 组部分重叠模型对（示例：Bunny）	RMSE 与耗时（原文表 1，RMSE 单位为 $\times 10^{-3}$）	Bunny：鲁棒点到点 0.85/0.69，0.34 s；Sparse ICP 0.94/0.71，24.06 s	Super4PCS 先粗对齐；在点集上叠加随机外点（论文
[15]	“owl” 虚拟扫描对齐（原文图 4）	RMSE（相对真值）	粗初值：4.0×10^{-1}；$p=2$ + 剔除：$d_{th}=5\%$ 为 4.1×10^{-1}、10% 为 2.9×10^{-2}、20% 为 7.5×10^{-2}；$p=1$ 为 1.6×10^{-2}；$p=0.4$ 为 4.8×10^{-4}	d_{th} 以包围盒对角线百分比定义；$p\in[0,1]$ 的 ℓ_p 残差 + ADMM；附录注明 shrink 更新常 2–3 次迭代收敛
[34]	CE-Shape-1（Apple/Pocket/Ray）+ Stanford 3D（Happy/Bunny/Dragon，仿真）	$\varepsilon_s, \varepsilon_R, \varepsilon_{\bar{t}}$ + 耗时	例：Apple：SCICP $\varepsilon_s=0.0020$、$\varepsilon_R=9.0351\times 10^{-4}$、$\varepsilon_{\bar{t}}=0.0817$；Scale ICP 0.0687/0.3031/80.0321；CPD 0.1061/0.0889/32.0936（原文表 1）。耗时：Apple SCICP 0.0083 s，CPD 0.0462 s（原文表 2）。3D：Dragon $\varepsilon_{\bar{t}}$ $0.0196\rightarrow 6.8352\times 10^{-5}$（原文表 3）	相似变换（含各向同性尺度）；MCC 作为相似度，交替更新对应与变换，并证
[35]	Aeva Aeries I FMCW LiDAR（5 序列）+ CARLA（2 序列）	Path Error / 平移 RPE / 平均迭代次数（原文表 II）	Baker-Barry Tunnel：Path Error 525.35 m→1.23 m；迭代 30.8→7.6。Brisbane Lagoon Freeway：Path Error 4337.18 m→4.16 m	与 Open3D P2Pl ICP 对比；加入 Doppler 残差与动态点剔除（DOR）
[37]	U3M 注入噪声/降采样/孔洞 + 四数据集评测	干扰强度设置 + 对应集统计（实验设置层面）	高斯噪声 0.5–3.0 pr（步长 0.5）；均匀/随机降采样保留 80%→10%；孔洞 6→26（单孔洞删除邻域点数 $2\%\times\lvert\mathbf{P}'\rvert$）；示例对应集内点率 56.41%（117 对）到 13.17%（129 对）	14 种 RANSAC 风格估计器分解到“采样/生成/评估/停
[36]	3D 基准 + 3DMatch（论文实验概述）	外点耐受与量级（论文摘要与实验小结）	已知尺度时可容忍 >99% 外点；TEASER++ 运行在毫秒级	TLS 代价 + 图论剪枝 + 可认证松弛；工程上常用作“全局初值” ICP 精修
[38]	ETH/WHU、Stanford Bunny/Armadillo、3DMatch/3DLoMatch	外点耐受与 RR（消融 + 基准表）	>99% 外点（已知/未知尺度）仍可工作；SUCOM 处理后：20%–98% 外点剩余 0%，99% 外点剩余 ≤10%；3DLoMatch（已知尺度）RR=43.14%；ROFT 多数仅需 2–3 次迭代	先在兼容图上做 SUCOM 大规模剪枝，再用 ROFT 灵活阈值精炼

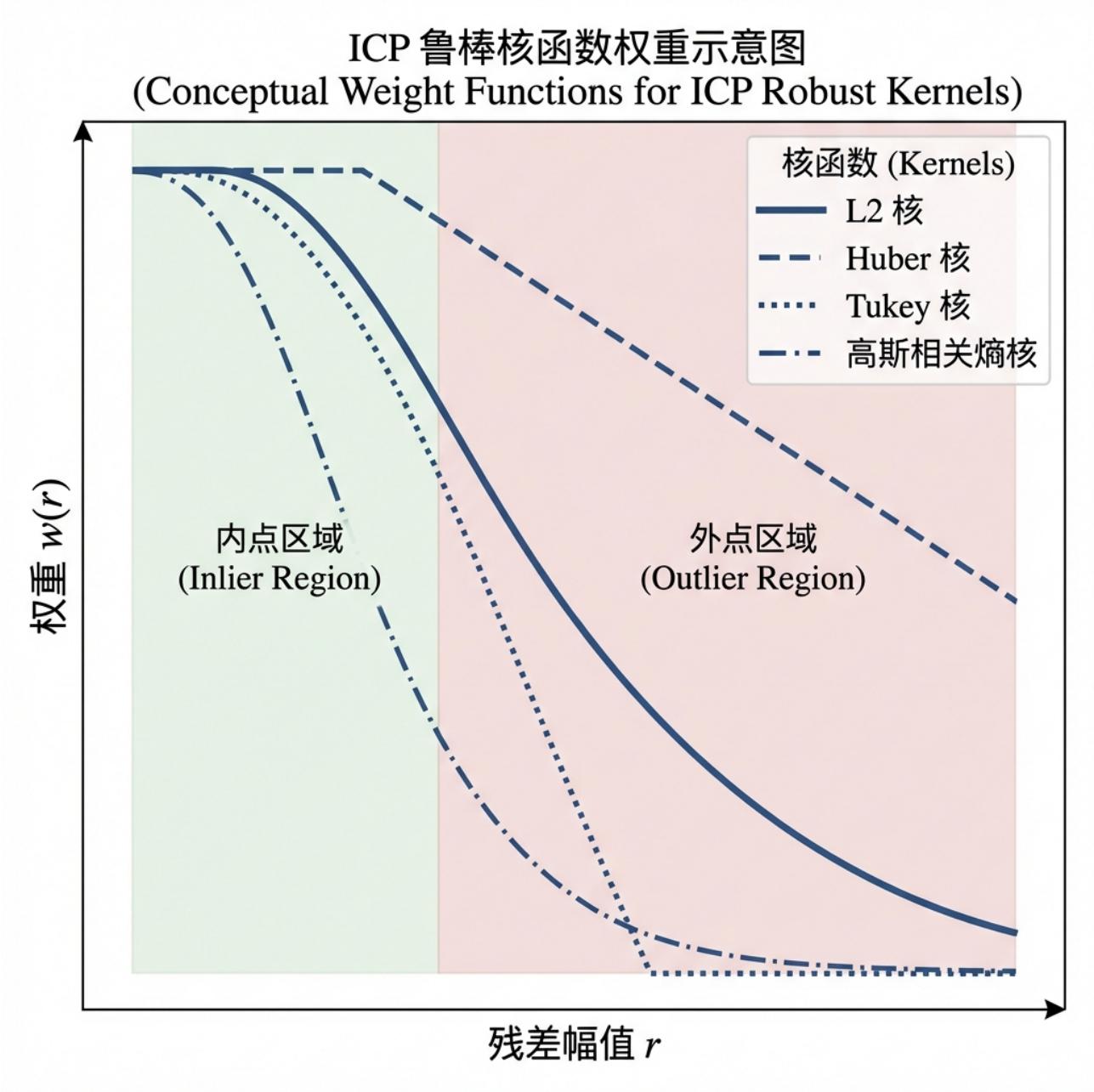


图 13: 四类鲁棒核/权重函数对大残差的抑制方式对比（示意）。$w(r) = \rho'(r)/(2r)$ 将残差大小 r 映射为“该对应对优化的有效贡献”。相较于 ℓ_2（恒定权重），Huber 在大残差处线性降权，Tukey 在阈值外近似归零，而互熵（Gaussian）平滑衰减。

与对应策略（见第 3.1 节 ）的关系在于：外点处理与对应建立相互耦合。TrICP、M-估计量和互熵 ICP 在标准最近邻对应基础上叠加鲁棒化；DICP 则在对应搜索阶段引入多普勒先验主动过滤；而 SUCOFT 工作在特征对应集合上，属于前置外点剔除，独立于 ICP 的几何对应搜索。在实际系统中，常把前置对应过滤（距离/法向量角度阈值，见第 3.1 节 ）与迭代中的连续权重衰减联合使用，以覆盖不同来源的外点。第 3.6 节 将进一步讨论全局初始化如何将初始误差压入可收敛区域，从而减少外点主导的失败模式。

3.3 收敛加速与迭代优化 (Convergence Acceleration and Iterative Optimization)

标准 ICP 属于交替优化的不动点迭代：每一步都以当前估计为线性化中心，重复“建立对应 → 更新位姿”。在噪声、弱约束或收敛方向不均衡的情况下，该迭代可能表现为收敛缓慢或轨迹振荡。[13] 指出许多工程技巧（采样、层次化、终止准则）本质上都在改变“每步更新的有效信息量”。本节从五个维度系统梳理收敛加速方法：（1）Anderson 外推（历史信息复用）；（2）Majorization-Minimization（MM）与渐进非凸（GNC）下的鲁棒加速；（3）速度/运动先验提供更好的初值；（4）多分辨率策略重塑目标函数景观；（5）自适应终止准则。

从单步计算代价看，ICP 的每次迭代仅包含对应搜索与刚体最小二乘求解，计算量有限；然而收敛效率在实践中往往受限于两类原因：其一，更新方向不准确，导致迭代轨迹在错误盆地附近反复振荡；其二，每步引入的有效几何约束量不足，导致残差下降幅度持续偏小。[13] 的实验对此有较为直观的说明：在约 10^5 点规模的网格上，每轮仅采样约 2000 个点（约 1%）参与更新；在 550 MHz Pentium III Xeon 上，初值位于正确收敛盆地时，两幅 range image 的对齐可压至几十毫秒量级。迭代次数、每步计算代价与每步有效约束量三者的共同作用，决定了 ICP 的实际收敛效率。

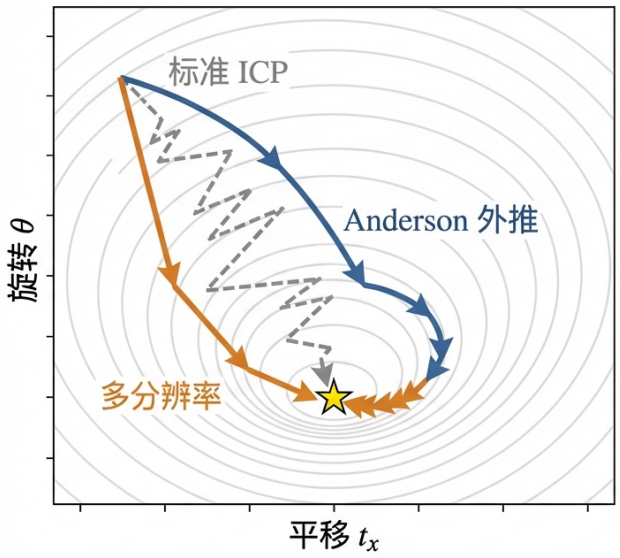


图 1：三种 ICP 方法的收敛路径对比

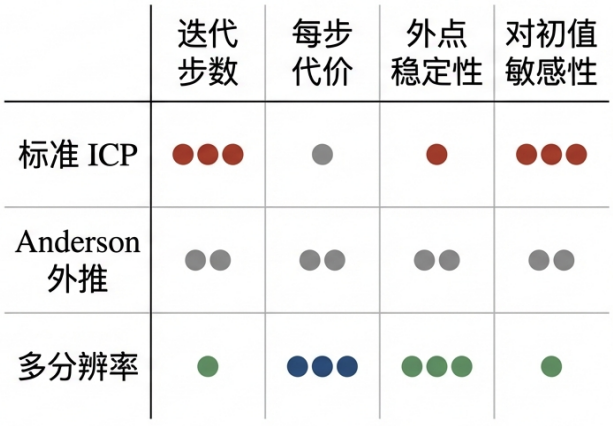


图 2：方法特性定性比较

图 14: ICP 收敛加速策略对比（示意）。左：目标函数等高线图上，标准 ICP（灰色虚线）的锯齿状路径 vs. Anderson 外推（蓝色）的平滑路径 vs. 多分辨率 ICP（橙色）的由粗到精路径。右：不同策略在“迭代步数/每步代价/稳定性”之间的定性权衡：历史外推减少步数、鲁棒 MM 降低外点主导风险、多分辨率扩大可收敛区域但引入多层开销。

3.3.1 ICP 作为不动点迭代：加速的统一视角

ICP 的每次迭代本质上是一个不动点更新。设位姿参数向量 $\boldsymbol{\xi}^{(k)} \in se(3)$（李代数六维向量），” 对应建立 + 变换估计” 的复合映射记为 G，则

$$\boldsymbol{\xi}^{(k+1)} = G\left(\boldsymbol{\xi}^{(k)}\right) \tag{21}$$

标准 ICP 相当于 Picard 迭代——每步只利用 $G(\boldsymbol{\xi}^{(k)})$。其收敛速率（线性收敛）由 G 的 Lipschitz 常数 $\kappa = \|G'\| < 1$ 决定：每次迭代误差以 κ 倍递减。当 κ 接近 1 时（如旋转分量收敛慢，平移分量收敛快的不平衡场景），需要大量迭代。Picard 迭代的低效正是因为它在每次更新时" 忘记" 了历史方向信息。Anderson 加速和 Majorization-Minimization（MM）框架分别从" 外推利用历史" 和" 最优化更紧代理函数" 两条路径解决这一问题。

有必要区分 ICP 迭代低效的两种根本原因。其一，迭代已进入正确收敛盆地，但每步更新幅度过小，轨迹呈锯齿状振荡；其二，有效几何约束被外点、初值误差或分辨率层次稀释，导致每步残差下降量持续不足。Anderson 加速主要针对前者，通过历史外推加大有效步幅；MM 框架、鲁棒核与多分辨率策略则更多应对后者，通过提升目标函数质量或扩大可收敛区域来改善收敛性。混淆两类问题容易将“减少迭代步数”与“提高配准成功率”等同，而实际上两者针对的是不同层次的失效模式。

3.3.2 Anderson 加速：AA-ICP

Anderson 加速（Anderson Acceleration，AA）最早由 [39] 在不动点迭代的语境里提出。其应用条件相对宽松：无需显式梯度信息，仅要求能够计算映射 $G(u)$。具体而言，算法记录最近 m 步的残差历史，求解一个带约束 $\mathbf{1}^\top \alpha = 1$ 的小规模最小二乘，将多个历史方向线性组合为一次外推步。将其引入 ICP 时，对应搜索与位姿估计模块无需修改，仅需将 Picard 更新 $\boldsymbol{\xi}^{(k+1)} \leftarrow G(\boldsymbol{\xi}^{(k)})$ 替换为包含历史信息的组合更新，实现代价较低 [17]。

[17] 的实验把关键细节交代得比较清楚：在 TUM RGB-D Benchmark 的 Freiburg1 子集里，他们只用深度点云做 scan matching，把“帧间匹配”改成“隔 5 帧匹配”来模拟关键帧，总共处理了 2738 对扫描；在 Stanford Bunny（bun000 与 bun045）上，每帧约 4 万点，随机施加旋转/平移扰动做了 1000 次测试。参数上，收敛阈值取 $\varepsilon = 0.001$，最大迭代数 100，并限制外推系数（$|\alpha_j| \le 10$ 且 $\alpha_0 > 0$）来降低“外推把自己拽出盆地”的概率。对应到结果，论文报告中位数加速约 35%（平均约 30%），并且在约 97% 的 case 里最终误差更小（中位数改善约 0.3%）[17]。

值得注意的是，TUM RGB-D benchmark 的传感器配置与数据规模对结果的适用范围有一定约束：原始序列共 39 段，Kinect 采集 640×480 分辨率的 RGB-D 数据（30 Hz），真值轨迹由 8 台高速运动捕获相机以 100 Hz 频率提供；序列涵盖手持与 Pioneer 3 机器人载体 [40]。因此，AA-ICP 在该数据集上的加速效果主要反映“室内、连续帧、初值接近真值”的配准场景。

设历史残差矩阵 $F_k = [f_{k-m+1}, \dots, f_k]$，$f_j = G(\boldsymbol{\xi}^{(j)}) - \boldsymbol{\xi}^{(j)}$，AA 求解

$$\min_{\boldsymbol{\alpha}} \|F_k \boldsymbol{\alpha}\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^\top \boldsymbol{\alpha} = 1 \tag{22}$$

然后以加权组合外推新位姿：

$$\boldsymbol{\xi}^{(k+1)} = \sum_{j=0}^{m} \alpha_j^{(k)} \, G\Big(\boldsymbol{\xi}^{(k-j)}\Big) \tag{23}$$

[17] 的实现以“外层封装”为主要思路：ICP 内核的对应搜索、位姿求解与残差计算保持不变，仅在位姿更新环节插入历史外推步。Anderson 加速的额外开销主要体现在两方面：维护长度为 m 的历史残差矩阵，以及每轮求解规模随 m 线性增长的约束最小二乘。为保证数值稳定性，作者引入两类回退机制：当外推导致目标值恶化时回退至标准 Picard 步；当检测到收敛盆地发生切换时重置历史窗口，以防止历史方向失去几何意义而导致外推偏离。

从几何角度理解，若最近若干次迭代的更新方向高度相关，Picard 迭代实际上在反复执行幅度偏小、方向近似的修正步。Anderson 加速通过求解系数组合，使历史残差在当前点附近尽量相互抵消，从而构造出类似 secant 步的外推方向。该方法的主要适用场景是迭代已进入正确收敛盆地、但因步长不足而收敛缓慢的情形；若历史方向本身受到假极小值污染，外推步将放大偏差，这正是论文中设置回退与重置机制的原因。

稳定性增强：AA 在外点占优或错误盆地附近进行外推时，更新方向可能被“假极小值”牵引而变差。[17] 讨论了若干稳定化技巧，例如在外推导致目标值恶化时回退到标准 Picard 步、在收敛盆地发生切换时重置历史窗口等。这类“外推失败时退回”的策略在工程上很常见，能够在保留加速收益的同时降低外推带来的风险。

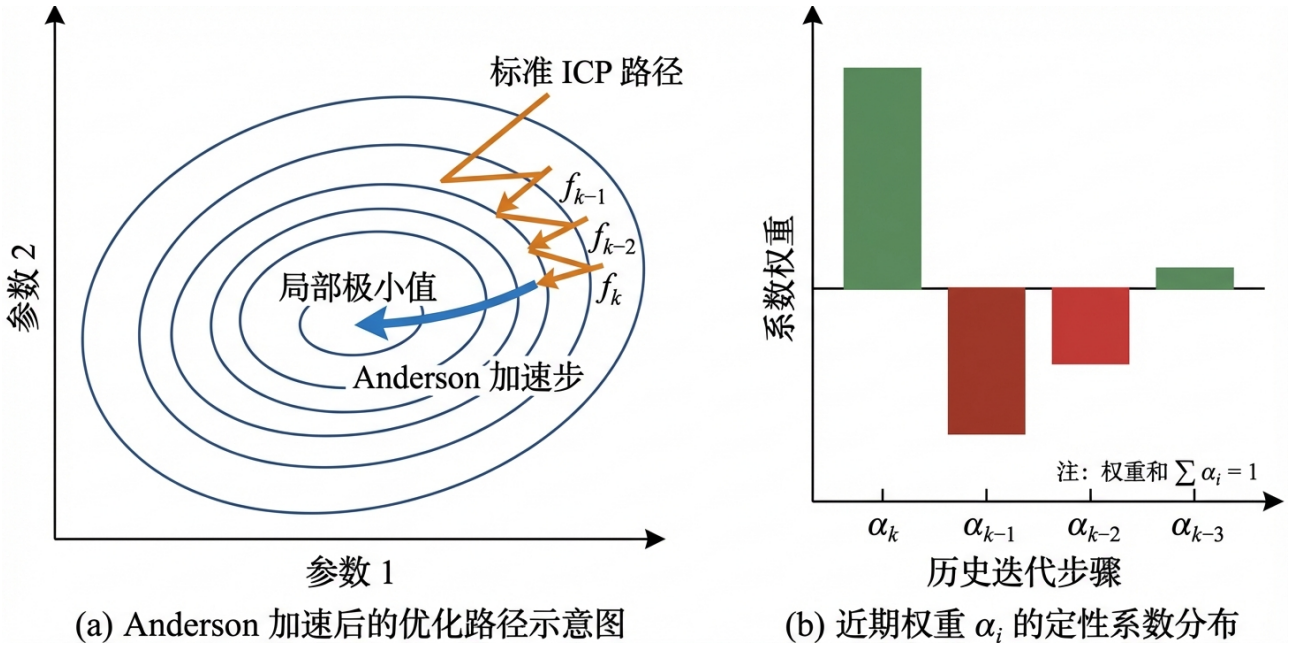


图 15: Anderson 加速在 ICP 迭代中的机制示意。面板（a）给出位姿空间中的迭代轨迹：标准 ICP 往往呈锯齿状，而 AA 通过线性组合最近若干步的更新方向生成外推步，使轨迹更平滑、步数更少。面板（b）示意历史窗口中各步权重系数 α_j 的相对贡献。

它的局限也很明确：AA 几乎不改变 ICP 的收敛盆地，更多是在既有盆地里“走得更像样”。如果初值本来就在错盆地，或者对应关系在几轮之间频繁跳变，历史残差就不再代表同一个局部几何结构，外推反而容易失真。换句话说，AA 的前提不是“任何 ICP 都能套”，而是局部迭代已经有相当强的一致性。

3.3.3 Majorization-Minimization 框架与 FRICP

[33] 将 ICP 从 AA 视角进一步推进：证明经典 P2P ICP 等价于一个 Majorization-Minimization（MM）算法，并基于这一视角同时实现加速与鲁棒化。

MM 视角：MM 算法通过最小化目标函数的代理（surrogate）上界来迭代逼近最优解。P2P ICP 最小化

$$\mathcal{E}(R,t) = \sum_i \|Rp_i + t - \hat{q}_i\|^2 \tag{24}$$

在固定对应 $\hat{q}_i$ 时，这是一个关于 (R,t) 的严格凸二次型（Σ 的代理），通过 SVD 一步求解。[33] 证明从一步到下一步这个更新就是 MM 步，从而可以直接应用 Anderson 加速于位姿的李代数参数化 $\boldsymbol{\xi} \in se(3)$，避免了欧拉角表示的奇异性问题。

将 ICP 纳入 MM 框架后，“固定对应”这一操作获得了严格的理论支撑：它等价于在当前迭代点处构造目标函数的代理上界，并对该代理上界进行精确最小化。在此视角下，MM 框架保证每轮迭代均对应目标函数的单调下降；Anderson 加速则进一步利用历史步的信息构造外推方向。两者在作用层次上相互正交：MM 框架关注每步更新的可靠性，Anderson 加速关注步长的有效性，组合使用时各自针对不同层次的收敛问题。

Welsch 函数鲁棒化：为处理外点，[33] 将 L_2 损失替换为 Welsch 函数

$$\psi_\nu(x) = 1 - \exp\left(-\frac{x^2}{2\nu^2}\right) \tag{25}$$

其中 ν 为尺度参数。Welsch 函数的性质：当 $x \ll \nu$ 时近似 L_2（内点区域）；当 $x \gg \nu$ 时趋于 1（外点区域的饱和损失），效果等同于给予外点接近零的权重。在 MM 框架下，Welsch 函数对应的迭代加权方案为

$$\omega_i^{(k)} = \exp\left(-\frac{\|d_i^{(k)}\|^2}{2\nu^2}\right), \quad d_i^{(k)} = Rp_i + t - \hat{q}_i \tag{26}$$

权重自适应地降低外点的贡献，同时对内点保留全权重，无需手动设置截断阈值。整个框架（MM + Anderson 加速 + Welsch 权重）即 FRICP（Fast Robust ICP），其设计目标是在保持鲁棒性的同时尽量把每步更新维持为“加权最小二乘 + 高效位姿更新”的形式，从而在工程实现上更易获得较高效率 [33]。

这这也是 FRICP 与 Sparse ICP 的关键分野。Sparse ICP 将“外点应当稀疏出现”直接编码进目标函数，鲁棒性较强，但求解器随之变重；FRICP 则有意选择 Welsch 等与 MM/加权最小二乘结构相容的鲁棒核，不追求最激进的稀疏建模，而是在保持鲁棒性的同时维护每轮更新的计算效率。其设计目标在于使鲁棒化不对迭代链条引入过多额外计算开销。

FRICP 的“快”并不是靠把停止条件放松出来的。[33] 明确写了对比设置：ICP、ICP-l 与 AA-ICP 统一使用同一套终止准则（最大迭代数 1000，或相邻两次变换差满足 $\|\Delta T\|_F < 10^{-5}$，ΔT 为两次迭代变换之差）；Sparse ICP 在 RGB-D SLAM 数据上取 $p = 0.8$、其余实验取 $p = 0.4$（Sparse ICP-l 全部取 $p = 0.4$）。在这套规则下，他们给出的时间单位是秒，误差用平均/中位 RMSE（表头注明 $\times 10^{-3}$）。

更关键的是数字本身。以 RGB-D SLAM 那组表为例（8 个序列），标准 ICP 的单对点云耗时大致在 0.23–0.93 s；AA-ICP 往往能压到 0.16–0.59 s；Fast ICP（非鲁棒版本）进一步到 0.14–0.43 s，并且这三者的 RMSE 基本维持在同一量级（例如 `fr1/xyz` 的 RMSE 都是 2.1/0.89 左右）[33]。鲁棒这边差别就更明显：同一张表里，Ours (Robust ICP) 在 `fr1/xyz` 上用 0.60 s 把 RMSE 做到 0.5/0.43，而 Sparse ICP 则是 11.2 s（RMSE 1.6/0.86）。在“部分重叠”的合成测试里，Bimba 一组更夸张：Sparse ICP 约 37.90 s，而 Ours (Robust ICP) 约 0.96 s、RMSE 0.87/0.67；标准 ICP 虽然更快（0.33 s），RMSE 却停在 68/60 这个量级 [33]。这几组数字基本把 FRICP 的位置讲透了：它愿意多花一点计算，把优化轨迹从“被外点拽着走”拉回到“内点在主导”。

Sparse ICP 对比：[15] 把外点“稀疏化”进目标函数：用 ℓ_p（$p < 1$）去惩罚残差，并用 ADMM 做变量分裂求解。它确实能把大残差压下去，而且论文里给的数字非常有说服力：在 Owl 的虚拟扫描实验中，初始误差 $e = 4.0 \times 10^{-1}$，把 p 往小推到 0.4 后能把误差压到 4.8×10^{-4}；即便用 $p = 1$，也能到 1.6×10^{-2} 的量级（另外，剪枝阈值 d_{th} 用 bbox 对角线的百分比来设，$d_{th} = 10\%$ 时 $p = 2$ 的误差约 2.9×10^{-2}）[15]。代价同样是“写在算法结构里”的：每轮不再是一个小的刚体闭式解，而要维护一套近端/乘子更新；因此在大规模点云上，时间往往被求解器吃掉，运行效率更依赖实现细节与超参。[33] 的 FRICP 用 Welsch 核得到类似的“抑制大残差”效果，但在 MM 框架下每步仍保持为加权最小二乘，更容易做出一个速度-鲁棒性都不难看的折中。

FRICP 自己也不是没有代价。它依赖鲁棒核尺度和退火/更新策略把“哪些点该被快速降权”处理得足够稳；若场景里内点本身就很少，或者对应几乎全被错误初值带偏，Welsch 权重也只能在局部框架内做修补，不能替代真正的全局初始化。

3.3.4 速度预测初始化：VICP

在连续扫描序列中（如 LiDAR 里程计），相邻帧之间的位姿变化可以用前帧的运动速度外推预测。VICP（Velocity ICP）沿这一思路显式估计传感器速度，并用速度更新来补偿扫描畸变与累积误差 [41]。若上一帧估计到的角速度和线速度为 $(\boldsymbol{\omega}_k, \boldsymbol{v}_k)$，则下一帧 $k+1$ 的初始位姿估计可写为

$$T_0^{(k+1)} = \exp(\hat{\boldsymbol{\omega}}_k \Delta t) \cdot T^{(k)} \cdot \exp(\hat{\boldsymbol{v}}_k \Delta t) \tag{27}$$

其中 Δt 为帧间时间间隔，$\hat{(\cdot)}$ 为从 $\mathbb{R}^3$ 到 $so(3)$ 或 $se(3)$ 的 hat 映射。速度预测的意义在于：把“完全未知的初值”替换为“由短时运动连续性外推得到的初值”，从而更频繁地把 ICP 送入正确的收敛盆地，减少无效迭代与发散风险。

VICP 所针对的问题不同于 AA 与 FRICP：后两者以“初始位姿已接近收敛区域”为前提，旨在减少局部阶段的冗余迭代；VICP 则工作在更早一步，利用运动连续性将初始估计拉近正确收敛盆地，从而为局部优化提供有效起点。两类方法在作用层次上互补：运动先验缩小初始误差，局部加速提升盆地内的收敛效率，通常在连续帧里程计系统中联合部署。

适用场景：VICP 更适合帧率较高、运动相对连续的序列配准（如里程计/建图前端）。当出现急转弯、加减速或点云畸变明显时，纯速度外推可能产生较大初值偏差，通常需要与 IMU 预积分或其他先验联合使用。典型 SLAM 系统如 FAST-LIO2 [42] 便以 IMU 约束提供更可靠的初值，本质上属于“用运动先验缩小 ICP 搜索范围”的思路。

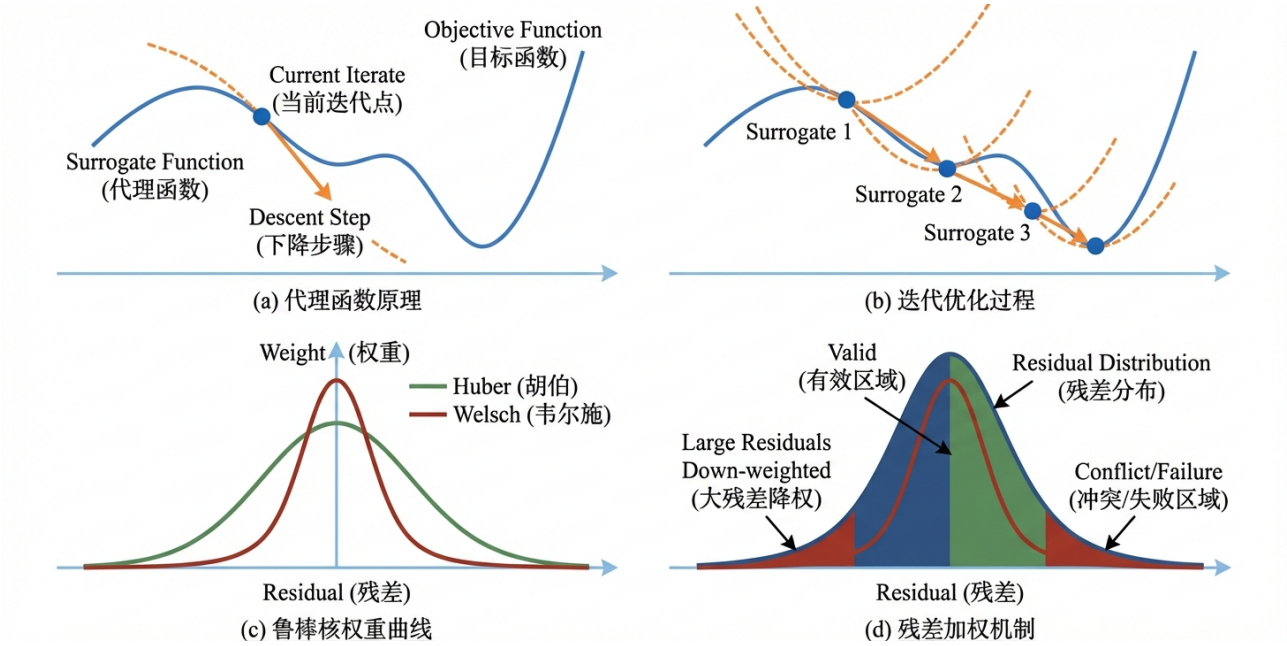


图 16: MM 代理函数与鲁棒核的机制示意。(a) MM 以“易优化的代理函数”上界原目标，在当前点处相切并逐步更新，使目标值单调下降；(b) 多步迭代中代理函数逐渐逼近原目标的局部极小；(c) 不同鲁棒核对应的权重函数 $w(r) = \rho'(r)/(2r)$：残差越大，权重越小，对应对优化的有效贡献被抑制；(d) 以残差分布为背景可视化权重对“内点区/外点区”的区分作用（示意）。

因此它的局限几乎都来自运动模型本身：一旦平台发生急剧机动、打滑、碰撞、或者时间同步与畸变补偿做得不好，速度先验就可能系统性地把初值推错。对于低帧率、弱连续性的配准任务，这类方法的收益也会明显下降。

VICP 的论文里给了一个很硬的对照：他们用 Hokuyo URG-04LX（扫描周期 100 ms/scan）在 7.2 m × 7.8 m 的室内环境跑了 4 组轨迹，其中两组是更快的运动（平均速度约 2.7 m/s，另外两组约 1.2 m/s）。用累计漂移误差对比，标准 ICP 在 Experiment 1/2 的旋转漂移分别达到 58.14° 和 79.98°、平移漂移分别是 2191 mm 和 2014 mm；VICP 则降到 7.28°/17.06° 与 177 mm/65 mm。即便在相对温和的 Experiment 3/4，VICP 也把旋转漂移从 16.80°/54.59° 降到 6.88°/3.28°（平移漂移 1490/2942 mm 降到 408/210 mm）[41]。这组数字基本说明：当扫描本身被运动拉“歪”了，单纯靠更快的局部求解器去加速 ICP 意义不大，先把畸变补回去，才谈得上收敛速度。

FAST-LIO2 属于“IMU 把初值兜住”的路线：它在 19 个公开序列的基准对比里报告自己在 17 个序列上精度最好，并把里程计与建图频率做到最高可达 100 Hz；论文还专门点名了两类极端设置：杂乱室内、旋转速度可到 1000 deg/s，以及最高 7 m/s 的高速运动 [42]。放在这里的意义很简单：速度/运动先验首先是为了把起点拉进“能收敛的区域”，其次才是为了少迭代几步。

连续时间 ICP：更进一步，连续时间运动模型（Continuous-Time ICP，CT-ICP）以时间连续轨迹来建模扫描内运动，并在配准过程中对扫描畸变进行补偿，是速度预测思想的时间连续延伸。[43] 将其用于实时 LiDAR 里程计，并在 KITTI odometry leaderboard 上报告平均相对平移误差（RTE）为 0.59%；同时给出单线程 CPU 上平均 60 ms/scan 的运行时间。

另外两条“实现层面的数字”也很有参考价值：为了保证前端节奏，CT-ICP 在实时模式下通常把单帧优化迭代上限压到 5 次；闭环部分则单独计时（高度图匹配平均约 1.1 s、位姿图优化约 1.2 s），这样就不会把后端的不确定性拖到前端里程计的时间预算里 [43]。

3.3.5 多分辨率策略：扩大收敛盆地

核心思想：” 先粗后精”。在分辨率较低（点云大幅降采样）的层级上 ICP 的目标函数更平滑——局部极小值被模糊掉，收敛盆地更宽；一旦在粗层找到近似正确的位姿，在精层恢复细节精度。标准三层金字塔设计为

$$T^* = \mathrm{ICP}_{r_0}(\mathrm{ICP}_{r_1}(\mathrm{ICP}_{r_2}(T_0, \mathcal{P}_{r_2}, \mathcal{Q}_{r_2}), \mathcal{P}_{r_1}, \mathcal{Q}_{r_1}), \mathcal{P}_{r_0}, \mathcal{Q}_{r_0}) \tag{28}$$

其中 $r_0 < r_1 < r_2$ 为点云分辨率（r_2 最粗，r_0 最细）。每层传入下层的是上层收敛的位姿估计，不是随机初始化。[25] 在三维 NDT 框架中展示，多分辨率策略能够显著提高对较大初始误差的容忍，从而更稳健地把优化推进到可精修的盆地内。

如果要把“容忍更大的初值”说得更具体一点，[25] 在其 3D-NDT 的评估里给了一个相对明确的量级：引入多分辨率离散化与三线性插值后，算法在初始平移误差到 0.5 m、初始旋转误差到 0.2 rad 的设置下仍能更稳定地收敛；而与 ICP 的对比实验也显示，3D-NDT 在差初值场景下往往更不容易被“卡死”。这里并不是说多分辨率能替代全局初始化，而是它确实能把局部优化的可收敛区域往外推一圈。

层数与降采样比的设计：工程上常用少量层级（例如三层）即可取得明显收益：粗层负责抹平局部极小并快速进入正确盆地，细层负责恢复细节精度。体素质心降采样（第 4.2 节）常用作粗层的默认选择，以在显著降低点数的同时保留整体几何；法向空间降采样（NSS）可在精层优先保留几何变化丰富区域，从而提升精修阶段的有效约束。

多分辨率真正省下来的，也不只是“粗层点少，所以每轮更便宜”。更重要的是粗层先把高频几何细节压掉，使目标函数的局部起伏少很多，很多原本会把局部优化绊住的假极小先消失了；一旦位姿被推入正确的收敛盆地，再将细节逐层恢复以完成精修。也正因为如此，多分辨率常常同时提升成功率和总收敛效率，而不只是减少单轮算量。

与 AA 的正交叠加：多分辨率策略与 Anderson 加速在作用机理上基本正交——粗层主要负责把初始误差压入可收敛盆地，AA 则在盆地内减少迭代步数。工程上常见的做法是在粗层/中层完成“入盆地”，再在精层启用 AA 或鲁棒核来加快收敛；其代价主要来自多层表示与一次小规模最小二乘外推的维护开销 [13]，[17]。

多分辨率策略的局限在于，当粗层降采样过于激进时，可能同时抹去区分真解与局部极小值所需的几何细节，使目标函数在多个候选解附近呈现相似的代价景观，反而失去辨别能力。因此，多分辨率策略更适合作为局部优化的前置初始化机制，而非独立承担全局搜索的职责。

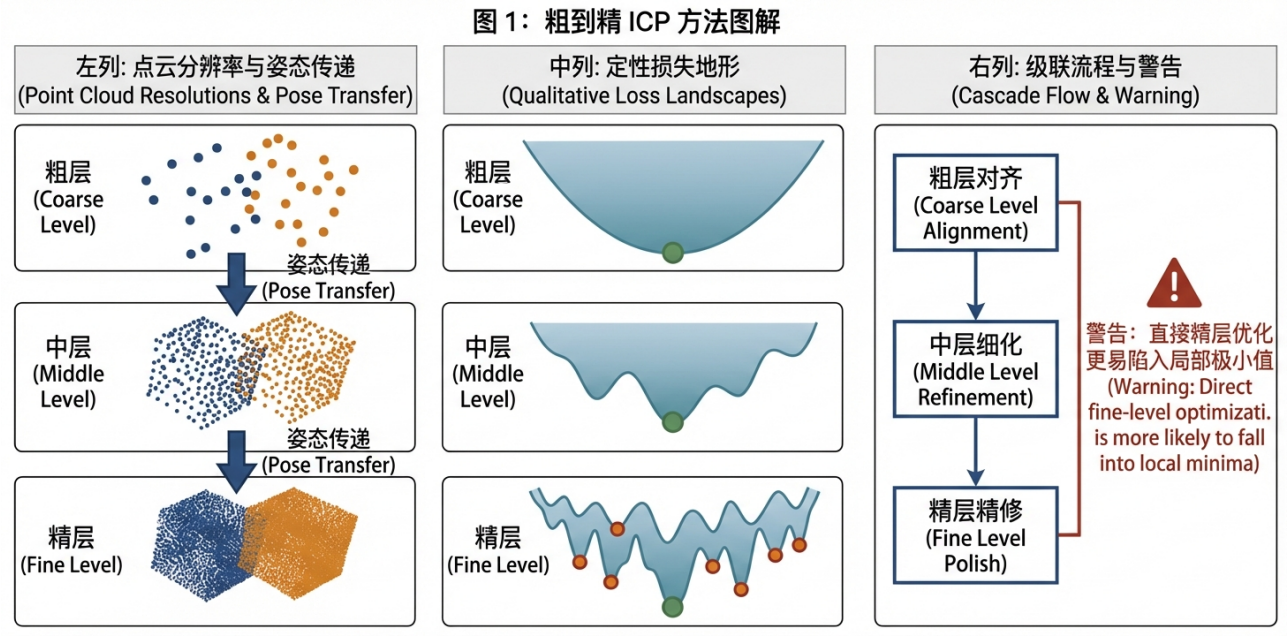


图 17: 多分辨率 ICP 的三层金字塔与“先粗后精”的作用机理示意。左列：点云从粗到细的三层表示；中列：对应层级下的代价景观，粗层更平滑、局部极小被弱化，细层细节丰富但局部极小更多；右列：级联流程，粗层负责进入正确盆地，细层负责精修到高精度。

3.3.6 自适应终止准则

ICP 的终止准则直接影响精度与计算时间的平衡。标准 ICP 实现通常同时采用以下三类准则，任意一个触发即停止：

变换增量准则：

$$\|\Delta R\|_F < \epsilon_R \quad \text{且} \quad \|\Delta t\|_2 < \epsilon_t \tag{29}$$

当相邻两次迭代的位姿变化量同时低于阈值时停止。阈值应结合点云尺度与噪声水平设置，以避免过早停止或无谓迭代。

残差相对改善准则：

$$\frac{\left|\mathcal{E}^{(k)} - \mathcal{E}^{(k-1)}\right|}{\mathcal{E}^{(k)}} < \delta \tag{30}$$

残差相对下降量低于阈值时停止。这一准则能检测” 平台期”：残差已不再改善但仍未触发变换增量准则的情况。

最大迭代次数：$k > k_{\max}$。硬截断保证最坏情形下运行时间有界，是实时系统的必要安全阀。

自适应 $k_{\max}$：进一步地，可根据残差改善趋势动态调整 $k_{\max}$：当连续多步改善极小或呈震荡时更早终止；当改善趋势稳定且仍显著时允许额外迭代以换取更低残差。阈值应结合点云尺度、噪声水平与实时性预算设置，避免把“参数默认值”误当作跨场景通用规律。

终止准则被纳入收敛加速的讨论范畴，原因在于算法效率的比较高度依赖于停止条件的一致性。若终止判据不统一，加速效果可能仅仅来自更宽松的停止阈值，而非更高效的迭代策略。AA-ICP 与 FRICP 均在实验中明确给出统一的终止条件设置，从而保证加速比数据反映真实的迭代效率差异，具有可比性。

当然，终止准则本身也有局限。变换增量阈值在退化场景里可能过于乐观，因为“几乎不动了”未必代表“已经对齐了”；残差改善阈值又可能在重尾噪声下过早进入平台期。真正稳妥的做法通常不是依赖某一条单一准则，而是把变换、残差和最大迭代数一起看，并让阈值跟点云尺度和实时预算绑定。

终止准则的精确报告是评估收敛加速效果的必要条件，以防止将提前停止与迭代效率提升相混淆。AA-ICP 明确给出收敛阈值 $\varepsilon = 0.001$、最大迭代数 100 及外推系数限制 $\alpha_l = 10$ [17]。FRICP 的对比实验中，ICP / ICP-l / AA-ICP 的停止条件统一设定为最大迭代数 1000 或 $\|\Delta T\|_F < 10^{-5}$ [33]，从而保证报告的加速比反映真实的迭代效率差异，而非停止时机的不同。

3.3.7 收敛加速方法综合对比

表 9: 第 3.3 节收敛加速方法对比（定性）：加速来源、初值容忍、外点稳定性、额外开销与典型使用方式。

方法	加速来源	初值容忍	外点稳定性	额外开销	更适合的使用方式
标准 ICP	—	低	低	低	有可靠初值、低噪声场景的局部精修
AA-ICP	历史外推	低	中	低（历史缓存 + 小规模最小二乘）	在同一盆地内的快速收敛 [17]
FRICP	鲁棒 MM/GNC + 外推	中	高	中（权重更新 + 外推）	含外点/噪声的稳定精修 [33]
Sparse ICP	稀疏惩罚	低	高	高（ADMM/近端迭代）	离线或小规模的强鲁棒配准 [15]
VICP [41]	运动先验提供初值	中	中	低	连续帧里程计/建图前端（与 IMU/里程计结合）
多分辨率	粗层入盆地 + 细层精修	高	中	中（多层表示与多次求解）	大初始误差或局部极小丰富的场景 [13]
组合策略	先入盆地再加速	高	高	中—高	全局初始化/多分辨率/鲁棒核/AA 的按需组合

3.3.8 收敛速度的根本限制

值得指出，上述方法主要加速的是“已在盆地内”的局部收敛速率，而**不直接扩大收敛盆地本身**。Anderson 加速和 MM 框架仍属于局部策略：若初始位姿处于错误盆地，外推反而可能放大偏离。多分辨率策略通过粗层平滑目标函数景观，能在一定程度上提升对初始误差的容忍，但仍不构成全局最优保证。真正意义上的全局初

表 10: 第 3.3 节引用数据汇总（覆盖本节全部引用）。表中数字用于把“加速/更鲁棒”落到可复现实验设置上。

引用	任务/数据	指标	关键数字（设置/结果）
[13]	合成网格（约 10^5 点）	采样与时间	每轮采样 2000 点（约 1%）；550 MHz PIII Xeon；对齐耗时“几十毫秒”（初值良好时）
[39]	不动点迭代（方法本身）	计算规模	历史长度为 m；每轮解一个 m+1 维系数的约束最小二乘（额外开销主要在小规模线性代数）
[17]	TUM RGB-D（Freiburg1）+ Bunny	加速比/设置	2738 对扫描；隔 5 帧匹配；Bunny 每帧约 4 万点、1000 次扰动；$\varepsilon = 0.001$、$\alpha_l = 10$、max iters=100；中位数加速约 35%、误差中位数改善约 0.3%
[40]	TUM RGB-D Benchmark	数据规模/频率	39 个序列；Kinect 640×480@30 Hz；动捕 8 相机给 100 Hz 真值轨迹；含手持与 Pioneer 3 载体
[33]	RGB-D SLAM + 部分重叠模型	时间/RMSE/停止条件	停止条件统一：max iters=1000 或 $\|\Delta \mathrm{T}\|_F < 10^{-5}$；例如 `fr1/xyz`：ICP 0.23 s、AA-ICP 0.16 s、Robust ICP 0.60 s，RMSE 分别约 2.1/0.89 与 0.5/0.43（表头注明 $\times 10^{-3}$）
[15]	Sparse ICP（Owl 虚拟扫描）	误差	初始 $e = 4.0\times 10^{-1}$；$p = 0.4$ 后 $e = 4.8\times 10^{-4}$；$p = 1$ 时 $e = 1.6\times 10^{-2}$；d_{th} 以 bbox 对角线百分比计（如 10%）
[41]	2D 激光里程计	漂移误差	URG-04LX：100 ms/scan；Exp1：ICP 58.14°/2191 mm vs VICP 7.28°/177 mm；Exp2：79.98°/2014 mm vs 17.06°/65 mm；快运动均速约 2.7 m/s
[42]	LiDAR-IMU 里程计	频率/极端动态	19 个序列基准；17 个序列精度最好；最高 100 Hz；旋转可到 1000 deg/s；最高 7 m/s 运动
[43]	LiDAR-only SLAM	RTE/时间预算	KITTI leaderboard：RTE 0.59%；60 ms/scan（单线程）；实时模式迭代上限常设 5；闭环匹配约 1.1 s、图优化约 1.2 s
[25]	3D-NDT 多分辨率注册	初值容忍	报告多分辨率 + 三线性插值后，对初始平移 0.5 m、初始旋转 0.2 rad 的鲁棒性显著提升

始化与可认证求解见第 3.6 节 （如 Go-ICP、TEASER++ 等），其作用是先将误差压入可收敛区域，再由多分辨率加局部加速在精层快速收敛。第 4 章将进一步讨论软件层面的优化（如批处理近邻查询、并行化线性代数）如何降低每次迭代的绝对时间开销。

3.4 变换估计方法 (Transformation Estimation)

ICP 两步交替框架的第二步是：给定固定的对应集 $\{(p_i, q_{j(i)})\}_{i=1}^{n}$，求解最优刚体变换 (R^*, t^*) 使加权距离平方和最小。这一子问题本身数学上是闭式可解的（封闭形式解），但不同参数化方案（旋转矩阵 SVD、单位四元数、李代数 $se(3)$、对偶四元数）在数值稳定性、与更新规则的兼容性和不确定性传播能力上有显著差异。本节系统梳理五类方法，以及从统计学角度统一它们的概率框架。

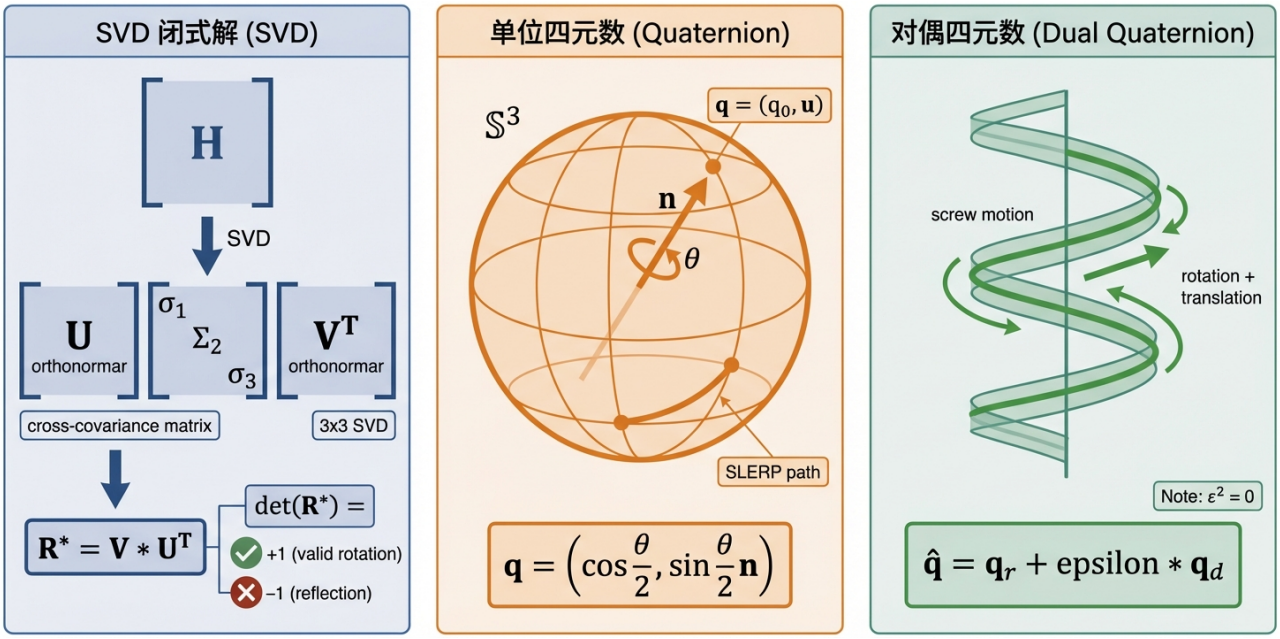


图 18: 三种旋转参数化方案的几何示意。左：SVD 闭式解将交叉协方差矩阵 H 分解为 $H = U\Sigma V^\top$，最优旋转 $R^* = VU^\top$，行列式检查确保真旋转。中：单位四元数将旋转编码为三维单位超球面 S^3 上的点，球面测地距离对应旋转角，SLERP 在球面大圆弧上插值。右：对偶四元数 $\hat{q} = q_r + \varepsilon q_d$ 将旋转（实部）与平移（对偶部）统一为单个代数结构，表示螺旋运动。

3.4.1 问题形式化

ICP 变换估计步的标准形式为：给定 n 个加权对应点对 $\{(p_i, q_i, \omega_i)\}$，求解

$$(R^*, t^*) = \arg\min_{R\in SO(3),\, t\in\mathbb{R}^3} \sum_{i=1}^{n} \omega_i \|Rp_i + t - q_i\|_2^2 \tag{31}$$

其中 $\omega_i \geq 0$ 为第 i 个对应点对的权重（在鲁棒 ICP 中由 M-估计器动态确定，在 P2P ICP 中均为 1）。注意到目标函数关于 t 是强凸的——固定 R 时最优平移为 $t^* = \bar{q} - R\bar{p}$，其中 $\bar{p} = \frac{\sum \omega_i p_i}{\sum \omega_i}$，$\bar{q} = \frac{\sum \omega_i q_i}{\sum \omega_i}$ 为加权质心。去均值后问题化简为纯旋转估计，是各方法的核心区别所在。

3.4.2 SVD 闭式解（Kabsch 算法）

去均值后构造加权交叉协方差矩阵：

$$H = \sum_{i=1}^{n} \omega_i (p_i - \bar{p})(q_i - \bar{q})^\top \in \mathbb{R}^{3\times 3} \tag{32}$$

对 H 做奇异值分解 $H = U\Sigma V^\top$，最优旋转与平移为

$$R^* = VU^\top, \qquad t^* = \bar{q} - R^*\bar{p} \tag{33}$$

奇异性处理：当 $\det(VU^\top) = -1$ 时，SVD 给出的是反射（reflection）而非旋转，需将 V 的最后一列取反：

$$R^* = V\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & \det(VU^\top)\end{pmatrix}U^\top \tag{34}$$

[11] 于 1987 年给出了完整的 SVD 推导和奇异性处理，证明了该解法在 L_2 意义下的全局最优性，以及当 H 的最小奇异值为零（点集严格共面）时退化的处理方法。SVD 更新的一个重要工程优势是：位姿求解本身计算代价很低、数值稳定性好，因而在多数实现中并非瓶颈；ICP 的主要耗时通常来自近邻查询与残差/雅可比的批量评估。

这件事在他们给出的计时对比里也很直观：[11] 在 VAX 11/780 上,对比了 SVD 法、四元数法与迭代法的端到端计算时间(包含构造 H、分解与求解)。以点对数 $N = 7/11/16/20/30$ 为例,SVD 法约为 37.0/40.0/39.2/40.4/44.2 ms，四元数法约为 26.6/32.8/39.9/45.2/48.3 ms；迭代法则约为 94.2/110.8/120.5/135.0/111.0 ms，对应迭代次数分别为 5/7/10/6/6 次（论文表格直接报数）[11]。因此，对“固定对应、解一次最小二乘”这一步来说，真正值得优化的通常不是 3×3 SVD 或 4×4 特征分解，而是更前面的对应建立与更后面的鲁棒权重更新。

SVD 闭式解的局限性与其假设前提直接相关：该方法默认输入对应集可靠且权重合理，一旦对应集中包含大量外点或几何结构出现退化，闭式解将同样给出无意义的结果。该方法解决的是给定对应集下的刚体最小二乘子问题，对应集的质量与问题的可观测性由上游模块负责保障。

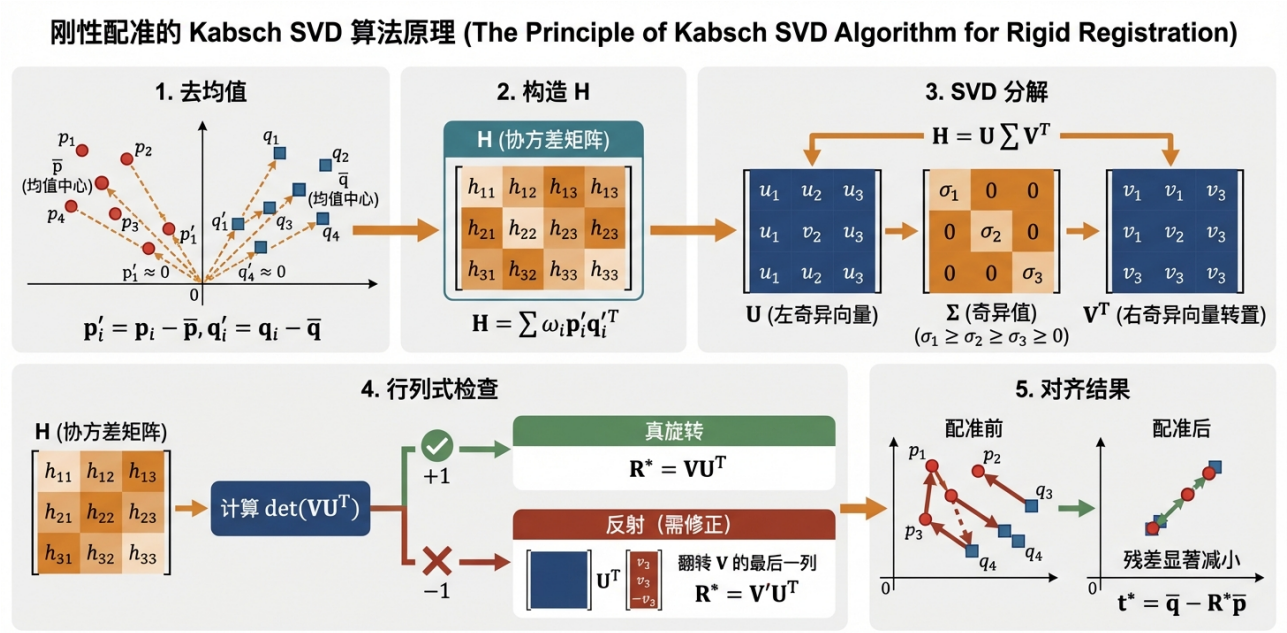


图 19: Kabsch（SVD）闭式解的五步骤图解（示意）：去均值 → 构造交叉协方差 H → SVD 分解 → 行列式检查（防反射）→ 得到 R^*, t^* 并将源点对齐到目标点。

Horn 同年的等价结果：[12] 基于四元数方法独立得出等价结论，其推导路径不同但最终给出的旋转矩阵与 Arun 的 SVD 方法完全一致。两种方法的等价性说明 SVD 和四元数特征分解在数学上描述的是同一个问题结构。

3.4.3 单位四元数法

[12] 将旋转 R 参数化为单位四元数 $\mathbf{q} = (q_0, q_1, q_2, q_3)^\top$，$\|\mathbf{q}\| = 1$，构造对称 4×4 矩阵 N（由交叉协方差矩阵 H 的元素组成）：

$$N = \begin{pmatrix} H_{xx}+H_{yy}+H_{zz} & H_{yz}-H_{zy} & H_{zx}-H_{xz} & H_{xy}-H_{yx} \\ H_{yz}-H_{zy} & H_{xx}-H_{yy}-H_{zz} & H_{xy}+H_{yx} & H_{zx}+H_{xz} \\ H_{zx}-H_{xz} & H_{xy}+H_{yx} & -H_{xx}+H_{yy}-H_{zz} & H_{yz}+H_{zy} \\ H_{xy}-H_{yx} & H_{zx}+H_{xz} & H_{yz}+H_{zy} & -H_{xx}-H_{yy}+H_{zz} \end{pmatrix} \tag{35}$$

最优旋转四元数 $\mathbf{q}^*$ 是 N 的**最大特征值**对应的特征向量。四元数表示的核心优势：

1. **无奇异性**：欧拉角存在万向锁（gimbal lock），四元数无此问题。
2. **插值友好**：SLERP（Spherical Linear Interpolation）在单位球面 S^3 上做测地线插值，保持旋转的连续性，适合位姿滤波（卡尔曼滤波、粒子滤波）。
3. **计算量相当**：4×4 特征分解比 3×3 SVD 略重，实践中差异可忽略。

就可解条件而言，四元数法与 SVD 法并无本质差别：均需至少 3 对不共线的对应点方能约束完整的三维旋转；一旦点集退化为严格共线或共面，H 的秩会下降，旋转的某些自由度将变得不可观，[12] 与 [11] 均在推导中专门讨论了此类退化情形。

与 SVD 等价性的深层含义：两种方法的“分解部分”均作用于固定大小的矩阵（3×3 SVD 或 4×4 特征分解），代价基本可视为常数；但这并不意味着变换估计步与点数完全无关。构造质心与交叉协方差 H 仍需 $O(n)$ 的线性遍历，只是该步骤的常数极小，通常远小于一次 KD-tree 最近邻查询的开销。在多数工程实现中，变换估计步很少成为瓶颈，瓶颈更常出现在对应搜索、法向/协方差估计或鲁棒核的权重更新上。

四元数法的局限不在计算效率，而在其解决的问题与 SVD 一样，仍然只是固定对应下的闭式刚体估计。采用四元数参数化本身并不会提升对坏对应或退化结构的鲁棒性；其优势主要体现在表示层面：更适合插值、滤波与连续旋转处理。

3.4.4 李代数参数化与 $SE(3)$ 直接优化

在 FRICP [33] 和许多现代 SLAM 框架中,旋转以李代数 $se(3)$ 参数化代替旋转矩阵,主要优势是在 Anderson 加速和梯度下降中可以进行” 加法更新”:

$$\boldsymbol{\xi} = (\boldsymbol{\omega}, \boldsymbol{v}) \in \mathbb{R}^6, \quad T = \exp(\hat{\boldsymbol{\xi}}) \in SE(3) \tag{36}$$

其中 $\hat{\boldsymbol{\xi}}$ 为 $se(3)$ 矩阵，exp 为矩阵指数（可通过 Rodrigues 公式解析计算）。李代数参数化使得迭代更新为

$$\boldsymbol{\xi}^{(k+1)} = \boldsymbol{\xi}^{(k)} + \Delta\boldsymbol{\xi}, \quad T^{(k+1)} = \exp(\hat{\Delta\boldsymbol{\xi}}) \cdot T^{(k)} \tag{37}$$

这一线性化更新与 Anderson 加速天然兼容（可以在 $\mathbb{R}^6$ 空间直接做线性组合），而旋转矩阵本身不在线性空间中，直接做 $R_1 + R_2$ 会破坏正交性约束。FRICP 正是通过将 Anderson 加速应用于 $se(3)$ 参数化的 $\boldsymbol{\xi}$，避免了原始 AA-ICP 在欧拉角参数化下的万向锁奇异性 [33]。更关键的是，该参数化方案具有实验验证：[33] 将 ICP / ICP-l / AA-ICP 的终止条件统一设定为“最多 1000 次迭代，或两次迭代的变换差满足 $\|\Delta T\|_F < 10^{-5}$”，从而能在统一停止口径下直接比较不同更新规则的收敛表现。

但李代数参数化也不是“用了就更稳”。李代数参数化解决的是表示与更新规则的兼容性问题，而非可观测性问题；若 Hessian 本身病态，采用 $se(3)$ 参数化仅改善更新规则的数值结构，不会弥补缺失的几何约束。在大角度初值下，指数映射与局部线性化的有效范围仍需依赖更好的初值或多分辨率策略来保障。

3.4.5 对偶四元数法与 Sim(3) 扩展

对偶四元数（Dual Quaternion）以 $\hat{\mathbf{q}} = \mathbf{q}_r + \varepsilon\mathbf{q}_d$（$\varepsilon^2 = 0$）将旋转四元数 $\mathbf{q}_r$ 和平移四元数 $\mathbf{q}_d$ 合并。其代数运算：

$$\hat{\mathbf{q}}_1 \otimes \hat{\mathbf{q}}_2 = \mathbf{q}_{r1}\mathbf{q}_{r2} + \varepsilon(\mathbf{q}_{r1}\mathbf{q}_{d2} + \mathbf{q}_{d1}\mathbf{q}_{r2}) \tag{38}$$

天然处理旋转与平移的耦合，表示" 螺旋运动" (screw motion) ——绕轴旋转同时沿轴平移，是 $SE(3)$ 的 Plücker 坐标表示。

[44] 将对偶四元数框架扩展以引入各向同性缩放因子 s，将变换群从 $SE(3)$ 扩展到相似变换群 Sim(3)。从参数维度上看，$SE(3)$ 是 6 自由度，而 Sim(3) 变为 7 自由度（多了 1 个尺度）；从实现上看，对偶四元数用 8 维向量表示（实部四元数 4 维 + 对偶部 4 维），再配上单位范数等约束把自由度压回到“刚体/相似变换”该有的维数。这类表示在“尺度误差不可忽略”的场景里很有用：例如跨传感器标定、不同扫描分辨率的点云拼接、或三维重建存在尺度漂移时，刚体模型会把尺度误差硬塞到平移与旋转里，误差会以系统性偏置的形式扩散到整条轨迹。

在实验数据层面, [44] 明确提到在模拟 3D 曲线与真实点云上验证, 并点名使用 Princeton Shape Benchmark 与 Stanford 3D Scanning Repository（Bunny）作为真实点云来源；这两类数据集的选择各有其依据：Princeton Shape Benchmark 提供了几何多样性较高的形状族，适合暴露尺度估计误差；Stanford Bunny 则是点云配准领域最常用的公开基准之一，便于与现有方法横向对比 [44]。具体地，目标函数扩展为：

$$(R^*, s^*, t^*) = \arg\min_{R,s,t} \sum_i \omega_i \|sRp_i + t - q_i\|^2 \tag{39}$$

其中尺度 $s > 0$ 的最优解有闭式表达式 $s^* = \frac{\mathrm{tr}(\Sigma_W R^{*\top})}{\mathrm{tr}(W\Sigma_p)}$，可与 SVD 一步联立求解。Sim(3) 配准适用于跨传感器标定、多视角点云拼接（不同扫描头分辨率差异）和稀疏-稠密激光雷达融合场景，是医学影像配准（CT 与 MRI 分辨率差异）的标准工具。

它的局限是把问题从 6 自由度抬到 7 自由度之后，尺度会和旋转、平移产生更强耦合。若真实场景其实没有明显尺度误差，或者对应本身就不稳，额外引入尺度自由度反而可能把噪声吸收到 s 里，造成看似拟合更好、实则物理解释更差的结果。因此 Sim(3) 更适合确实存在尺度漂移或跨模态比例差异的场景，而不是刚体配准的默认替代品。

3.4.6 广义 ICP：概率框架的统一

广义 ICP（Generalized ICP，GICP）[26] 为每个点分配由局部邻域协方差矩阵确定的不确定性 Σ_p^i, Σ_q^j，将 P2P 目标改写为马氏距离形式：

$$\mathcal{E}_{\mathrm{GICP}} = \sum_i \mathbf{d}_i^\top \left(R\Sigma_p^i R^\top + \Sigma_q^{j(i)}\right)^{-1} \mathbf{d}_i, \quad \mathbf{d}_i = Rp_i + t - q_{j(i)} \tag{40}$$

协方差矩阵 Σ_p^i 通过点 p_i 的 k 个近邻点估计得到，其特征值结构反映局部几何：平面区域 Σ 的最小特征值接近零（法向方向不确定性小），边缘区域 Σ 接近均匀（各向同性）。在 [26] 的实现细节里，邻域规模给得很具体：用每个点的 20 个最近邻来做经验协方差的特征分解；而在对比设置中，标准 ICP 的迭代上限设为 250，而点到面与 GICP 设为 50 次迭代。真实数据部分还给出车载 Velodyne 的配对示例，描述为两帧扫描约 30 m 间隔、量测范围约 70–100 m 的室外场景 [26]。这些细节表明 GICP 的优势来源于将局部几何统计引入权重分布，使之更贴近真实噪声模型，而非依赖更复杂的后端求解器；代价是邻域统计估计与矩阵分解所引入的常数开销。

三种对应度量的统一：当 $\Sigma_p^i = \Sigma_q^i = I$ 时，GICP 退化为 P2P ICP；当 $\Sigma_p^i = 0$，Σ_q^j 为切平面投影矩阵（沿法向的单位矩阵，切向为零）时，退化为 P2Pl ICP；GICP 的完整协方差矩阵形式则在两者之间做自适应加权，由局部曲率自动决定 [26]。这一统一视角明确了 P2P 和 P2Pl 在统计学意义上的关系：P2Pl 假设目标点云是理想平面，P2P 假设各向同性噪声，而 GICP 使用真实的各向异性噪声估计，因此精度更高。

GICP 的最优化：内层最优化（固定对应，求 (R, t)）可用 Gauss-Newton 或 LM 法在 $se(3)$ 上迭代求解，因为协方差权重使该问题不再有 Kabsch 那样的线性闭式解。由于需要估计协方差并求解加权非线性问题，单次迭代代价往往高于 P2P；但在几何与噪声更符合其假设时也可能以更少迭代达到稳定解，因此总耗时与收敛质量依赖实现细节与场景分布。

GICP 的局限也恰恰埋在它最强的地方：局部协方差如果估得不准，统计模型就会把错误的几何假设一并放大。稀疏点云、邻域选择不合适、法向/曲率本身噪声较大时，协方差椭球未必比简单的 P2P/P2Pl 更可信；在此情形下，较高的建模与求解代价未必能换来更稳定的配准结果。

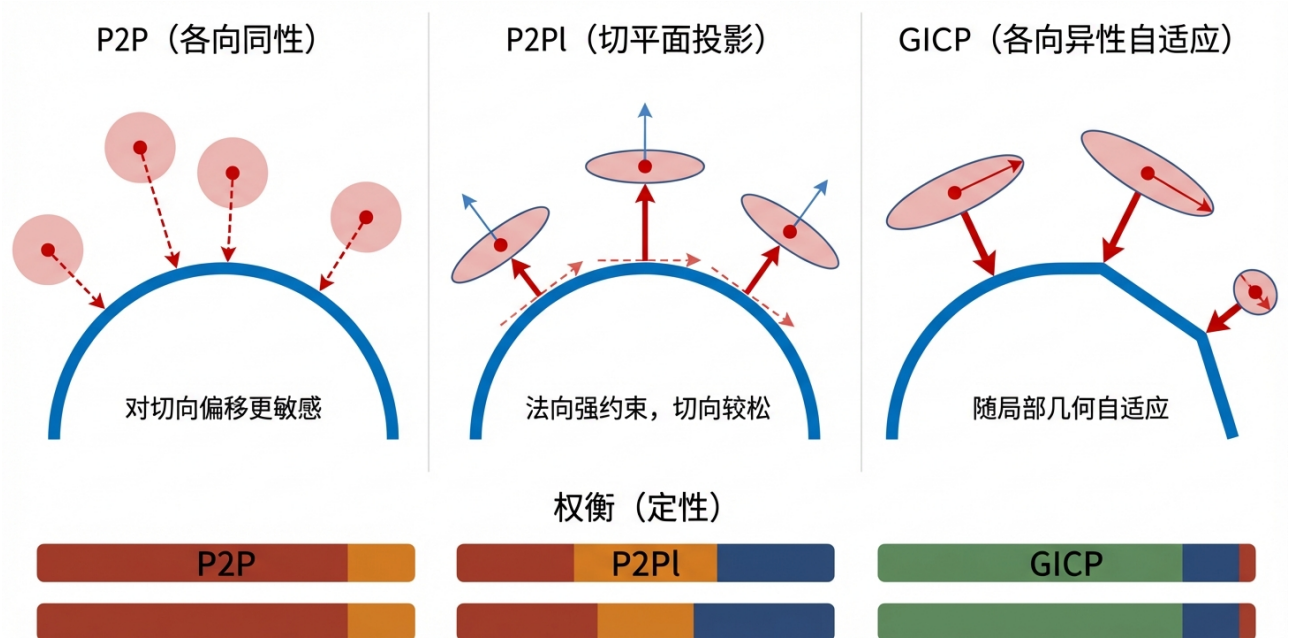


图 20: GICP 中三种距离度量的几何解释对比（示意）。左（P2P）：各向同性噪声假设对应球形误差，对曲面处的切向偏移更敏感。中（P2Pl）：切平面投影对应“法向强约束、切向更松”。右（GICP）：用局部协方差自适应权重，不同几何处（平面/边缘/角点）的误差椭球形状不同，从而在统计意义上折中 P2P 与 P2Pl。底部用定性条形对比强调：三者在误差与鲁棒性上的取舍随场景而变。

闭式协方差估计是另一条常用的不确定性建模路线：在收敛附近对 ICP 目标做小扰动线性化，并结合测量噪声模型，可推导位姿估计协方差的闭式表达。[45] 给出了经典推导，并用几组“欠约束/可观测性”很强的场景把误差口径对齐。例如在一个 10 m 边长的正方形环境里，设置真值位姿增量为 $x = (0.1\ \mathrm{m}, 0, 2°)$，论文报告定位误差的“真值标准差”约为 $(5.3\ \mathrm{mm}, 5.3\ \mathrm{mm}, 0.039°)$，闭式估计给出 $(5.4\ \mathrm{mm}, 5.4\ \mathrm{mm}, 0.042°)$；而在 scan-matching 口径下，真值约为 $(7.6\ \mathrm{mm}, 7.8\ \mathrm{mm}, 0.058°)$，闭式估计为 $(7.7\ \mathrm{mm}, 7.7\ \mathrm{mm}, 0.060°)$（论文表格直接对照）[45]。这类对比的意义在于：当线性化点合理、噪声模型不过分离谱时，闭式协方差并不是“拍脑袋的 Hessian 逆”，它能在毫米/百分之一度的量级上把统计量对齐。与 Stein ICP 的粒子后验相比，闭式协方差计算更轻量、易于在线集成，但它也更依赖噪声假设与收敛附近的线性化前提。

因此闭式协方差更适合回答“在当前单峰近似下，不确定性大概有多大”，并不擅长处理明显多模态、强非线性或数据关联本身不确定的情形。它很实用，但不能把它误当成完整后验。

3.4.7 Stein ICP：后验分布估计

Stein ICP [46] 将变换估计从点估计（MAP）扩展到完整后验分布估计 $p(R, t|\mathcal{P}, \mathcal{Q})$。采用 Stein 变分梯度下降（SVGD），维护一组粒子 $\{\boldsymbol{\xi}_k\}_{k=1}^{K}$（每个粒子代表一个候选变换），通过下式同时最大化每个粒子的似然并保持粒子间多样性：

$$\dot{\boldsymbol{\xi}}_k = \frac{1}{K}\sum_{j=1}^{K}\Big[k(\boldsymbol{\xi}_j, \boldsymbol{\xi}_k)\nabla_{\boldsymbol{\xi}_j}\log p(\boldsymbol{\xi}_j|\mathcal{P}, \mathcal{Q}) + \nabla_{\boldsymbol{\xi}_j}k(\boldsymbol{\xi}_j, \boldsymbol{\xi}_k)\Big] \tag{41}$$

其中 $k(\cdot,\cdot)$ 为 RBF 核。右侧第一项驱动粒子向高概率区域移动（类似梯度上升），第二项通过核梯度产生粒子间排斥力以保持多样性，避免所有粒子塌缩到同一模式。Stein ICP 的输出是一个粒子集而非单一变换，可直接用于后续的安全决策（如自动驾驶避障）和传感器融合（如将 ICP 输出的位姿不确定性输入 EKF）。

在对称场景（如走廊、圆柱形容器）中，ICP 的目标函数可能有多个等效极小值。Stein ICP 能识别并表示这种多模态分布，而点估计方法只返回其中一个，可能是错误的那个 [46]。

这篇论文把“粒子后验到底有没有用”用两组数字说得很实在。[46] 在 challenging 数据集上用 KL 散度和重叠系数（overlap, OVL）评价后验拟合质量，并在粒子数消融后取 $K = 100$ 作为默认设置；对照的 Bayesian ICP

需要 1000 个样本。定量对比表里，Stein ICP 的 KL（中位数）大致落在 0.6–5.7，OVL 在 0.7–0.9；Closed-form ICP 的 OVL 近乎为 0（几乎不重叠，说明“高斯 + 忽略数据关联不确定性”的假设在这些场景下非常乐观）[46]。效率方面，作者在 GPU 上对运行时做了组件分解：Stein ICP 的总耗时约为 Bayesian ICP 的 1/8–1/5，并报告“超过 5 倍”的整体加速。这些数值背后对应的是算法结构差异：SVGD 可以按粒子并行更新，而 MCMC 链式采样天然串行，硬件利用率拉不开就只能用样本数硬堆。

即便如此，Stein ICP 的局限仍然很重：它终究要维护一批粒子，计算和显存开销都远高于单解方法；核带宽、粒子数和初始化分布也会直接影响后验质量。对实时前端来说，它通常更像分析工具或高安全需求模块，而不是默认的每帧求解器。

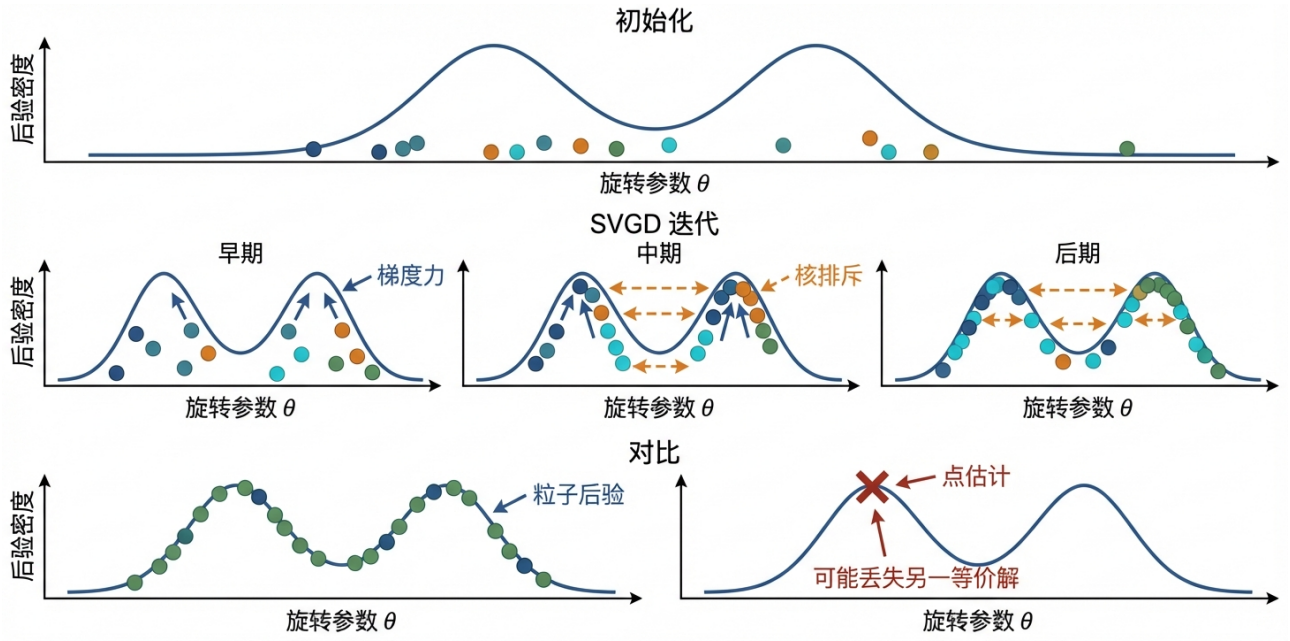


图 21: Stein ICP 以 Stein 变分梯度下降（SVGD）维护一组粒子来近似位姿后验，从而能表达对称/重复几何导致的多模态不确定性。上行：初始粒子在参数空间中分散；中行：在“梯度吸引 + 核排斥”作用下粒子向多个高概率模式移动但不塌缩；下行：与点估计对比，粒子集可以同时覆盖多个等价解，避免把多模态问题强行压缩为单一解。

3.4.8 变换估计方法综合对比

表 11: 第 3.4 节 ICP 变换估计方法综合对比：参数化方案、可估计变换群、计算代价、不确定性表示能力与主要适用场景。

方法	参数化	可估计量	计算代价	不确定性	主要适用场景
SVD（Kabsch / [11]）	旋转矩阵	$SE(3)$	$O(1)$（3×3 SVD）	无	通用首选，数值最稳定
单位四元数（[12]）	四元数	$SE(3)$	$O(1)$（4×4 特征分解）	无	需插值/滤波序列
李代数（$se(3)$）	六维向量	$SE(3)$	$O(1)$ + 矩阵指数	无	Anderson 加速兼容
对偶四元数（[44]）	对偶四元数	Sim(3)	$O(1)$	无	含尺度估计
GICP（[26]）	旋转矩阵 + 协方差	$SE(3)$	$O(nk)$（协方差估计）	隐式（协方差）	曲面噪声大的场景
Stein ICP（[46]）	粒子集	$SE(3)$ 后验	$O(K^2n)$（K 粒子）	显式（完整后验）	安全关键，多模态

3.4.9 选择建议

对于大多数工程应用，SVD 闭式解（Kabsch/Arun）是默认选择——数值稳定、实现简单、3×3 矩阵分解几乎可以忽略的计算代价。当需要集成到 Kalman 滤波器或轨迹优化框架时，四元数或 $se(3)$ 表示更为自然。当

表 12: 第 3.4 节代表性“可复现设置 + 定量结果”汇总（覆盖本节正文出现的全部引用）。

文献	场景/数据集	指标口径	结果（数值）	关键设定（便于复现）
[11]	合成点对（点数 N 变化）	端到端求解时间（ms）	VAX 11/780：$N = 7/11/16/20/30$ 时，SVD 37.0/40.0/39.2/40.4/44.2 ms；四元数法 26.6/32.8/39.9/45.2/48.3 ms；迭代法 94.2/110.8/120.5/135.0/111.0 ms（迭代次数 5/7/10/6/6）	对比三类求解器：SVD、四元数特征分解、迭代法；表格
[12]	刚体配准闭式解	解的结构（矩阵规模/约束维数）	构造 4×4 对称矩阵 N，取最大特征值对应特征向量作为最优四元数	旋转用 4 维单位四元数；分解是固定规模特征分解（但 H 的构造仍需线性遍历点对）
[33]	FRICP 实验设置	终止条件（公平对比口径）	ICP / ICP-l / AA-ICP：最多 1000 次迭代，或 $\|\Delta T\|_F < 10^{-5}$	在 $se(3)$ 上做“加法更新”，便于 Anderson 加速；统一终止口径避免“停得早/晚”带来的假象
[44]	模拟 3D 曲线 + 真实点云	变换群维数与数据源	$SE(3)$ 6 自由度扩展到 Sim(3) 7 自由度；点名使用 PSB 与 Stanford Bunny	用对偶数四元数把旋转/平移/尺度并入同一框架；每轮迭
[26]	模拟 + Velodyne 实测扫描对	邻域规模、迭代预算与场景尺度	协方差用 20 近邻估计；ICP 迭代上限 250，P2Pl/GICP 上限 50；实测示例两帧约 30 m 间隔，量测范围约 70–100 m	以“点级协方差”把 P2P/P2Pl 统一到同一概率度量；内层求解用 GN/LM
[45]	10 m 正方形环境（欠约束分析示例）	位姿协方差（标准差）对照	真值：$(5.3\text{mm}, 5.3\text{mm}, 0.039°)$；闭式估计：$(5.4\text{mm}, 5.4\text{mm}, 0.042°)$；scan-matching 口径真值：$(7.6\text{mm}, 7.8\text{mm}, 0.058°)$，闭式估计：$(7.7\text{mm}, 7.7\text{mm}, 0.060°)$（真值位姿增量 $x = (0.1\text{m}, 0, 2°)$）	收敛附近线性化 + 噪声模型推导闭式协方差；讨论欠约束时可观测子空间的
[46]	RGB-D（碗/杯等对称物体）+ challenging LiDAR 场景	KL / OVL + 运行时	取 $K = 100$ 粒子；Bayesian ICP 1000 样本。Stein ICP 的 KL（中位数）约 0.6–5.7，OVL 约 0.7–0.9；运行时约为 Bayesian ICP 的 1/8–1/5，整体加速 >5×	SVGD 按粒子并行更新（GPU 友好）；能表达对称/重复几何导致的多模态后验

处理多传感器标定或多分辨率场景时，Sim(3) 的对偶四元数扩展是标准选择。当目标是提供位姿不确定性估计（如用于安全决策或融合）时，GICP 提供协方差近似，Stein ICP 提供完整后验——前者计算代价适中，后者精确但慢，适合离线后处理或高安全需求场景。

变换估计方法的选择与第 3.1 节 中的对应度量直接耦合：P2P + Kabsch 是最简单组合；P2Pl + 线性化 + Gauss-Newton 是工程中常用的高效组合；GICP 则自动在数据驱动下折中 P2P/P2Pl 的约束形态。第 3.6 节 将讨论在无可靠初始位姿时，如何通过全局方法为局部 ICP 提供可用的收敛起点。

3.5 几何退化与可定位性 (Geometric Degeneracy and Localizability)

第 3.4 节 的 Kabsch/SVD 求解在 $H = J^\top J$ 满秩时给出唯一的最优变换。然而，当点云的几何结构无法约束所有 6 个自由度时，H 在某些方向上奇异，ICP 的解沿这些方向变得任意，这便是**几何退化**（geometric degeneracy）。退化与外点（见第 3.2 节 ）的区别在于：外点问题源于“数据坏了”，可以通过鲁棒估计消掉；退化问题源于“点云本身在某些方向上没有足够几何约束”，再精细的对应也救不回来。自 [47] 在 ICRA 2016 将退化问题明确地拉到“可观测性/病态方向”这一层面后，围绕检测、量化与处理退化的研究形成了独立分支。Zhang 等在其自建的视觉 + 激光组合里程计实验里给了一个很直观的量化：在一段 538 m 的轨迹上，采用其 solution remapping 后，终点位置误差约为轨迹长度的 0.71%[47]。后续的 X-ICP 系列工作进一步把“哪几个自由度退化、退化到什么程度、如何把更新限制在可信子空间”做成了可复用模块：在 Seemühle 地下矿坑（VLP-16）的对照里，X-ICP 报告的终点误差为 0.27 m；而同一设置下，二值阈值法（Zhang）与更悲观的条件数检测（Hinduja）分别达到 6.37 m 与 24.17 m，这组对比几乎把“能不能用”直接分开了 [48]。

3.5.1 退化的数学本质

P2Pl ICP 的线性化目标函数在一次迭代内等价于求解正规方程：

$$H\,\delta x = b, \quad H = \sum_i n_i n_i^\top \otimes J_i^\top J_i \tag{42}$$

其中 n_i 为目标点法向量，J_i 为变换对应的点 Jacobian。H 是一个 6×6 正半定矩阵，其特征值分解 $H = U\Sigma U^\top$ 揭示了优化的几何含义：特征值 σ_k 越大，对应方向 u_k 上的约束越强；$\sigma_k \to 0$ 则意味着无论 δx 在 u_k 方向上取何值，目标函数几乎不变——ICP 更新量在该方向上无意义。

这种“用 Hessian 谱刻画约束强弱/可观测性”的观点也常用于 ICP 的不确定性建模，例如闭式协方差估计中会显式讨论欠约束方向如何影响位姿不确定性 [45]。Censi 在一个 $10\,\mathrm{m}\times 10\,\mathrm{m}$ 的方形环境里用 $x = (0.1\,\mathrm{m}, 0, 2^\circ)$ 的位姿扰动做示例，把闭式协方差的结论直接写成“毫米/角秒”级别的数字：在欠约束分析口径下，真值标准差约为 $(5.3\ \mathrm{mm}, 5.3\ \mathrm{mm}, 0.039^\circ)$，闭式估计为 $(5.4\ \mathrm{mm}, 5.4\ \mathrm{mm}, 0.042^\circ)$；而在 scan-matching 口径下，真值约为 $(7.6\ \mathrm{mm}, 7.8\ \mathrm{mm}, 0.058^\circ)$，闭式估计为 $(7.7\ \mathrm{mm}, 7.7\ \mathrm{mm}, 0.060^\circ)$（论文表格直接对照）[45]。这组数字的意思很朴素：一旦某个方向的几何约束变弱，协方差会立刻“鼓起来”，这不是求解器写错了，而是信息本身不够。

退化几何的典型模式。走廊（无限长直壁）：所有法向量集中于水平面内，法向量矩阵的零空间包含沿走廊轴的平移方向，H 有 1 个近零特征值；圆形隧道：法向量分布于以轴为中心的圆柱面，H 有 2 个近零特征值（轴向平移 + 绕轴旋转）；开阔平面（停机坪）：仅有竖直法向量，H 有 5 个近零特征值，仅约束竖直方向平移。

对应关系的物理意义。[47] 将退化问题推广到一般基于优化的状态估计：若 J 的 SVD 的最小奇异值 $\sigma_{\min} < \tau$，则相应状态分量的估计不可靠。这一判据直接适用于 P2Pl ICP，其中 J 即为点到平面残差关于 δx 的 Jacobian。

3.5.2 退化检测方法

退化检测的目标是在 ICP 求解前（或求解中）识别哪些自由度约束不足，以便在这些方向上引入先验约束或触发传感器融合。

特征值阈值法（[47]）最直接：计算 H 的最小特征值 $\sigma_{\min}$，若低于阈值 τ，就判为退化，把更新量投影到非退化子空间。问题也出在这根阈值上：它既受场景尺度和点云密度影响，又很难给出“一劳永逸”的经验值。后续工作在复现实验时往往不得不把它当作显式超参去扫：LP-ICP 的消融里，同一条序列把阈值 Thr 从 50 增到 100，RMSE（ATE）会从 36.60 m 直接飙到 195.57 m，几乎等于把系统推向失效边缘 [49]；X-ICP 的对照

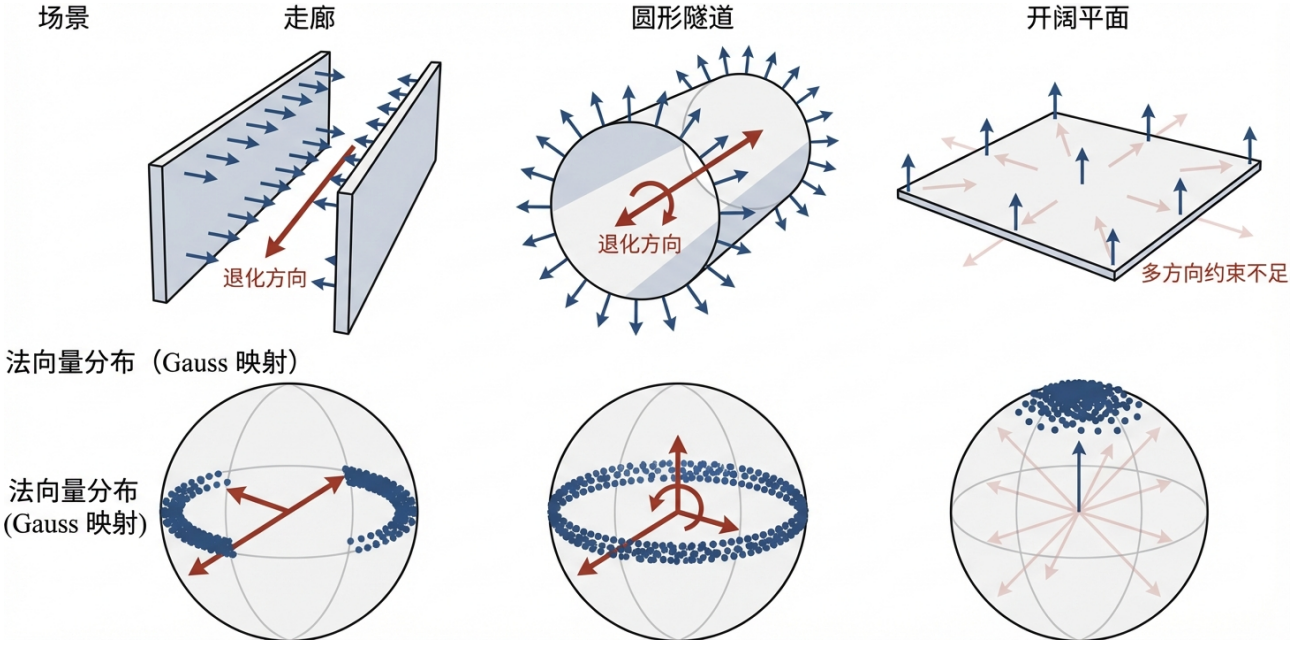


图 22: 三类典型几何退化场景及其法向量分布与信息矩阵谱的定性关系。上行：走廊通常出现“单一方向约束不足”（轴向平移不易被平面法向约束）；圆形隧道可能同时缺失轴向平移与绕轴旋转的约束；开阔平面则只约束少数自由度，更多方向呈现近零曲率。下行：对应的 Gauss 映射（法向量在单位球面上的分布）直观解释了“约束来自哪些法向集合”，并与 Hessian 的近零特征方向一一对应。

实验同样指出，如果不针对环境调阈值（文中给出的区间是把 Thr 从 120 调到 200），Zhang 的二值退化检测很容易在回程隧道段出现 LiDAR slip，导致终点误差到 6.37 m，而 X-ICP 同设置下仅 0.27 m [48]。因此，阈值法更像一把“能用但难伺候”的扳手：一旦环境换了、点云密度换了，扳手的刻度就不再对。

它的局限不只是“参数敏感”这么简单，而是这种敏感往往没有明显先兆：同一根阈值在一段走廊里还有效，换到开阔区或稀疏矿道就可能突然开始误报或漏报；一旦检测结果错了，后面的约束策略也会跟着一起错。

X-ICP 多类别可定位性分析（[48]）将退化细分为三类：完全可定位（6 DOF 均约束充分）、部分可定位（部分 DOF 退化）和不可定位（严重几何对称）。对每一个对应关系 (p_i, q_i, n_i)，X-ICP 计算其在各主方向 u_k 上贡献的约束强度 $s_{ik} = |n_i^\top J_i u_k|^2$，聚合后得到每个方向的可定位性分数（localizability score）$L_k = \sum_i s_{ik}$。这种“方向分辨”的输出不是为了好看，而是直接能对接后端约束提交：在 Seemühle 地下矿坑（VLP-16）的对照里，X-ICP 报告的终点误差仅 0.27 m，而二值阈值法（Zhang）和更悲观的条件数检测（Hinduja）分别到 6.37 m 和 24.17 m，差距几乎把“能不能用”分开了 [48]。

概率退化检测（[50]）对 P2Pl Hessian 的不确定性建模：H 由含噪点坐标和法向量构成，其扰动 δH 的分布可由传感器测量噪声推导。由此将” 某方向是否退化” 转化为带置信度的判别（例如：$P(\sigma_k < \tau)$ 超过触发阈值），从而把“特征值阈值法”的经验调参，替换为与噪声模型相一致的概率声明。作者在其四组真实场景实验中展示了更稳定的触发行为与更好的退化缓解效果；同时也给出了运行代价的量化：在 Intel i7-12800H 上，实验 2 的 LOAM 扫描匹配中位运行时间为 17 ms，其中退化检测开销占比为 5.4%；在 Khadas VIM4 上使用单个 A73 核心时，中位运行时间为 54 ms [50]。

这类方法的局限在于噪声模型本身的准确性至关重要。若点坐标噪声、法向误差或时间同步误差的统计假设与真实系统存在显著偏差，则“概率更合理”的声明也可能仅仅是建立在错误模型上的精细化计算。

点分布退化检测（[51]）用点到分布（point-to-distribution）匹配替代点到平面，通过自适应体素分割捕获点云的局部几何模型。该方法对噪声更鲁棒（分布估计平滑了单点扰动），并在论文实验中以“误检测次数/Precision-Recall”等指标对比特征值类方法，强调其能降低“噪声诱发误报警”的问题：例如在 M2DGR 的 street 01 场景中给出误检测次数从 125 降至 10；在走廊类场景的 Precision-Recall 实验中，报告 Recall 从 0.7178 提升到 0.9568 [51]。

代价是它把退化检测又往前推了一层建模：体素划分、分布拟合和邻域统计一旦设置不合适，检测结果同

样会跟着漂。相比直接看谱，它更稳，但实现链条也更长。

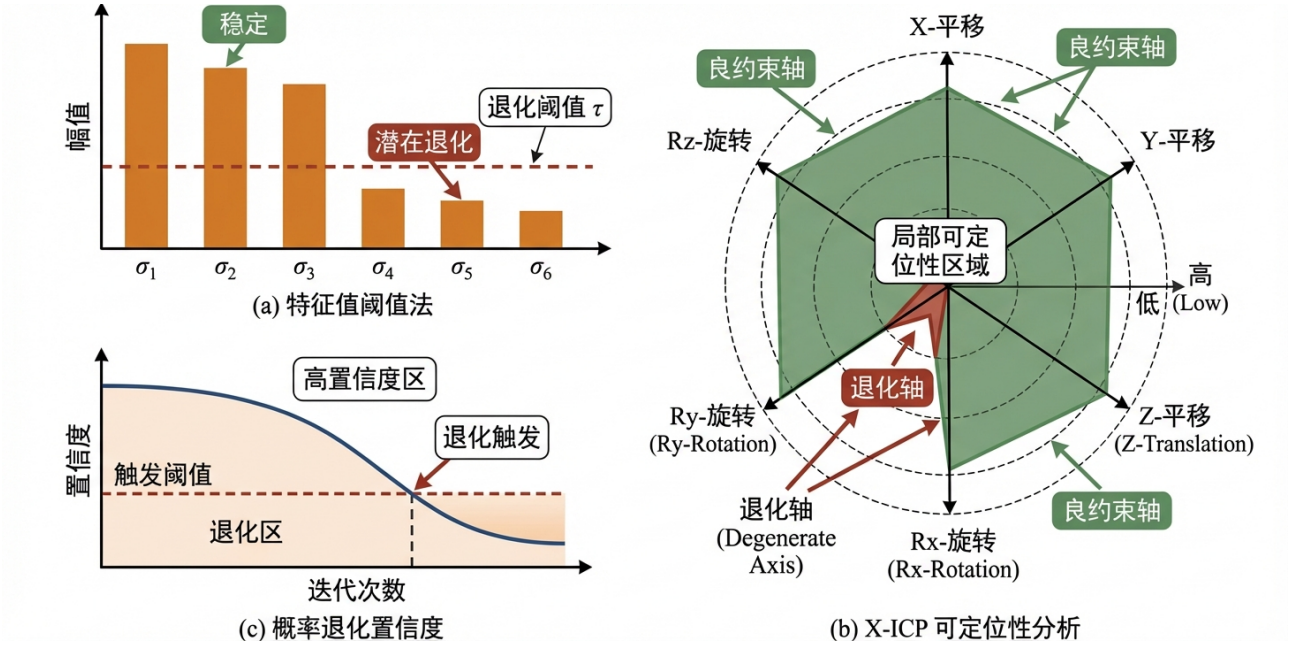


图 23: 三种退化检测范式的对比示意。左：特征值阈值法通过“谱是否过小”给出整体退化判别，简单但易受尺度与密度影响；中：X-ICP 将约束强度分解到 6 个自由度方向，以雷达图形式输出可定位性分数，便于后续“只在可信方向提交约束”；右：概率退化检测输出各方向的退化置信度随迭代的变化，用于在系统中触发软约束/传感器融合。

3.5.3 退化方向的约束优化

检测到退化后，ICP 需要在退化方向上引入替代约束，而非直接求解病态线性系统。主要策略分为” 主动约束”（修改优化更新方向）和” 软约束”（引入先验惩罚项）两类。

截断奇异值分解（Truncated SVD, TSVD）是最早也是最简单的主动约束方案：对 $H = U\Sigma V^\top$ 截断奇异值，令小于阈值的奇异值置零，得到伪逆 $H^+ = V\Sigma^+U^\top$，位姿更新变为 $\delta x = H^+ b$。效果等价于将更新量投影到良约束子空间，退化方向的更新量强制为零。缺陷是零更新量意味着在退化方向上完全不移动，等价于隐式施加” 退化方向位移为零” 的硬约束，与实际场景（走廊中仍需沿轴向移动，只是来自 ICP 的约束不可靠）不匹配。

也正因为这个原因，TSVD 更适合短时“保守止血”，不适合长时间连续运行在单向退化环境里。时间一长，真实运动和“被归零的更新”之间的偏差会持续累积。

Tikhonov 正则化（软约束）在目标函数中加入先验惩罚：

$$\delta x^* = \arg\min_{\delta x} \|J\delta x - r\|^2 + \lambda_k \|\delta x - \delta x_{\text{prior}}\|_W^2 \tag{43}$$

其中 δx_{prior} 来自 IMU 预积分或恒速运动外推，正则化权重 λ_k 与退化程度（例如奇异值越小，越“拉向先验”）成正比。与 TSVD 的区别在于：退化方向不被归零，而是被“拉向先验”，既避免了硬投影带来的不连续，也给了系统在退化方向上继续前进的空间。[52] 在 2025 年的野外系统评测里把这件事说得很实在：在 Ulmberg 隧道这种“长时间单向退化”的工况下，基线 P2Plane 的 ATE 到 2.90 m，而非线性正则（NL-Reg.）能压到 1.097 m；对应的局部误差（RTE）上，不等式约束与 NL-Reg. 的最好结果分别是 0.033 m 与 0.035 m（Prior only 为 1.54 m）。在 ANYmal 森林实验里，如果只依赖腿部里程计先验，ATE 为 0.662 m；加入正则后，NL-Reg. 与 NL-Solver 分别到 0.364 m 与 0.342 m（Eq. Con. 为 0.490 m）[52]。这些数字背后的取舍也很明确：NL-Reg. 的计算代价约是线性方法的 5 倍（作者报的量级是 80 ms vs 20 ms），是否采用非线性正则化，取决于系统能否承受将 10 Hz 预算中约一半的时间分配给该模块。

X-ICP 紧耦合约束优化（[48]）将可定位性分析结果直接嵌入 ICP 优化步骤：构造方向选择矩阵 $S_{\mathcal{L}}$（保留良约束方向）和 $S_{\mathcal{D}}$（退化方向），将变换更新分解为：

$$\delta x = S_{\mathcal{L}}\, H_{\mathcal{L}}^{-1} b_{\mathcal{L}} + S_{\mathcal{D}}\, \delta x_{\text{ext}} \tag{44}$$

其中 δx_{ext} 来自外部传感器（IMU、轮速计）对退化方向的约束。良约束方向的更新无漂移（受 ICP 精确约束），退化方向的更新来自外部先验，实现两者的零漂移组合。

它的局限非常现实：一旦外部先验本身带偏，系统就会把这个偏差明确地灌进退化方向。X-ICP 的优势来自“知道该相信谁”，但这也意味着先验质量一差，收益会迅速缩水。

不等式约束（[52]）把退化方向的更新写成界约束 $|(\delta x)_k| \le \epsilon$，再用 QP 去解。这里的 ϵ 就是工程上能直接把握的“我最多允许它在这条方向上抖多少”。在 Ulmberg 隧道上，作者给出的折中值是 $\epsilon = 0.0014$，并配套报告了正则项的调参（例如线性正则 $\lambda = 440$，非线性正则 $\lambda_D = 675$）；他们还做了先验噪声敏感性测试：把平移先验噪声拉到 $\sigma_t = 0.05\,\text{m}$ 后，除 Eq. Con. 外多数方法会发散 [52]。这组实验让“不等式/正则化到底在吃什么信息”更清楚：不是它们本身多神奇，而是它们把系统对外部先验质量的依赖暴露成了可量化的边界条件。

其局限同样明确：约束过紧会将真实运动一并过度限制，约束过松则无法有效抑制退化方向的漂移。与 TSVD 相比，该策略更为平滑，但本质上仍是在“安全性”与“机动性”之间进行权衡。

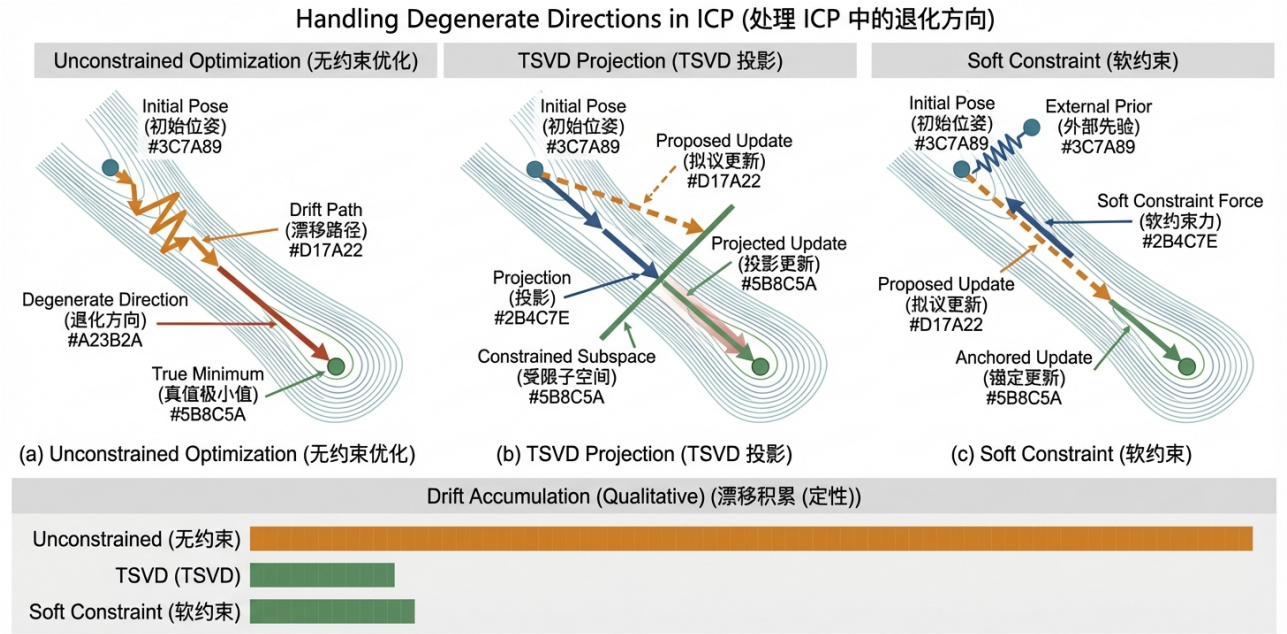


图 24: 退化方向约束优化三策略的更新量几何直观（二维示意）。左：无约束 ICP 的等值线在退化方向形成“长槽”，更新量可在槽内滑动导致漂移；中：TSVD 将更新量投影到非退化子空间，能抑制漂移，但同时压制了退化方向的真实运动；右：软约束（如 Tikhonov）将外部先验作为“锚点”把长槽收紧，使更新在退化方向随先验变化而非任意漂移。下方以定性曲线示意三者的漂移累积趋势。

3.5.4 退化鲁棒的 ICP 变体

除在优化步骤引入外部约束外，另一类方法通过算法自身的自适应机制降低对外部传感器的依赖，在纯 LiDAR 场景下提升鲁棒性。

GenZ-ICP（[53]，RA-L 2025）重新审视 P2P 与 P2Pl 两种误差度量的互补性：P2Pl 在平面丰富的环境（室内房间）精度高，但在走廊中法向量平行导致退化；P2P 在走廊场景中仍提供轴向约束（点到点距离不受法向量分布影响）。GenZ-ICP 根据当前帧的局部几何特征（平面点比例、法向量集中度）自适应计算融合权重 $w_{\text{P2Pl}}, w_{\text{P2P}}$（两者之和为 1）：

$$\mathcal{L}_{\text{GenZ}} = w_{\text{P2Pl}} \sum_i (n_i^\top (Rp_i + t - q_i))^2 + w_{\text{P2P}} \sum_i \|Rp_i + t - q_i\|^2 \tag{45}$$

走廊场景自动偏向 P2P（$w_{\text{P2P}} \to 1$），平面丰富场景自动偏向 P2Pl（$w_{\text{P2Pl}} \to 1$），无需外部传感器或手动切换。原文在 SubT-MRS 的 Long_Corridor 序列上也给了直观的量化对比：Point-to-plane ICP 的 APE 均值为 32.84 m、Point-to-point ICP 为 6.83 m，而 GenZ-ICP 将 APE 降至 1.69 m（同表 CT-ICP 为 44.18 m，Zhang et al. 为 19.43 m）[53]。

它的局限是仍然把希望押在“两种经典度量的加权足够覆盖主要退化模式”上。若场景几何比“走廊 vs 平面”更复杂，或者法向统计本身不稳定，自适应权重就未必能准确反映真实可观性。

LP-ICP（[49]，2025）把 X-ICP 的可定位性分析从“只盯着点到平面”扩展到“线 + 面一起算”：边缘点（低平滑度）通过点到线约束补上走廊轴向信息，平面点（高平滑度）继续提供法向约束。它的优势主要体现在“平面约束不够、但线特征还在”的极端工况里：作者在 PLAM 的 `a2_traverse` 上报告 ATE 从 36.60 m 降到 12.31 m；在 SubT-MRS 的长走廊序列上从 24.71 m 降到 11.92 m；在 PLAM `a3_odom` 上从 18.08 m 降到 7.44 m；在 CERBERUS 的 `ANYmal 2` 上从 0.32 m 降到 0.24 m。更关键的是它没有靠“加很多算力”换结果：在 Intel i7-12700H 上，scan-to-map 配准平均耗时 35.87 ms，与 LVI-SAM（38.04 ms）同量级 [49]。

DAMM-LOAM（[54]，2025）走的是“先把几何结构分清楚，再谈退化”的路线：用球面投影的法向量图把点云分成地面、墙面、屋顶、边缘与非平面点五类，再把不同类别的残差按退化程度做加权融合。它在走廊类数据上的量化很有代表性：在 SubT-MRS 的 Long_Corridor 序列上，APE RMSE 报告为 1.72 m（KISS-ICP 为 8.72 m，GenZ-ICP 为 1.99 m），APE Max 为 4.08 m；在 Ground-Challenge 的 Corridor1 上 APE RMSE 仅 0.06 m（GenZ-ICP 0.24 m，KISS-ICP 2.17 m），Corridor2 为 0.08 m（GenZ-ICP 0.20 m）[54]。上述结果表明：在走廊类场景中，即便同为 LiDAR-only 方案，能否有效区分墙面、地面与边缘等不同结构类别的约束，直接决定了退化方向上的漂移是否会被放大。

但这类方法也更依赖前面的结构划分是否稳定。一旦类别分错了，后面的加权就会建立在错误的几何理解上，系统复杂度也会明显高于“直接做一个统一 ICP”。

退化感知位姿图因子（[27]，IROS 2019）把“只在可信方向提交约束”这件事落到了因子图里：退化感知 ICP 不再硬塞一个全 6-DOF 的相对位姿，而是把信息矩阵在退化方向对应的分量置零，只保留良约束子空间的约束，让后端优化明确知道“哪些方向这条边说不准”。他们在水下声纳 SLAM（DIDSON 声纳，强退化几何）里给了一个很工程的指标：在某组数据上，动态阈值机制会直接拒绝 25 个错误闭环，从源头上避免把退化对齐的“假约束”灌进图优化里 [27]。

3.5.5 系统融合视角：退化检测与传感器切换

退化检测的最终目的不是“打标签”，而是触发合适的约束提交与传感器融合策略。[55] 在 2025 年的退化环境 SLAM 综述里把这类系统抽象成“感知–决策”的闭环：一方面，GNSS 定位至少需要 4 颗卫星信号，一进隧道/矿井就天然缺观测；另一方面，多传感器融合在工程上几乎离不开高频 IMU（常见 >100 Hz）去填补 LiDAR/视觉较低频帧率的间隙。于是退化检测一旦触发，系统做的不是“继续硬解一个 6-DOF”，而是把不可靠的方向交给更合适的先验或传感器来约束：

1. **检测阶段**：X-ICP 可定位性分析 / 概率检测给出每个方向的退化程度。
2. **决策阶段**：根据退化程度和可用传感器选择处理策略——轻微退化可用 Tikhonov 软约束稳定更新；严重退化且有 IMU/里程计时采用紧耦合约束优化（X-ICP 模式）提交“部分可信”的位姿增量；严重退化且缺乏外部先验时，可切换到算法自适应的度量（如 P2P+P2Pl 融合）或使用 TSVD 投影并降低因子权重。
3. **验证阶段**：系统级指标（配准残差、IMU 一致性检验）验证约束策略是否有效，动态调整退化检测阈值。

这一框架使 LiDAR SLAM 系统能够在退化环境中实现” 优雅降级”（graceful degradation）而非硬性失败。

表 13: 第 3.5 节退化感知 ICP 方法综合对比：退化检测类型、优化处理策略、是否依赖外部传感器及代表性测试场景。

方法	退化检测类型	处理策略	需外部传感器	代表场景
Zhang et al. [47]	特征值阈值	TSVD 投影	否	走廊（理论分析）
X-ICP [48]	多类别可定位性	紧耦合约束优化	是（IMU/里程计）	地下矿山、隧道
Hatleskog & Alexis [50]	概率 Hessian 噪声	平滑衰减更新	否	走廊、室外开阔区
Ji et al. [51]	点到分布自适应体素	检测触发融合	可选	走廊、隧道
GenZ-ICP [53]	隐式（自适应权重）	P2P+P2Pl 联合度量	否	走廊（纯 LiDAR）
LP-ICP [49]	线 + 面多类别	扩展 X-ICP 约束	是（IMU/里程计）	非结构化极端环境
Hinduja et al. [27]	特征值（ICP 内）	部分约束位姿图因子	否（位姿图级）	水下声纳 SLAM
Tuna et al. 2025 [52]	多方法比较	TSVD/Ineq/Tikhonov	是（IMU）	系统评测综合场景

表 14: 第 3.5 节代表性“可复现设置 + 定量结果”汇总（覆盖本节正文出现的全部引用；仅摘录原文或原文综述中口径清晰的报数）。

文献	场景/数据集	指标口径	结果（数值）	关键设定/前提
[47]	视觉 +LiDAR 自建里程计测试（文中 Test 3）	终点位置误差（相对轨迹长度）	轨迹长度 538 m；终点位置误差约为 0.71%	用特征分解识别退化方向，并用 solution remapping 在退化方向用 best guess 替代求解
[45]	10 m × 10 m 方形环境（论文数值示例）	位姿协方差（平移/旋转标准差量级）	扰动 $x = (0.1\,\mathrm{m}, 0, 2^\circ)$；示例中 σ_t 量级约 10^{-3} m、σ_r 量级约 10^{-2} 度	闭式协方差估计强调欠约束方向会导致不确定性
[48]	Seemühle 地下矿坑（VLP-16）	End Position Error + APE/RPE	End Pos. Error 0.27 m；APE 平移均值/标准差 2.45(1.35) m；RPE(10 m) 平移 0.17(0.12) m	多类别 localizability 检测 + 方向选择矩阵；对比 Zhang/Hinduja（同场景 End Pos. Error 6.37 m/24.17 m）
[50]	4 组真实场景（论文实验 2 含 LOAM scan matching）	运行时与检测开销占比	i7-12800H：中位 17 ms，检测开销占比 5.4%；VIM4 单 A73：中位 54 ms	用噪声模型推导 Hessian 扰动分布，把触发写成概率事件
[51]	M2DGR street 01 + 走廊类场景	误检测次数 + Precision/Recall	street 01 误检测次数 125 → 10；Recall 0.7178 → 0.9568	点到分布模型 + 自适应体素，抑制噪声诱发的误报警
[53]	SubT-MRS Long_Corridor	APE 均值（m）	P2Pl 32.84 m；P2P 6.83 m；GenZ-ICP 1.69 m	自适应融合 P2P/P2Pl，走廊自动偏向 P2P，平面丰富区域偏向 P2Pl
[49]	PLAM + SubT-MRS + CERBERUS	ATE + 运行时间	`a2_traverse` 36.60 → 12.31 m；长走廊 24.71 → 11.92 m；`a3_odom` 18.08 → 7.44 m；ANYmal 2 0.32 → 0.24 m；平均 35.87 ms（i7-12700H）	点到线 + 点到平面联合 localizability；并报告 Thr 从 50 到 100 会把 RMSE 从 36.60 m 拉到 m（阈值敏感性）
[54]	SubT-MRS Long_Corridor + Ground-Challenge Corridor1/2	APE RMSE（m）与 Max	Long_Corridor：APE RMSE 1.72 m（KISS-ICP 8.72 m，GenZ-ICP 1.99 m），APE Max 4.08 m；Corridor1 0.06 m（GenZ-ICP 0.24 m），Corridor2 0.08 m（GenZ-ICP 0.20 m）	五类几何分类 + 退化加权 WLS + Scan Conte
[27]	水下声纳 SLAM（DIDSON）	错误闭环拒绝数量（示例）	某组数据直接拒绝 25 个错误闭环	退化 ICP 输出以“部分约束因子”入图，退化方向信息矩
[52]	ANYmal 森林 + Ulmberg 隧道	ATE/RTE + 参数与时延	Ulmberg：P2Plane ATE 2.90 m，NL-Reg. 1.097 m；Ineq. Con. RTE 0.033 m；调参 $\epsilon = 0.0014$、$\lambda = 440$、$\lambda_D = 675$；NL-Reg. 约 80 ms vs 线性法约 20 ms；先验噪声 $\sigma_t = 0.05\,\mathrm{m}$ 时多法发散	系统性对比 TSVD/Ineq./Tikhonov，强调主动退化缓解与
[55]	退化环境 SLAM 综述（GNSS 拒止/视觉退化/特征退化等）	退化场景下的观测与融合节拍（综述中的工程事实）	GNSS 至少需要 4 颗卫星信号；多传感器融合常依赖 >100 Hz IMU 补低频 LiDAR/视觉	用“感知–决策”框架讨论退化检测触发的策略

3.5.6 方法综合对比

退化检测 vs 算法自适应的边界正在模糊：GenZ-ICP 和 LP-ICP 展示了通过丰富度量本身（而非显式检测-处理分离）来” 内生性” 抵抗退化的可行性，未来方向是将退化感知能力嵌入 ICP 的每一次迭代，使其对几何结构的变化自动响应，无需外部触发。

3.6 全局初始化与两阶段配准框架 (Global Initialization and Two-Stage Registration)

ICP 是局部方法：它只能在当前估计附近搜索极小值，无法跨越势垒到达全局最优。这意味着在未知初始位姿的通用场景中，单独依赖 ICP 常会陷入错误盆地。第 2.2.4 节已从收敛盆地角度解释了这种敏感性。本节关注“如何把误差压入盆地”：全局配准方法通过不同策略提供粗初始值（特征匹配 + 鲁棒估计）、或通过全局搜索/可认证优化获得可靠解，再交由 ICP 完成局部精修。

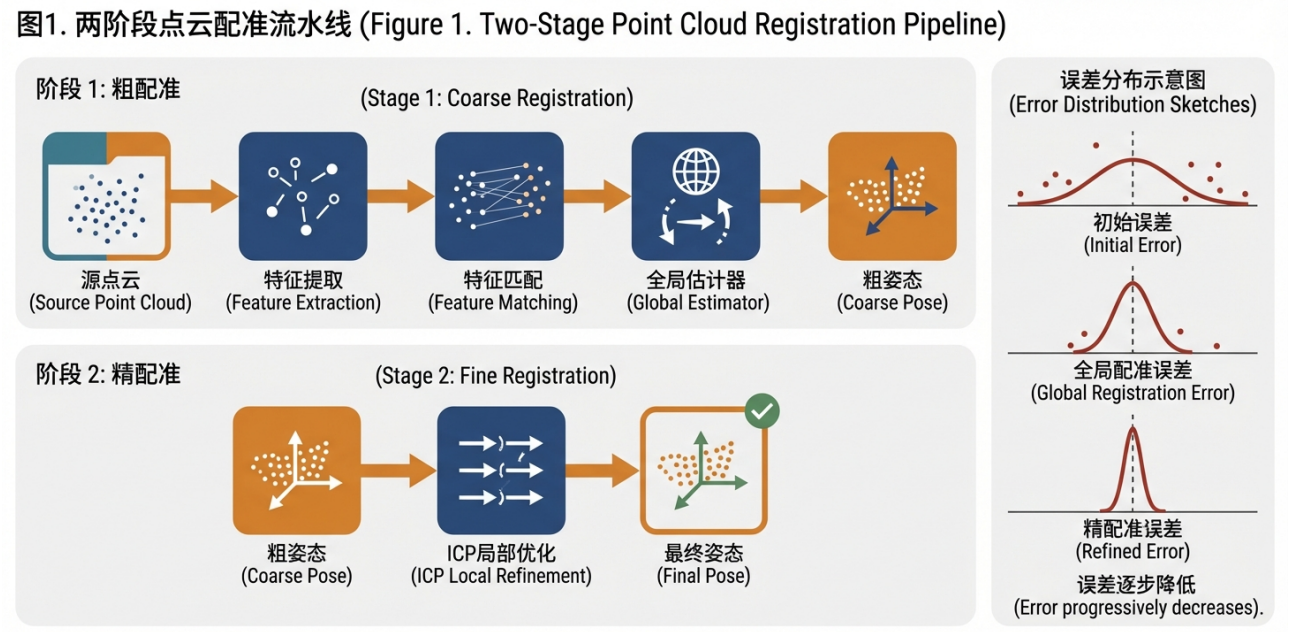


图 25: 全局初始化 + ICP 精修的两阶段 pipeline 示意图。上行：特征提取（如 FPFH/FCGF）→ 特征匹配 → 全局估计（如 RANSAC/FGR/Go-ICP/TEASER++）→ 粗位姿 T_{global}。下行：以粗位姿为初始值进行 ICP 局部精修，输出 T_{final}。右侧以“误差分布逐步收敛”的形式示意：初始化误差较大，粗配准后显著收缩，ICP 精修后进一步集中。

3.6.1 两阶段框架的数学基础

全局粗配准 + ICP 精修的两阶段 pipeline 可以形式化为：

$$T_{\text{final}} = \underbrace{\text{ICP}(T_{\text{global}}, \mathcal{P}, \mathcal{Q})}_{\text{局部精修（见第 3.1–3.4 节）}} \circ \underbrace{T_{\text{global}}}_{\text{全局粗配准（本节）}} \tag{46}$$

两阶段的” 劳动分工” 可以概括为：全局阶段负责把初始误差压入 ICP 的可收敛区域，全局方法更重视鲁棒性与覆盖性；随后由 ICP 在局部盆地内完成高精度精修。以 FGR 为例，[28] 在 UWA benchmark（50 个场景、188 对配准测试，最低重叠率约 21%）上报告 0.05-recall 达到 84%；在这类“先把大误差压下去”的设置里，ICP 更像最后的几何抛光工序：负责把粗配准收进某个局部极小附近，而不是从零开始兜底全局搜索。

3.6.2 局部几何描述子

全局配准的第一步是为每个点提取旋转不变的局部几何特征，再通过特征最近邻匹配生成候选对应集。描述子的判别力（区分不同几何位置的能力）和鲁棒性（对噪声、密度变化的不敏感性）决定了候选对应集的质量。

3.6.2.1 FPFH　FPFH（Fast Point Feature Histograms） [9] 是点云配准领域最广泛使用的手工描述子。对于点 p 及其 k 个近邻，计算每对 (p, p_j) 的三个角特征：

$$\alpha = \mathbf{v} \cdot \mathbf{n}_j, \quad \phi = \frac{\mathbf{u} \cdot (p_j - p)}{\|p_j - p\|}, \quad \theta = \arctan(\mathbf{w} \cdot \mathbf{n}_j, \mathbf{u} \cdot \mathbf{n}_j) \tag{47}$$

其中 $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ 为由 p 的法向量和 p–p_j 连线定义的局部坐标系，$\mathbf{n}_j$ 为近邻点 p_j 的法向量。三个角特征各量化为 11 个 bin，组成 33 维直方图，对刚体变换严格不变。

FPFH 的速度优化：相比原始 PFH（$O(k^2)$），FPFH 先为每个点独立计算 SPFH（simplified PFH，$O(k)$），再以 SPFH 的加权组合得到 FPFH（$O(k)$），从而显著降低计算代价，使其更适合作为工程中的默认手工描述子 [9]。

在实现层面，[9] 还在 bunny00 数据集的复杂度分析中展示了“重排序 + 计算缓存”的收益：对 PFH 而言，把点云索引按空间连续性重排后再用 FIFO 缓存重用中间量，计算时间可降低约 75%（图中对比了乱序/重排两种情况）。

局限性：FPFH 是纯几何描述子，在平面主导或高重复结构中判别力较弱，容易产生大量歧义匹配，从而显著增加后续鲁棒估计（如 RANSAC）的负担。

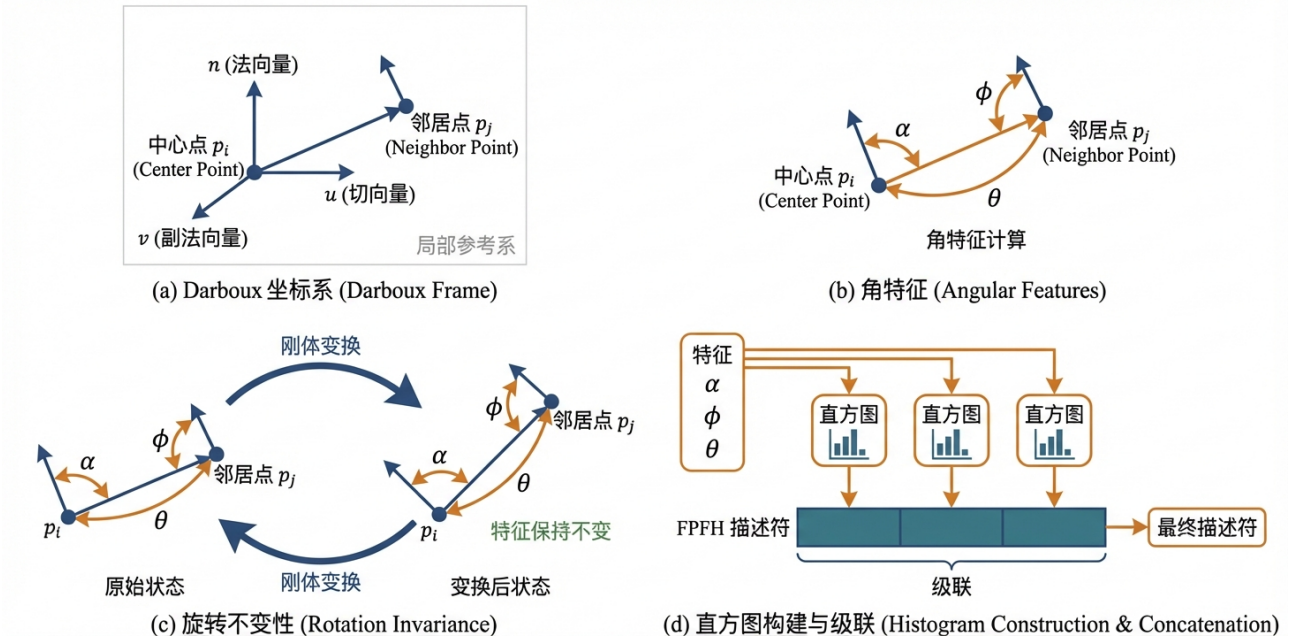


图 26: FPFH 特征提取的几何机制示意。(a) 以中心点法向与邻域连线构造 Darboux 参考系；(b) 在该参考系下定义三种角特征 (α, ϕ, θ)；(c) 刚体变换下参考系随点一起旋转，角特征保持不变；(d) 将角特征统计为直方图并拼接为局部描述子，用于特征匹配。

3.6.2.2 FCGF　FCGF（Fully Convolutional Geometric Features） [56] 将点云转换为稀疏体素表示，以深度卷积网络（Minkowski Engine 稀疏卷积）提取 32 维描述子。训练目标函数为度量学习的对比损失：

$$\mathcal{L} = \sum_{\text{正对}} \max(0, \|f_i - f_j\|_2 - m_p) + \sum_{\text{负对}} \max(0, m_n - \|f_i - f_j\|_2) \tag{48}$$

使得对应点（正对）的特征距离小于 m_p，非对应点（负对）的特征距离大于 m_n。在 3DMatch 基准（阈值 $\tau_1 = 0.1\,\text{m}$、$\tau_2 = 0.05$）上，[56] 报告 32 维 FCGF 的 Feature Match Recall（FMR）为 0.952（STD 0.029），同表中 FPFH 的 FMR 为 0.359（STD 0.134）。在同一工作站上统计的特征提取耗时也给得很细：原文表 1 中按“每个特征”的时间计，FCGF 为 0.019 ms；按“每帧片段”的体素化提取计，2.5 cm 体素约 0.36 s、5 cm 体素约 0.17 s。

同一论文也把“跨场景/跨传感器”的落地成本摆在明面上：作者在 KITTI 上是单独训练并评测的，并把成功条件固定为 `RTE < 2 m` 且 `RRE < 5°`。在 hardest-contrastive 训练下，20 cm 体素的 FCGF + RANSAC 得到 RTE=4.881 cm、RRE=0.170°、成功率 97.83%；对比 3DFeat 的 RTE=25.9 cm、RRE=0.57°、成功率 95.97% [56]。

它的局限也正写在这组结果里：一旦场景、传感器或采样尺度变了，学习到的描述子往往需要重新适配。FCGF 比 FPFH 判别力强，但也更依赖训练分布，不能把“在 3DMatch/KITTI 上有效”直接等同于“换域后仍稳”。

3.6.3 RANSAC 与对应过滤

随机采样一致性（RANSAC）是将任意特征描述子与刚体变换估计对接的通用框架。每次从候选对应集随机选取 3 对点（最小子集），用 Kabsch/SVD 估计变换 T_{hypo}，统计与 T_{hypo} 一致的内点数量，取最多内点的假设为最终结果：

$$T^* = \arg\max_{T_{\text{hypo}}} \left|\{(p_i, q_j) : \|T_{\text{hypo}}(p_i) - q_j\| < \tau\}\right| \tag{49}$$

RANSAC 的迭代次数下界由外点率 ϵ 和目标成功率 P 决定：

$$k \geq \frac{\log(1-P)}{\log(1-(1-\epsilon)^s)} \tag{50}$$

其中 $s = 3$ 为最小子集大小。该式表明：当外点比例升高时，达到固定置信度所需的迭代次数会快速增长，尽管假设评估可并行化，整体仍可能成为粗配准阶段的主要瓶颈。RANSAC 的优势是通用、实现简单且具备概率意义上的置信度控制；其局限在于高外点率场景下效率不稳定，并高度依赖候选对应集的质量。

3.6.4 快速全局配准（FGR）

FGR（Fast Global Registration） [28] 规避了 RANSAC 的随机采样，直接以所有候选对应同时优化 Geman-McClure 鲁棒目标函数：

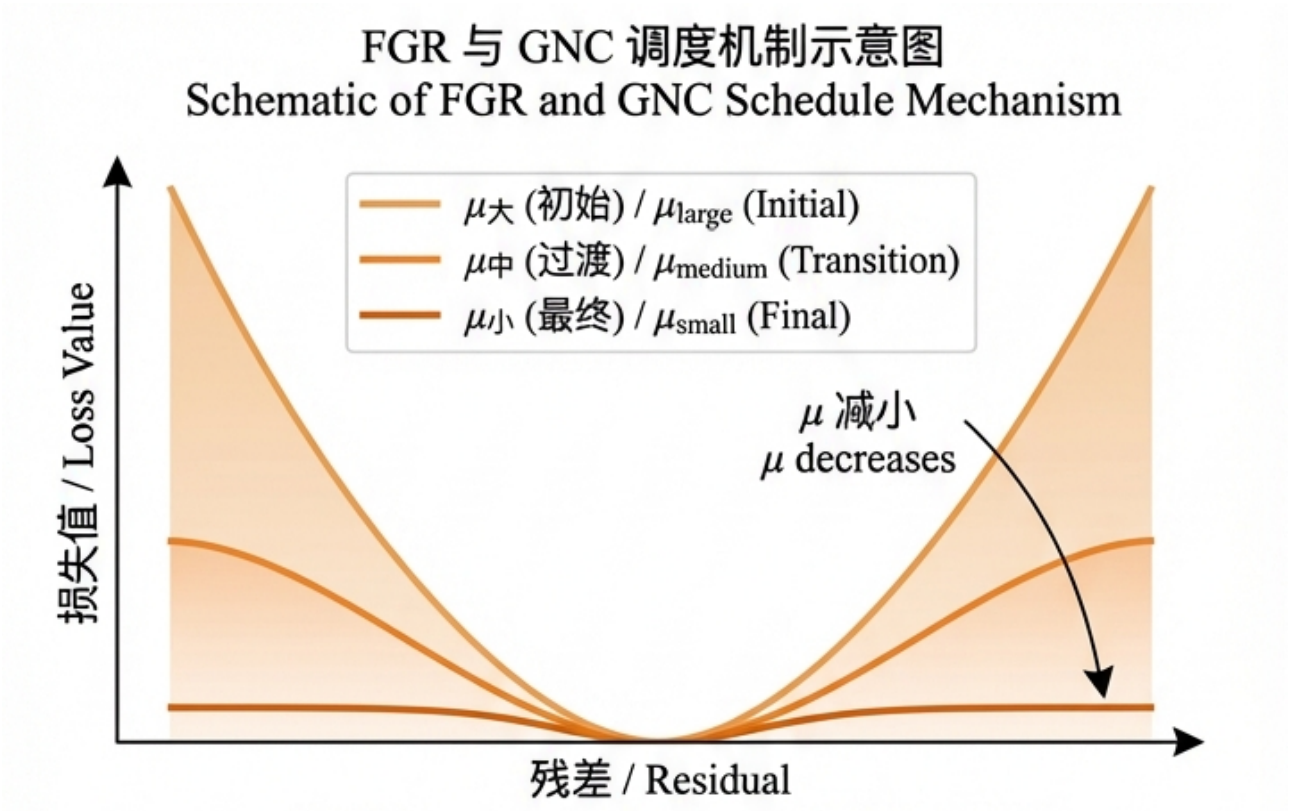
$$\mathcal{E}_{\text{FGR}}(T) = \sum_k \Phi_\mu(\|T(p_k) - q_k\|), \quad \Phi_\mu(x) = \frac{\mu x^2}{\mu + x^2} \tag{51}$$

其中 Φ_μ 为 Geman-McClure 函数，对大误差（外点）提供饱和损失，对小误差（内点）近似 L_2。FGR 采用渐进非凸化（GNC）策略：从 $\mu \to \infty$（近凸）逐步减小 μ 到目标值，每步以前一步结果热启动，避免陷入局部极小值。FGR 的关键性质：

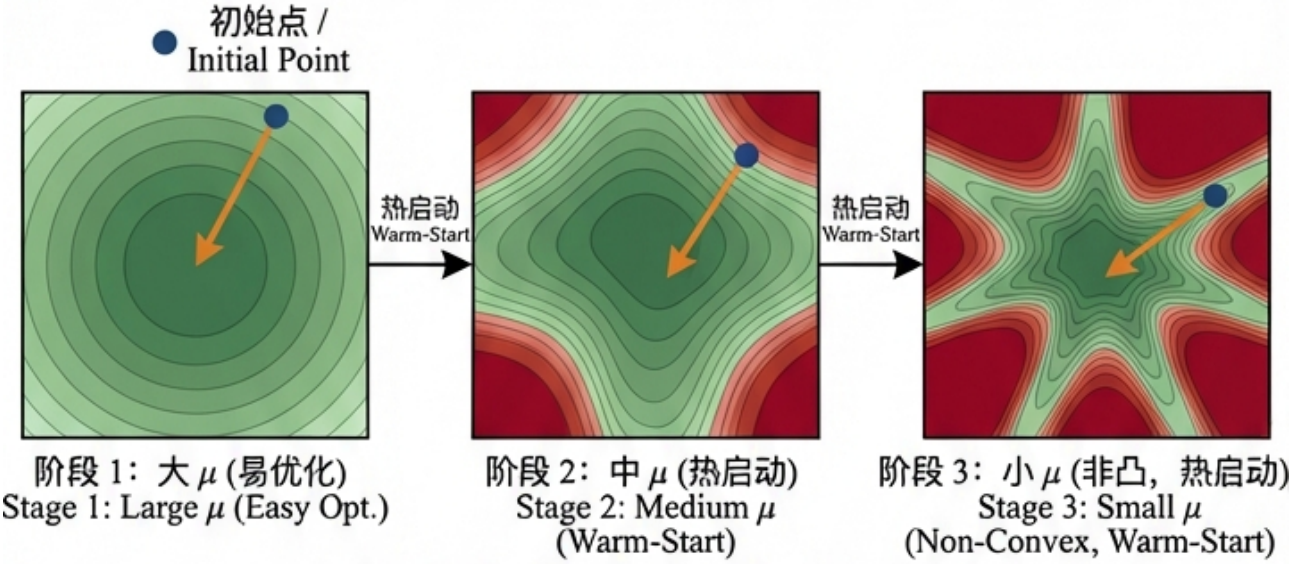
1. **无初始化**：目标函数从 μ 大时的凸初始状态出发，无需初始位姿猜测。
2. **迭代结构规整**：内循环主要是矩阵运算（不更新对应关系），便于高效实现与并行化 [28]。
3. **适合作为粗配准**：在候选对应质量可控时，FGR 常被用作 ICP 前的粗配准模块，为后续局部精修提供可用起点。

FGR 的实验设置给得比较“工程化”：在合成 range 数据上，点云规模约 8,868–19,749 点、重叠率 47%–90%，并添加噪声 $\sigma \in \{0, 0.0025, 0.005\}$；在最高噪声 $\sigma = 0.005$ 的设置下，作者给出的平均 RMSE 为 0.008、最大 RMSE 为 0.017 [28]。在前述 UWA benchmark 上，0.05-recall=84%（最低重叠约 21%）；所有运行时间在 i7-5960X @ 3.0GHz 单线程下统计。

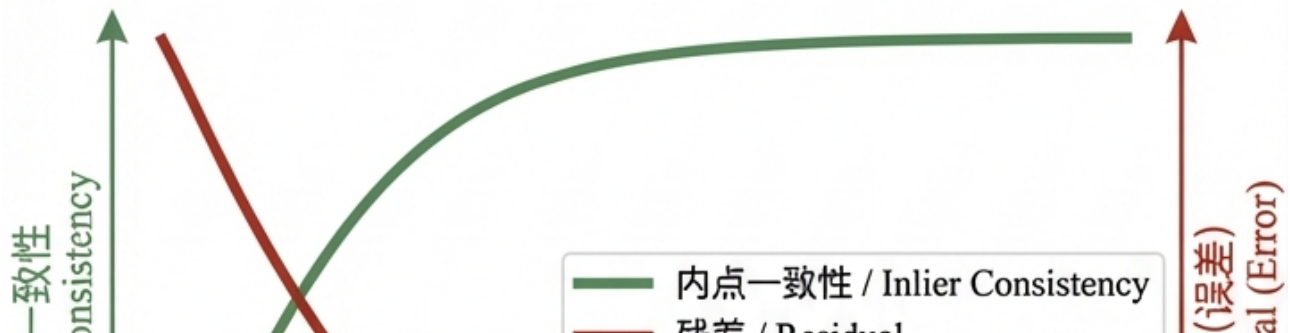
局限性：FGR 对候选对应集的质量敏感——当匹配集合被大量误配主导时，GNC 可能收敛到错误极小值或表现不稳定。



(a) Geman-McClure 损失函数族（随 μ 减小）
Geman-McClure Loss Family (as μ decreases)



(b) 优化景观演变与热启动 (GNC 调度)
Optimization Landscape Evolution & Warm-Start (GNC Schedule)



3.6.5 Go-ICP：分支定界全局最优

Go-ICP [16] 通过分支定界（Branch-and-Bound，BnB）在完整 $SE(3)$ 运动空间上搜索全局最优，理论上保证找到 L_2 意义下的全局最小：

$$T^* = \arg\min_{T\in SE(3)} \sum_i \|Tp_i - q_{NN(Tp_i)}\|^2 \tag{52}$$

关键技术：BnB 将 $SO(3)$ 分解为嵌套超正方体，对每个子块计算旋转误差的下界 $\underline{e}(C)$（基于 Euclidean distance transform 对平移做内层 BnB）。若 $\underline{e}(C) >$ 当前最优值，则剪枝整个子块。子块越小，下界越紧，剪枝越有效。Go-ICP 将局部 ICP 集成到 BnB 中：在每个旋转子块的中心运行局部 ICP 以更新全局上界，加速剪枝 [16]。

Go-ICP 的“代价边界”在论文里有一整组量化实验：以 Stanford bunny/dragon 的 10 个 partial scan 为数据点集、重建模型为目标点集，作者把搜索域设为 $[-\pi,\pi]^3 \times [-0.5, 0.5]^3$，每次随机生成初始位姿做 100 次测试（总计 2,000 次任务），并统一采样 $N = 1000$ 个 data points、收敛阈值设为 $\epsilon = 0.001 \times N$。在这组实验里，Go-ICP（DT 最近邻距离检索）实现了 100% 正确配准；旋转误差均 <2°，平移误差均 <0.01；平均/最长运行时间分别为 bunny 1.6 s / 22.3 s、dragon 1.5 s / 28.9 s；若把 DT 换成 kd-tree，运行时间通常会再长 40–50 倍 [16]。

进一步在“部分重叠 + 修剪”的设置里，作者给出了不同修剪比例 ρ 下的 mean/max time：例如 bunny（$\rho = 10\%$）为 0.81 s / 10.7 s，dragon（A→B 取 $\rho = 20\%$）为 2.99 s / 43.5 s；这组测试中点集对的重叠率在 50%–95% 之间 [16]。这些数字也解释了它的典型定位：精度和全局性强，但不太可能长期跑在高频前端循环里。

它的局限因此也很直接：保证来自系统性的全空间搜索，而全空间搜索本身就意味着时间预算难以下到实时前端的量级。点数再大、搜索域再宽、重叠再低，BnB 的代价都会迅速抬升。

3.6.6 TEASER：可验证的鲁棒全局配准

TEASER（TEASER++） [36] 主打两件事：一是 TLS 让优化对外点“钝化”，二是把“解是否足够好”做成可检查的最优性校验。论文里把对应对下采样到 $N = 1000$（Bunny），把外点率从 95% 扫到 99% 做 Monte Carlo，对比 FGR / RANSAC 等基线；在这组实验里 TEASER / TEASER++ 在 99% 外点下仍保持稳定，而 RANSAC1min 需要 60s 超时预算才能扛到 98% 外点（约 106 次迭代）[36]。

从时延上看，作者在文中直接给出一句结论：TEASER++ 在普通笔记本上可在 <10 ms 内求解“大量外点”的实例；而用于旋转子问题的最优性校验器（DRS）平均需要 24 次迭代把相对次优界压到 <0.1%，并给出“每次迭代约 50 ms（C++）”的实现量级 [36]。

核心技术仍然是三段式：

1. **截断最小二乘（TLS）**：将对应集中的外点视为” 截断点”，定义

$$\mathcal{E}_{\mathrm{TLS}}(T) = \sum_k \min\big(\|Tp_k - q_k\|^2, \bar{c}^2\big) \tag{53}$$

   $\bar{c}$ 为截断阈值，使得误差超过 $\bar{c}$ 的对应（外点）贡献固定量 $\bar{c}^2$ 而非实际误差，不影响最优化。

2. **图论框架解耦**：在对应集的相容性图（compatibility graph）上寻找一致性强的子集（如最大团/超核心等），以批量剔除明显不相容的外点，再进行尺度-旋转-平移的级联估计，降低变量耦合带来的优化难度。

3. **旋转的 SDP 松弛与紧致性检查**：TLS 旋转估计可被松弛为半定规划（SDP），并通过对偶间隙等证据检查该松弛在当前实例上是否紧致（tight）。检查通过时，可以对“解是全局最优/或足够接近全局最优”做事后校验 [36]。

TEASER++ 进一步用更高效的分裂/迭代策略替代直接 SDP 求解，并保留“可验证（certifiable）”的核心特性。[36] 的实验显示其在高外点占优的对应集合上依然具有较强鲁棒性，因此常被视作“实时可用的可认证全局配准”代表方法之一。

其局限同样值得注意：TEASER++ 要求候选对应集中至少包含具备几何一致性的内点子集。若对应图本身已被重复结构或低质量特征彻底破坏，可认证求解亦无法有效工作；此外，该方法通常更适合作为按需触发的粗初始化手段，而非每帧全量运行。

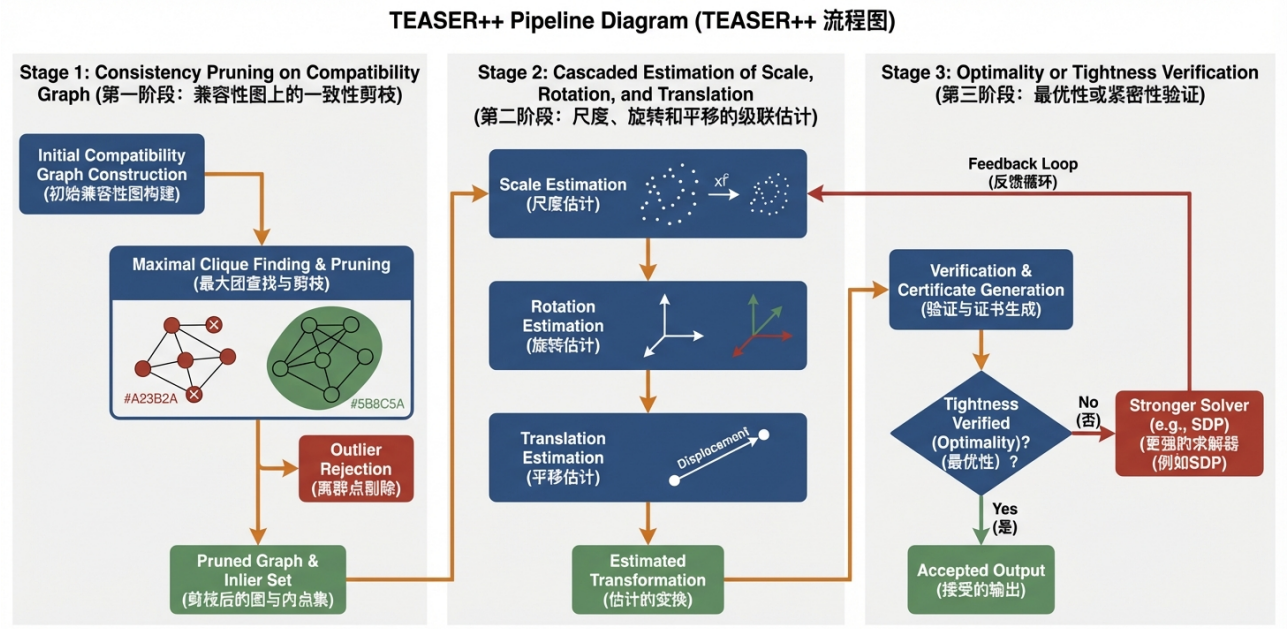


图 28: TEASER++ 的“剪枝 + 级联估计 + 最优性校验”流程示意。(a) 在相容性图上筛除不一致对应，保留几何一致性更强的子集；(b) 将尺度、旋转、平移解耦为级联估计，降低变量耦合；(c) 通过对偶间隙等证据检查当前解对应的松弛是否紧致，从而对解的最优性/次优界做事后校验（示意）。

3.6.7 GeoTransformer：端到端 Transformer 配准

GeoTransformer [57] 代表了一类不依赖手工描述子、也不依赖 RANSAC 的端到端方法。其核心贡献是几何自注意力（Geometric Self-Attention）模块，显式编码点对距离和三元组角度：

$$\mathbf{r}_{ij} = \mathbf{r}_{ij}^{D}\mathbf{W}^{D} + \max_{x}\{\mathbf{r}_{ijx}^{A}\mathbf{W}^{A}\} \tag{54}$$

其中 $\mathbf{r}_{ij}^{D}$ 为点对距离嵌入，$\mathbf{r}_{ijx}^{A}$ 为三元组角度嵌入。这种显式几何编码使得特征对刚体变换更稳定，并在低重叠、重复结构等困难设置下改善内点质量 [57]。

在 3DLoMatch 上，作者报告 Inlier Ratio 提升 17–30 个百分点、Registration Recall 提升超过 7 个百分点；由于对应集合更“干净”，其实现里用确定性的 LGR（Local-to-Global Registration）替代 RANSAC，可带来约 100× 的位姿求解加速 [57]。

这类方法的局限主要在训练域和评测协议绑定得更紧。GeoTransformer 能把对应做得更干净，但前提通常是训练数据里的重叠分布、噪声形态和真实部署场景不要差得太远；一旦域偏移明显，LGR 前提下“对应已经足够好”的假设也会随之变脆。

3.6.8 4PCS 与 Super4PCS：无特征全局配准

4PCS（4-Points Congruent Sets） [58] 用“共面四点基”把搜索空间压到几何全等约束里，本质上仍是 generate-and-test，但因为用的是宽基（wide base），在噪声/外点下更稳。论文把鲁棒性条件写得很直接：在噪声最高到 $\sigma = 4.0$、外点最高到 40%、重叠率降到 30%–40% 的设置下仍能完成配准（以 LCP 思路判定），同时给出了 $O(n^2 + k)$ 的提取复杂度上界 [58]。

Super4PCS [59] 主要是把 4PCS 的“找固定距离点对”做成 smart indexing，复杂度从 $O(n^2)$ 变为线性（$O(n + k)$，并且对输出敏感）。作者报告在保持配准精度的前提下获得约 3–10× 的加速，并展示其在约 25% 重叠、outlier margin 约 20% 的恶劣设置下仍可用 [59]。

它们的局限则在于一旦真实重叠太低、局部几何重复严重，或者噪声把宽基约束本身也破坏了，基于全等集合的筛选效率和稳定性都会下滑。它不依赖描述子是优势，但也意味着更多信息要从几何采样本身里硬挖出来。

3.6.9 方法综合对比

表 15: 第 3.6 节全局配准方法综合对比（定性）：思路、保证类型、鲁棒性侧重点与典型适用场景。

方法	代表论文	思路	全局/可认证保证	鲁棒性侧重点	主要适用场景
FPFH + RANSAC	[9]	描述子匹配 + 采样共识	否（概率）	依赖候选匹配质量	工程易实现、通用粗配准
FPFH + FGR	[28]	描述子匹配 + GNC 鲁棒优化	否（局部）	以鲁棒损失抑制误配	速度与可实现性优先的粗配准
FCGF + RANSAC	[56]	学习型描述子 + 采样共识	否（概率）	在低纹理/低重叠中改善匹配	有训练数据与算力的粗配准
Go-ICP	[16]	BnB 全局搜索	是（全局最优）	在受控噪声/外点下保证性强	精度关键、规模受限的场景
TEASER++	[36]	TLS + 图剪枝 + 可认证验证	是（可认证）	高外点占优下的鲁棒粗配准	需要鲁棒且可验证的全局初始化
GeoTransformer	[57]	端到端学习 + 几何注意力	否	以学习特征提升对应质量	低重叠、重复结构的学习型粗配准
Super4PCS	[59]	无特征的几何一致采样	否	不依赖特征、依赖几何全等	难以稳定提取特征的场景

表 16: 第 3.6 节数据汇总：本节每个引用工作在原文中明确给出的数据集、指标与代表性数值（便于复现与横向比较）。

引用工作	数据集/场景（原文）	指标/阈值（原文）	代表性数值（原文）	关键设置/平台（原文）
FPFH	bunny00 等点云（复杂度分析）	PFH 计算耗时对比	重排序 + 缓存使 PFH 计算时间降低约 75%	以空间连续性重排索引，FIFO 缓存 [9]
FCGF	3DMatch	FMR（$\tau_1 = 0.1$m, $\tau_2 = 0.05$）	0.952 ± 0.029（32 维）	原文表 1：0.019 ms/feature；5 cm 体素约 0.17 s/fragment [56]
FCGF	KITTI	成功：RTE<2 m 且 RRE<5°	20 cm：RTE 4.881 cm，RRE 0.170°，成功率 97.83%	hardest-contrastive；RANSAC 后端 [56]
FGR	UWA benchmark（50 场景、188 对，最低重叠约 21%）	0.05-recall	84%	i7-5960X@3.0GHz 单线程 [28]
FGR	合成 range 数据	RMSE	噪声 $\sigma = 0.005$：平均 0.008，最大 0.017	点数 8,868–19,749；重叠 47%–90% [28]
Go-ICP	Bunny/Dragon（Stanford partial scans）	旋转/平移误差；运行时间	2000 任务：rot<2°、trans<0.01；bunny 1.6 s/22.3 s；dragon 1.5 s/28.9 s	N=1000；域 $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$；DT 检索 [16]
TEASER/TEASER++	Bunny（$N = 1000$）	95%–99% 外点：旋转/平移误差箱线图	99% 外点下仍稳定；<10 ms（TEASER++）	RANSAC1min 需 60 s 才能扛到 98% 外点 [36]
GeoTransformer	3DLoMatch	IR, RR；位姿求解耗时	IR +17–30 个百分点；RR +>7 个点；约 100× 加速	RANSAC-free 的 LGR 后端 [57]
4PCS / Super4PCS	多组噪声/外点/重叠设置	成功配准（LCP）/运行时间	4PCS：σ 到 4.0、外点到 40%、重叠到 30%–40%；Super4PCS：约 3–10× 加速	4PCS 复杂度 $O(n^2 + k)$；Super4PCS 线性/输出敏感 [58] [59]

3.6.10 工程选型建议

在线 SLAM / 激光里程计：优先选择“易实现、稳定、可控开销”的粗配准（如 FPFH+FGR 或轻量 RANSAC）再接局部精修；若具备 GPU 与训练数据，可引入学习型方法提升低重叠与重复结构下的匹配质量。

离线精密重建：可更偏向鲁棒性与可验证性（如 TEASER++）并结合更强的局部模型（如 GICP/概率方法）以获得更稳定的整体质量控制。

工业检测（精度关键）：当需要全局保证时可考虑 Go-ICP 等全局搜索类方法；当更关注吞吐与工程复杂度时，常用“全局粗配准 + 局部精修”的组合取得足够稳定的效果。

资源受限平台：倾向使用手工描述子 + 轻量鲁棒估计 + 多分辨率/早停等策略，在纯 CPU 上获得可控的实时性。

第 3.7 节 将讨论深度学习方法如何以端到端可训练的方式统一特征提取与变换估计，并说明这种做法在低重叠与重复结构场景中的收益和边界。

3.7 深度学习驱动的类 ICP 方法 (Deep Learning-Based Registration Methods)

传统 ICP 在“低重叠 + 外点多 + 结构重复”的场景里最容易失手：最近邻几乎只认欧氏距离，碰上对称结构或遮挡，错误对应会被迭代一步步放大。2019 年以后，点云表征学习与可微模块逐渐成熟，研究者开始把神经网络嵌到配准流水线上：有的只替换某一环节（比如特征或对应），有的干脆把“对应 + 求解”合在一次前向里，甚至直接学“怎么执行 ICP”。

本节按“保留 ICP 结构的程度”由强到弱梳理：先看 PointNetLK 这种把迭代结构保留下来的方法，再看 DCP / RPM-Net 这类用软对应替代硬最近邻的端到端框架，最后落到 GeoTransformer / PointDifformer / NAR-*ICP 这类把一致性、鲁棒性和可解释性一起往前推的路线。

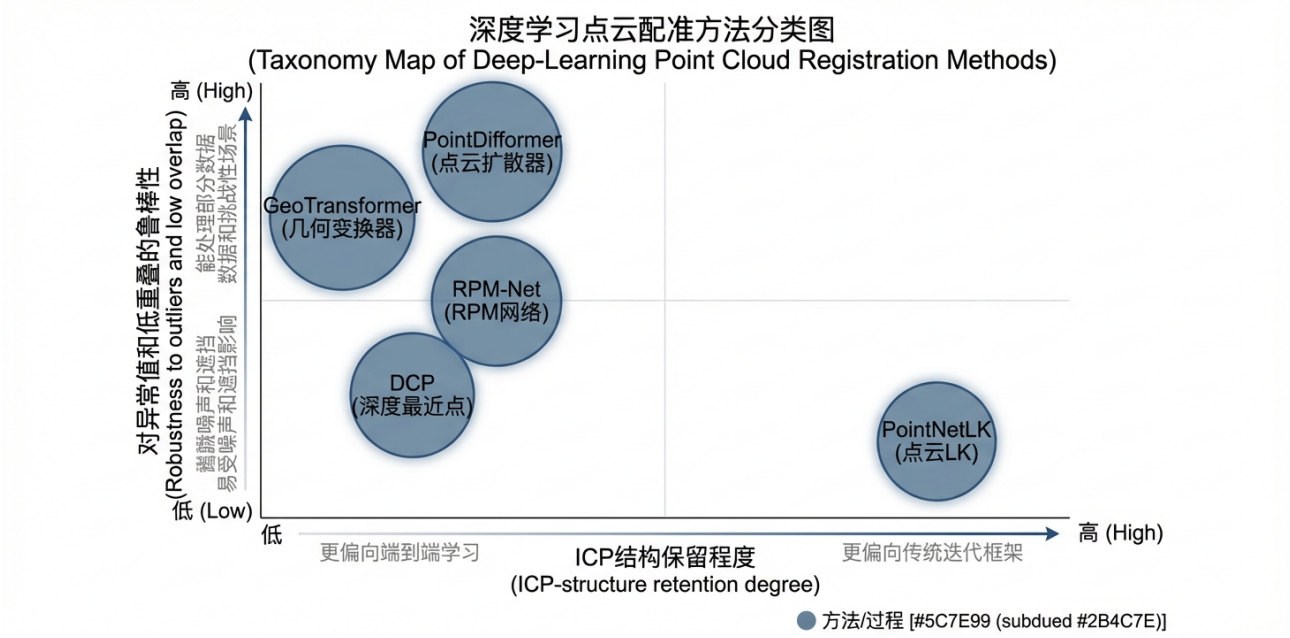


图 29: 深度学习点云配准方法的架构谱系（示意）。横轴：与经典 ICP 框架的保留程度（从最大保留到更端到端）。纵轴：对外点/低重叠的鲁棒性。主要方法从左到右：PointNetLK（ICP 结构 + DL 表征）→ DCP（软对应替代硬对应）→ RPM-Net（Sinkhorn 软对应）→ GeoTransformer（Transformer 特征 + LGR）→ PointDifformer（扩散过程生成对应）。气泡大小仅用于强调代表性方法的相对影响力，不对应固定评测数值。

3.7.1 PointNetLK：将 Lucas-Kanade 迁移到点云

PointNetLK [60] 是将深度学习引入 ICP 框架的先驱工作之一。其核心思想借鉴了 Lucas-Kanade（LK）类迭代：用可微分的表征函数构造 Jacobian，并通过一阶近似迭代估计变换。PointNetLK 将 PointNet 的全局特征向量作为可微分表征函数 [61]，将 LK 迭代展开为递归深度神经网络：

$$\Delta\boldsymbol{\xi}^{(k)} = J_{\Phi}^{+}\left(\Phi(\mathcal{Q}) - \Phi\left(T^{(k)}(\mathcal{P})\right)\right) \tag{55}$$

其中 Φ 为 PointNet 特征提取函数，J_{Φ}^{+} 为其伪逆 Jacobian（在训练中学习），$\Delta\boldsymbol{\xi}^{(k)} \in se(3)$ 为李代数更新量。PointNetLK 将该迭代展开为固定深度的网络并端到端训练，使位姿误差可通过反向传播直接监督表征学习。

PointNetLK 的关键点在于：它不再在点空间里做最近邻，而是让“变换前后的全局特征”对齐。这里用到的 PointNet 表征并不是凭空而来，原始 PointNet 在 ModelNet40 分类上给出的整体准确率为 89.2%（原文表 1）[61]，PointNetLK 直接复用这种全局几何编码作为 $\Phi(\cdot)$，把“配准”变成“特征差的迭代逼近”。

PointNetLK 的实验比较集中在“初值有扰动、但还没完全错开”的范围内：在 ModelNet40 上，测试时的初始平移取 $[0, 0.3]$，初始旋转取 $[0, 90]^{\circ}$，并且 PointNetLK 与 ICP 都固定迭代 10 次 [60]。更直观的一组结果来自 Stanford bunny：ICP 的旋转/平移误差为 $(175.51^{\circ}, 0.22)$，Go-ICP 为 $(0.18^{\circ}, 10^{-3})$，PointNetLK 为 $(0.2^{\circ}, 10^{-4})$；耗时上，ICP 约 0.36 s，PointNetLK 约 0.2 s，而 Go-ICP 约 80.78 s [60]。这说明 PointNetLK 在中等初值误

差下能比传统 ICP 更快收敛到正确解，但当初值已经落到错误吸引域时，它和 ICP 一样没有把局部配准改成全局配准。

它的问题也很明确：全局特征会把局部几何平均掉。当输入只剩局部可见区域，或者物体由细碎结构构成时，特征差对位姿的梯度会变弱，LK 更新会提前停在残差仍然偏大的位置 [60]。

3.7.2 DCP：软对应与 Transformer 注意力

DCP（Deep Closest Point） [18] 以三个创新替代了经典 ICP 的最近邻硬对应：（1）以 DGCNN 提取逐点局部特征；（2）以 Transformer 注意力机制计算源点云与目标点云之间的软对应权重；（3）以加权 SVD 估计变换。

Transformer 软对应：对于源点 p_i，计算其与目标点云所有点的对应权重：

$$a_{ij} = \mathrm{softmax}\left(\frac{f_i^P \cdot f_j^Q}{\sqrt{d}}\right), \quad \tilde{q}_i = \sum_j a_{ij} q_j \tag{56}$$

其中 f_i^P, f_j^Q 为经过 Transformer 交叉注意力的特征向量，d 为特征维度。软对应 $\tilde{q}_i$ 是所有目标点的加权平均，可微分地传递到 Kabsch/SVD 变换估计步骤。整个流程端到端训练，监督信号为变换参数的均方误差。

与经典 ICP 的关键区别：DCP 的对应不需要也不利用当前位姿估计——特征匹配直接在原始坐标空间中进行，不是点变换后的最近邻。这意味着 DCP 在一次前向传播中完成所有迭代，而不是真正的” 迭代”，从根本上改变了 ICP 的算法结构 [18]。

这篇工作用的是 ModelNet40 上最常见的一套合成配准设置：三轴旋转均匀采样于 $[0, 45]°$，平移采样于 $[-0.5, 0.5]$，训练集和测试集按“类别不重叠”拆开 [18]。在这组设置下，DCP-v2 的 RMSE(R) 为 1.1434°、MAE(R) 为 0.7706°，RMSE(t) 为 0.001786、MAE(t) 为 0.001195；作为参照，同样条件下 PointNetLK 的 RMSE(R) 为 15.0954°，FGR 为 9.3628°[18]。推理速度方面，在 i7-7700 + GTX 1070 上，512 点输入时 DCP-v2 约 0.0079 s，1024 点时约 0.0083 s（原文表 4）[18]。

DCP 的前提也比经典 ICP 更强。它要求训练得到的特征空间能把真对应和假对应分开；测试分布一旦偏离训练分布，或者局部几何存在大面积重复，注意力矩阵就会把错误对应整体抬高。由于 DCP 只做一次前向传播，前面这一步一旦偏掉，后面的 SVD 会直接给出带偏变换，没有经典 ICP 那种逐轮修正的机会 [18]。

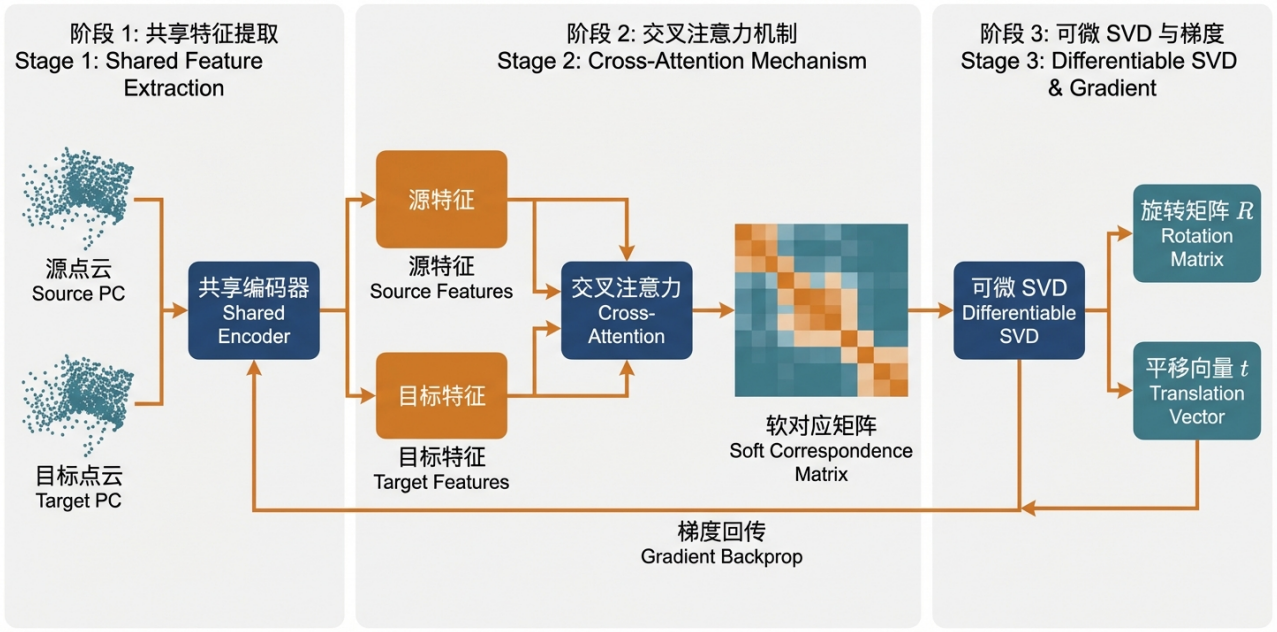


图 30: DCP（Deep Closest Point）端到端网络架构示意，展示从输入点云到变换估计的三段式数据流：（a）DGCNN 提取逐点特征；（b）Transformer 交叉注意力形成软对应（对应矩阵为连续权重而非硬匹配）；（c）可微分 SVD（Kabsch）从软对应中回归刚体变换，并将梯度回传到特征提取模块，实现端到端训练。

3.7.3 DeepVCP：显式关键点检测与对应

DeepVCP [62] 明确将配准分解为两个子问题：可重复关键点检测（repeatable keypoint detection）和可微分对应估计（differentiable correspondence estimation）。关键点检测器以点特征权重对原始点云加权聚合为稳定三维关键点，对应估计网络预测每对候选关键点的匹配概率，最终以全连接网络直接回归变换参数：

$$(R, t) = \mathrm{FCNet}(\{(k_i, c_j, s_{ij})\}_{i,j}) \tag{57}$$

其中 k_i 为源关键点，c_j 为目标关键点，s_{ij} 为匹配分数。论文把“关键点检测要避开动态物体”作为核心卖点，并用端到端结构把检测器也一起拉进训练里 [62]。

从结果看，DeepVCP 在真实车载数据上的精度是能站住的：在 KITTI 上，“Ours-Duplication”的旋转误差均值/最大值为 0.164°/1.212°，平移误差均值/最大值为 0.071 m/0.482 m（原文表 1）；在 Apollo-SouthBay 上，对应数值分别为 0.056°/0.875° 和 0.018 m/0.932 m（原文表 2）[62]。但它的代价也写得很清楚，端到端推理仍在秒级，柱状图读出来大约是 2 s[62]。

DeepVCP 的重点在于把关键点和对应一起学出来，而不是把每帧推理压到里程计前端能接受的时延。关键点若在遮挡、稀疏采样或跨域场景下失去重复性，后面的匹配分数和位姿回归会一起失真；再加上单次推理约 2 s，它不能直接充当实时前端 [62]。

3.7.4 RPM-Net：Sinkhorn 归一化与退火对应

RPM-Net（Robust Point Matching Network） [19] 将经典点集匹配中的 Softassign/Sinkhorn 归一化引入深度学习框架，解决 DCP 中 softmax 对应矩阵的两个问题：（1）行归一化但不列归一化，导致多对一匹配；（2）对位姿的全局变化不鲁棒。

Sinkhorn 层：对以混合特征（位置 + 法向）计算的相似度矩阵 M，应用多轮 Sinkhorn 迭代确保行列同时归一化：

$$S^{(0)} = \exp(M/T_{\mathrm{anneal}}), \quad S^{(l+1)} = \mathrm{ColNorm}\Big(\mathrm{RowNorm}\Big(S^{(l)}\Big)\Big) \tag{58}$$

其中 T_{anneal} 为温度参数。**退火策略**：从高温逐步降到低温，让对应矩阵从“分得开但不尖锐”过渡到“更接近一一匹配”[19]。RPM-Net 还显式加了“dustbin”槽位来吸收无对应的源点，从结构上把“外点”这件事写进匹配矩阵。

RPM-Net 的实验仍然放在 ModelNet40 上做，但比 DCP 多推了两步：一是把模型统一采样到 2048 点并归一化到单位球，二是专门把噪声和部分可见性单独拎出来看 [19]。在干净数据上，RPM-Net 的各向同性误差为 0.056°（旋转）与 0.0003（平移），明显优于 PointNetLK 的 0.847° 与 0.0054（原文表 1）；加入高斯噪声 $\mathcal{N}(0, 0.01)$ 并裁剪到 $[-0.05, 0.05]$ 后，DCP-v2 的各向同性误差为 2.426° 与 0.0141，而 RPM-Net 能压到 0.664° 与 0.0062（原文表 2）[19]。再往前走一步，在部分可见实验里，作者用随机半空间保留约 70% 的点并下采样到 717 点，PointNetLK 还要再加一个 $\tau = 0.02$ 的可见性筛选；在“部分可见 + 噪声”下，DCP-v2 的各向同性误差为 2.994° 与 0.0202，而 RPM-Net 为 1.712° 与 0.018（原文表 3）[19]。

这种改进不是没有代价。作者在 3.0 GHz i7-6950X + Titan RTX 上按 5 次迭代统计每对点云的平均推理时间：512 点时 RPM-Net 为 25 ms，而 DCP-v2 只有 5 ms；1024 点时分别为 52 ms 与 9 ms；2048 点时进一步拉开到 178 ms 与 21 ms（原文表 5）[19]。也就是说，Sinkhorn 双随机归一化和退火策略确实把匹配矩阵做得更规整，但计算账也更重，点数一上去增长得很快。

3.7.5 GeoTransformer：几何自注意力与 RANSAC-free 求解

GeoTransformer [57] 是将“可学习特征”与“基于一致性的全局估计”紧密耦合的代表性方法。它的目标并不是把 ICP 的每一步都替换为黑盒，而是让网络输出的候选对应在几何上更一致，从而减少对随机采样与硬阈值的依赖。其三个核心机制：

1. 几何自注意力（Geometric Self-Attention）：在 Transformer 注意力层中显式编码点对的距离和三元组的角度关系：

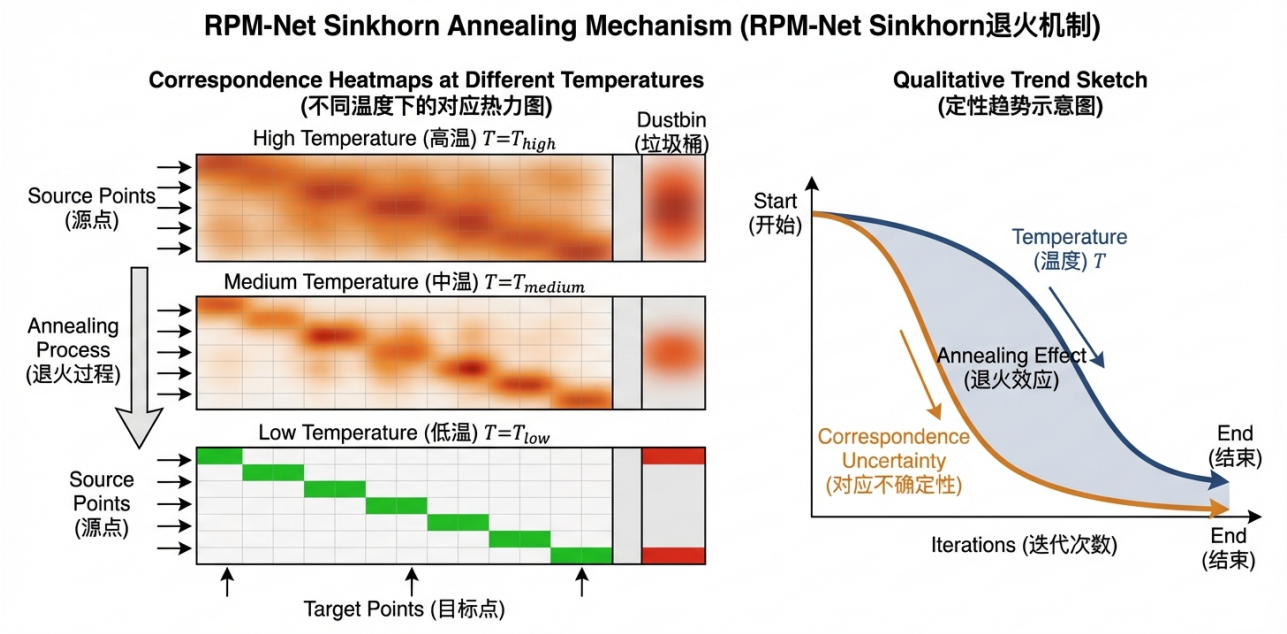


图 31: RPM-Net 中 Sinkhorn 归一化与温度退火的对应矩阵演化示意。高温阶段对应矩阵更“软”（分布更均匀），随着温度降低与 Sinkhorn 迭代，矩阵逐步变尖锐并趋向一一匹配；外点可被分配到“外点槽（dustbin）”。右侧以示意曲线展示温度下降与对应不确定性（如熵/分布宽度）降低的同步趋势。

$$e_{ij} = \frac{(f_i + r_{ij}^D)(f_j + r_{ij}^D)^\top}{\sqrt{d}} + \frac{\max_x r_{ijx}^A \cdot f_i}{\sqrt{d}} \tag{59}$$

其中 r_{ij}^D 为点对距离的嵌入，r_{ijx}^A 为三元组角度嵌入。这一设计使得注意力权重对刚体变换保持严格不变性，而不依赖于外部坐标系，大大提升了低重叠场景的内点比 [57]。

2. 超点到密点的层次化匹配：先在下采样的超点（superpoint）级别做全局匹配，再将超点对应通过局部传播回到稠密点，以兼顾计算可行性与几何细节。

3. LGR（Local-to-Global Registration）：在对应质量足够高时，可用局部-全局一致性求解替代随机采样估计（RANSAC）来求解刚体变换，使求解过程更确定、更易并行，并减少“采样失败”对结果的影响 [57]。

GeoTransformer 的说服力主要来自低重叠场景。作者在 3DMatch 和 3DLoMatch 上分别取重叠率大于 30% 和 10%–30% 的点云对，并保留了文献里常见的 50K 次 RANSAC 统计口径 [57]。在 RANSAC-50k、采样 5000 对对应的条件下, GeoTransformer 的 RR(%) 为 92.0 / 75.0 (3DMatch / 3DLoMatch), 总耗时约 1.633 s (原文表 1); 同一条件下, 采用粗到细对应传播的 CoFiNet [63] 为 89.3 / 67.5, 专门针对低重叠区域建模 overlap-attention 的 PREDATOR [64] 为 89.0 / 59.8。真正关键的是把 RANSAC 拿掉以后: 若直接用加权 SVD, GeoTransformer 的 RR 为 86.5 / 59.9，总耗时约 0.078 s；换成 LGR 后 RR 提到 91.5 / 74.0，总耗时约 0.088 s（原文表 2）。如果只看求解阶段，LGR 相比它自己的 RANSAC-50k 版本，从 1.633 s 降到 0.088 s，约快 18.6 倍；和 CoFiNet 的 LGR（0.143 s）相比也快约 1.6 倍（原文表 2）[57]。

GeoTransformer 能把 RANSAC 拿掉, 前提是前端给出的超点对应已经足够一致。若场景跨域明显、重叠率继续下降，或者局部几何存在重复结构，LGR 接收到的对应集就会失去一致性，此时求解阶段同样会失败 [57]。

3.7.6 PointDifformer：神经扩散与对应生成

PointDifformer [65] 将神经网络偏微分方程（Graph PDE）和热核签名（Heat Kernel Signature，HKS）引入点云配准，以扩散过程增强特征表示的鲁棒性。

热扩散特征提取：通过图神经 PDE 模块在点云图上传播特征，等价于在点云上求解热方程 $\partial_t F = \Delta_{\mathcal{G}} F$（$\Delta_{\mathcal{G}}$ 为图拉普拉斯算子）。热核签名描述了在时间 t 从点 p_i 出发的热量仍然留在 p_i 的概率，对等距变换严格不变——即便形状发生非刚体小变形，签名变化也有限，使 PointDifformer 对高斯噪声和三维形状扰动的鲁棒性更强 [65]。

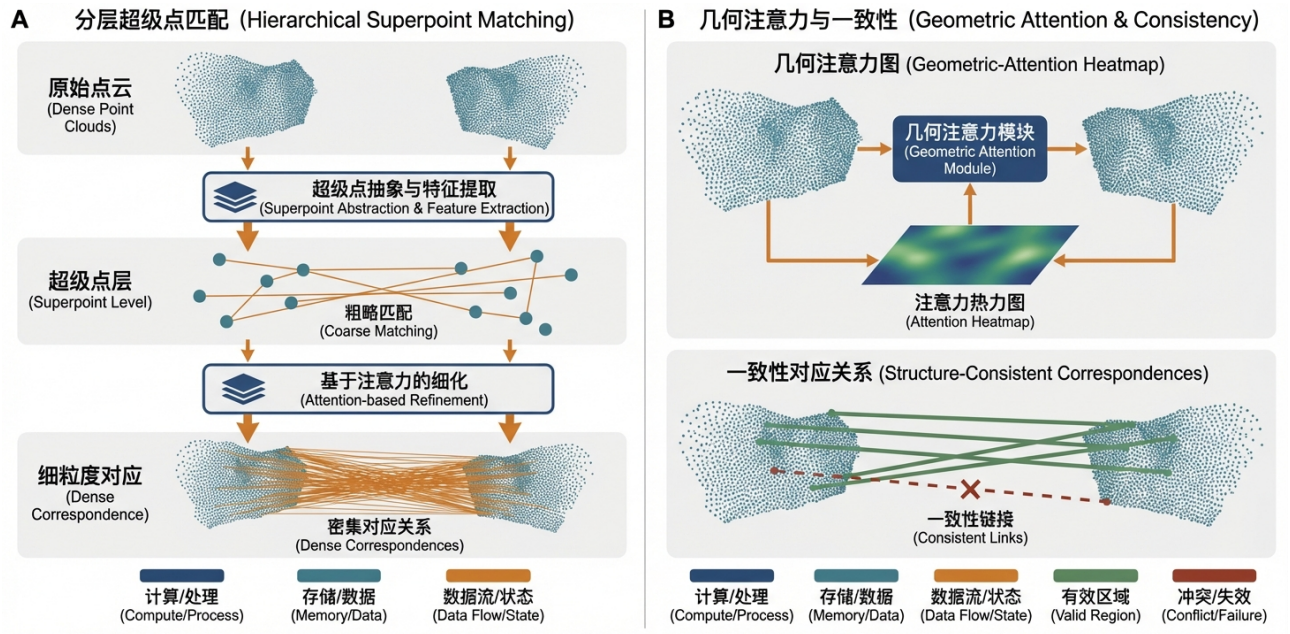


图 32: GeoTransformer 的关键直观：层次化超点匹配把“全局搜索”放在低分辨率表示上完成，再把对应传播回稠密点；几何自注意力热力图展示注意力权重更倾向于几何一致的结构对应，而非仅由欧氏距离决定。该图为机制示意，非特定数据集的定量复现。

可学习 SVD：最终变换估计通过带可学习权重的 SVD 模块完成——对应矩阵中每对点的权重由网络预测，而不是固定为内点/外点的二元分类。变换估计损失为

$$\mathcal{L}_{\text{mse}} = \frac{1}{K'} \sum_{i=1}^{K'} \|\hat{R}x_i + \hat{t} - y_i\|_2^2 \tag{60}$$

PointDifformer 没有只在合成物体上报数，而是把重点放回真实扫描。3DMatch / 3DLoMatch 上，它的 RR 分别为 93.0% / 75.2%；换到 KITTI，常规测试下的平移 MAE/RMSE 为 4.14 cm / 8.86 cm，旋转 MAE/RMSE 为 0.14° / 0.23°，RR 为 97.7%[65]。为了说明扩散特征对噪声和扰动更稳，作者又专门在 KITTI 上加了两组破坏：叠加高斯噪声（例如 $\mathcal{N}(0, 0.25)$）时，PointDifformer 的平移 RMSE 为 9.00 cm，而 GeoTransformer 为 14.43 cm（原文表 VI）；局部移除点形成形状扰动时，前者为 8.99 cm，后者为 13.08 cm（原文表 VIII）[65]。与此同时，工程代价也没有藏着：推理时间约 0.072 s，GPU 显存占用约 2.44 GB（原文表 XI）[65]。

这条路线的代价也很直接。扩散式特征提取、图上的 PDE 演化和后面的加权求解串起来以后，推理时间为 0.072 s，显存占用为 2.44 GB，训练和部署开销都高于 DCP、RPM-Net；因此它不能直接替代高频在线前端[65]。

3.7.7 NAR-*ICP：神经算法推理与可解释配准

NAR-*ICP [66] 基于神经算法推理（Neural Algorithmic Reasoning，NAR）范式，以图神经网络（GNN）学习” 执行” 经典 ICP 的每一步中间计算。与 DCP/GeoTransformer 将 ICP 替换为黑盒不同，NAR-*ICP 保留 ICP 的迭代结构作为归纳偏置（inductive bias），GNN 只学习每一步最近邻选择、权重计算和变换估计中的参数：

$$\boldsymbol{\xi}_\theta^{(k+1)} = \text{GNN}_\theta\left(\mathcal{P},\, \mathcal{Q},\, \boldsymbol{\xi}^{(k)}\right) \tag{61}$$

训练监督作用于每步中间状态，而非仅约束最终位姿，这赋予模型真正意义上的可解释性：每步输出均对应 ICP 的中间量（对应关系、相位估计、终止判断），可直接用于分析与诊断。

NAR-*ICP 的实验思路和前面几篇不太一样，它不是只看最后有没有对齐上，而是看“学出来的执行过程”能不能在不同扫描条件下稳住。作者从 SemanticKITTI 中抽取带语义标签的物体中心点，按 KITTI 常见的

RTE/RRE 指标评估 [66]。在合成数据和三档扫描间距（平均 1.6 m / 11.3 m / 24 m）的对比里，24 m 这一档最能看出差异：P2P-ICP 的 RTE/RRE 为 0.934 / 1.912，而 NAR-P2L 为 0.391 / 0.796；再加上文中提出的 ground-truth optimisation 后，NAR-GICP+ 可进一步降到 0.222 / 0.458（原文表 II、表 III）[66]。横向和学习型基线比较时，作者把 RR 定义为“RTE < 2 m 且 RRE $< 0.6°$ 的成功率”，并同时报告 RTEGT/RREGT/RRGT：GeoTransformer 为 0.335/0.512/85.2%，Predator 为 0.433/0.371/74.1%，DCP 为 0.147/0.376/99.4%，NAR-P2Pv2+ 为 0.148/0.334/98.2%（原文表 V）[66]。效率方面，在同一 GPU 上，GeoTransformer 平均约 0.13 s/对，Predator 约 0.34 s/对，DCP 约 0.03 s/对，而 NAR-P2Pv2 为 0.02 s/对；参数量约 773k（原文表 VI）[66]。

NAR-*ICP 更依赖训练时给出的中间监督。若训练阶段只覆盖少量扫描间距、少量几何类型或固定的执行顺序，网络学到的就是那一套 ICP 过程；测试时一旦超出这组条件，中间步骤就会先失真，最终位姿也会跟着偏掉 [66]。

3.7.8 深度学习方法与经典 ICP 的系统对比

表 17: 第 3.7 节深度学习点云配准方法综合对比（定性）：学习对象、变换估计方式、优势与代价，以及部署友好度。

方法	学到的对象	变换估计方式	优势（相对经典）	主要风险/代价	部署友好度（定性）
P2P ICP [1]	无（几何规则）	硬对应 + 闭式解	可解释、可控、易验证	对低重叠/外点敏感	极高
PointNetLK [60]	全局表征 + 可微 Jacobian	LK 式迭代（展开）	在特征空间缓解噪声/密度扰动	全局特征易丢局部细节，收敛域受限	中
DCP [18]	点特征 + 软对应	加权 SVD	端到端、对应可微、避免硬最近邻	注意力代价高，跨域需谨慎	低
RPM-Net [19]	双随机软匹配（含外点槽）	加权 SVD（迭代/退火）	软对应更稳定、更接近一一匹配	仍依赖训练分布与算力	低
GeoTransformer [57]	几何一致性特征 + 层次匹配	LGR（局部-全局一致性求解）	低重叠下更易获得一致对应，减少采样不确定性	对训练数据与评测设置较敏感，工程落地需监控失败模式	中
PointDifformer [65]	扩散式鲁棒表征/对应权重	可学习 SVD	对噪声与扰动更稳健	训练复杂、推理开销偏高	低
NAR-*ICP [66]	“如何执行 ICP”	GNN 预测每步中间量	可解释、对结构外推更友好	仍需精心设计监督与泛化评测	中

表 18: 第 3.7 节关键学习型方法的“数据集-指标-数值”摘录汇总（每条均对应原论文的表格/图示设置）。

方法	数据集与设置	指标与数值（摘录）	运行时/硬件（摘录）
PointNet [61]	ModelNet40 分类	overall accuracy 89.2%（原文表 1）	-
PointNetLK [60]	Stanford bunny；对比 ICP/Go-ICP	ICP (175.51°, 0.22)，Go-ICP (0.18°, 1e-3)，PointNetLK (0.2°, 1e-4)	ICP 0.36 s；PointNetLK 0.2 s；
DCP [18]	ModelNet40（类别不重叠；旋转 [0,45]°；平移 [-0.5,0.5]）	DCP-v2: RMSE(R)=1.1434°，MAE(R)=0.7706°；RMSE(t)=0.001786，MAE(t)=0.001195（原文表 1）	512 点：0.007932 s；1024 点： s（原文表 4；i7-7700+GTX1070）
DeepVCP [62]	KITTI / Apollo-SouthBay	KITTI: rot mean/max 0.164°/1.212°，trans mean/max 0.071 m/0.482 m；Apollo: rot mean/max 0.056°/0.875°，trans mean/max 0.018 m/0.932 m（原文表 1/2）	端到端推理约 2 s（原文图 3）
RPM-Net [19]	ModelNet40（类别不重叠；噪声 $\mathcal{N}(0, 0.01)$ 裁剪到 $[-0.05, 0.05]$；部分可见约保留 70% 并下采样 717 点）	噪声下：isotropic err 0.664°/0.0062（rot/trans，原文表 2）；部分可见 + 噪声：1.712°/0.018（原文表 3）	512/1024/2048 点：25/52/178 ms（原文表 5；i7-6950X+Titan RTX；5 iter）
GeoTransformer [57]	3DMatch（重叠 >30%）/3DLoMatch（10%–30%）；RANSAC-50k	RANSAC-50k: RR 92.0/75.0，总 1.633 s（原文表 1）；LGR: RR 91.5/74.0，总 0.088 s（原文表 2）	见左（Model/Pose/Total）
PointDifformer [65]	3DMatch/3DLoMatch；KITTI；噪声/扰动鲁棒性	RR: 93.0%/75.2%（3DMatch/3DLoMatch，原文表 X）；KITTI: RR 97.7%，trans RMSE 8.86 cm，rot RMSE 0.23°（原文表 III）	推理 0.072 s；显存 2.44 GB（
NAR-*ICP [66]	SemanticKITTI 派生数据；KITTI 风格 RTE/RRE；RR: RTE<2 m 且 RRE<0.6°	与学习基线：NAR-P2Pv2+ 的 RTEGT/RREGT/RRGT 为 0.148/0.334/98.2%（原文表 V）	推理 0.02 s；参数量约 773k（原文表 VI）

3.7.9 训练范式与迁移学习

深度学习配准方法的训练面临三个核心挑战：

1. 监督信号设计：早期方法（DCP、RPM-Net）以位姿真值监督（regression loss），需要精确的地面真值变换。GeoTransformer 以对应内点比（IR）作为辅助监督，更直接地引导特征学习。NAR-*ICP 以每一步 ICP 的中间状态监督，提供最细粒度的信号。

2. 域偏移（Domain Shift）：合成数据与真实扫描在噪声模型、密度分布、遮挡与采样机制上差异显著，导致“在合成上训练的模型”跨域到真实数据时性能下降。域适配（自训练、对抗训练等）可缓解这一问题，但难以彻底消除。

3. 零样本泛化：基础大模型（如 Point-MAE、PointBERT）的出现使得预训练-微调范式开始应用于点云配准。以自监督大模型特征替代从头训练的 FCGF/DGCNN，可在新场景上零样本部署，无需任何标注数据，是当前研究的热点方向。

3.7.10 深度学习 vs 经典 ICP：互补而非替代

深度学习方法并非全面取代经典 ICP，二者形成明确的互补关系：

- **经典 ICP（见第 3.1 节 至第 3.4 节 ）**在以下场景仍不可替代：对确定性行为与可解释误差传播要求高、算力/功耗受限（无 GPU 或仅轻量算力）、需要可验证的稳定性与工程可控性的安全关键系统、以及以低延迟为硬约束的实时闭环。
- **深度学习方法**在以下场景更容易体现收益：重叠不足或遮挡严重、外点与重复结构多、需要跨域鲁棒性（传感器/采样机制变化）、以及对语义一致性有需求的应用。

更稳妥的工程组合：由深度学习模块提供粗初始位姿或过滤后的高质量候选对应，再由经典 ICP（GICP 或 FRICP）完成可控的几何精修，并以传统几何/统计检验作为兜底。第 4 章和第 5 章将分别从软件和硬件两个维度讨论如何加速经典 ICP，以缩小其与深度学习方法在推理时延上的差距。

3.8 优化求解器视角

SVD 是 ICP 位姿求解的起点，却不是终点。点到平面残差的非线性结构、鲁棒 M-估计器的加权迭代、稀疏范数的变量分裂，以及因子图的增量平滑，均要求超越闭合解的迭代优化框架。本节以统一的最小二乘视角梳理这些求解器，并揭示它们与第 3.2 节 （异常值处理）、第 3.4 节 （变换估计）和第 3.5 节 （退化感知）各节的内在联系。

3.8.1 最小二乘统一框架

ICP 的位姿求解可统一表述为加权非线性最小二乘（NLS）问题。设源点 p_i、目标点 q_i，变换 $T=(R,t)\in SE(3)$，残差 $e_i(T)$ 随度量类型而异：

$$\min_{T\in SE(3)}\sum_{i=1}^{n}\rho\big(\|e_i(T)\|^2\big) \tag{62}$$

三种主流残差的具体形式为：P2P 残差 $e_i=Rp_i+t-q_i$；P2Pl 残差 $e_i=n_i^\top(Rp_i+t-q_i)$（n_i 为目标点法向量）；Symmetric-ICP 残差 $e_i=(n_i+m_i)^\top(Rp_i+t-q_i)/2$，其中 m_i 为源点法向量。

当 $\rho(s)=s$（无鲁棒核）时，对 T 在当前估计 T_0 处做左扰动 $T'=\exp(\hat{\xi})\cdot T_0$（$\xi\in\mathfrak{se}(3)$），一阶 Taylor 展开给出线性化残差 $e_i(T')\approx e_i(T_0)+J_i\xi$，其中 $J_i\in\mathbb{R}^{k\times 6}$ 为 Jacobian。令梯度为零得正规方程：

$$\underbrace{J^\top W J}_{H}\,\xi^*=-J^\top W e \tag{63}$$

其中 $W=\mathrm{diag}(w_1,\dots,w_n)$ 为权重矩阵，J 和 e 是各残差的堆叠。P2P 情形下 H 具有特殊的解析结构，可用 SVD 精确求解，这正是第 3.4 节 Kabsch 算法的来源，也是式 63在 $W=I$、$k=3$ 时的特例 [67]。P2Pl 情形下 $H\in\mathbb{R}^{6\times 6}$，无闭合解，须迭代求解。

3.8.2 Gauss-Newton 与 Levenberg-Marquardt

Gauss-Newton（GN）用 $H = J^\top W J$ 近似 Hessian，解线性系统得到增量 $\delta\xi$：

$$H\,\delta\xi = -J^\top W e.$$

对 ICP 而言，关键不是”会不会解线性系统”，而是 **Jacobian 到底长什么样**。以点到面残差为例，

$$r_i(T) = n_i^\top (R p_i + t - q_i)$$

（n_i 为目标点法向），对当前估计 (R,t) 做小扰动 $(\delta\theta, \delta t)$，有一阶近似：

$$r_i(T') \approx r_i(T) + \underbrace{\begin{bmatrix}(Rp_i \times n_i)^\top & n_i^\top\end{bmatrix}}_{J_i}\begin{bmatrix}\delta\theta \\ \delta t\end{bmatrix}.$$

于是每个对应点对都只往一个 6×6 的 $H = \sum_i w_i J_i^\top J_i$ 和 $g = \sum_i w_i J_i^\top r_i$ 里“加一笔”，最后解 $H\,\delta\xi = -g$。这里真正的开销在于遍历点对装配（$O(n)$），而不是求解本身（Cholesky 分解只是常数规模）。GN 在接近最优解时收敛快，但它对初值的依赖很强：一旦线性化偏离真实残差形状（重叠不足、外点占比高、退化方向明显），就容易走偏甚至发散。

Levenberg-Marquardt（LM）引入阻尼项将正规方程改为：

$$(J^\top W J + \lambda I)\,\delta\xi = -J^\top W e \tag{64}$$

$\lambda > 0$ 时可把它理解为“带正则的 GN”，或者更直白一点：**先把一步走得过大的风险压下去**。λ 大时更像梯度下降（收敛盆更大，但步子小），$\lambda \to 0$ 时恢复 GN 的二次收敛速率。工程上 LM 的价值主要体现在两点：一是当 H 条件数很差、甚至接近奇异（走廊、平面等退化结构）时，λI 往往能把系统“扶正”，避免数值崩掉；二是配合信赖域策略（根据“这一步带来多大真实下降”来调 λ），能在“快”和“稳”之间自动切换。

KISS-ICP 的位姿求解核心就是 LM：它用 λ 和数据关联阈值一起自适应调节更新强度，把前端的抖动压到能稳定跑实时的范围里 [68]。论文把系统参数写得很简洁，全系统只有 7 个自由参数（原文表 I），其中包括初始关联阈值 $\tau_0 = 2\,\mathrm{m}$、最小位移阈值 $\delta_{min} = 0.1\,\mathrm{m}$、每体素点数上限 $N_{max} = 20$、ICP 收敛阈值 $\gamma = 10^{-4}$；在 KITTI-raw 上，含 deskewing 的运行频率约为 38 Hz[68]。这里能看出的事实是：LM 用阻尼项控制更新步长，所以在 Hessian 条件数差、对应抖动或初值偏差中等时，迭代还能继续走下去；但它不能扩大收敛域。初值一旦落入错误盆地，或者对应关系已经系统性出错，LM 只会在错误方向上收敛。

twidthhtwidth

图 33: （a）P2P 与 P2Pl 目标函数在切空间的损失曲面示意：P2P 往往“更圆”（条件更好），P2Pl 在某些结构场景下更“狭长”（更病态）；（b）GN 与不同阻尼强度的 LM 从同一初始点出发的迭代轨迹：弱阻尼更接近 GN（收敛快但更依赖初值），强阻尼更接近梯度下降（更稳但更慢）；（c）不同度量构造的 Hessian 条件性对数值求解难度的影响示意（定性）。

3.8.3 IRLS 与 M-估计器

当 $\rho(\cdot)$ 为鲁棒函数（如 Huber、Cauchy、Geman-McClure）时，目标函数不再是标准 LS，无法直接套用 GN。**迭代加权最小二乘（IRLS）**将非 LS 问题转化为一系列加权 LS 子问题：固定当前估计 T_k，计算权重：

$$w_i = \frac{\rho'(\|e_i(T_k)\|)}{2\|e_i(T_k)\|} \tag{65}$$

然后以 $W_k = \mathrm{diag}(w_i)$ 解加权正规方程得 T_{k+1}，再重新计算权重，循环至收敛。三种常用核的权函数：Huber 核 $w_i = \min(1, \delta/|e_i|)$；Cauchy 核 $w_i = 1/(1+e_i^2/c^2)$；Geman-McClure 核 $w_i = \sigma^2/(\sigma^2+e_i^2)^2$。

IRLS 的收敛性由 **half-quadratic(HQ)松弛**保证:引入辅助变量 z_i,将原问题等价改写为 $\min_{T,z}\sum_i[z_i\|e_i\|^2/2 + \psi(z_i)]$，其中 $\psi(\cdot)$ 为对偶势函数。交替最小化 z_i（闭合解：$z_i^* = h(e_i)$）和 T（加权 LS）保证单调下降。第 3.2

节 中所有基于加权对应关系的方法（TrICP 的重叠权重、M-ICP 的 Huber 权重）均可统一纳入此框架，IRLS 是它们共同的求解引擎。

从工程上看，IRLS 的好处是改动小，现有的 GN/LM 框架都可以直接接上权重更新；但它的短板也同样明确。权重函数若过硬，真实内点中那些残差偏大的对应会被一起压低；尺度参数若过松，外点又会重新主导法方程。IRLS 只改变每个对应在当前迭代里的权重，不改变初始化、重叠率和几何可观性，因此它不能处理大初值误差、低重叠和退化结构。

3.8.4 ADMM 与近端方法

IRLS 通过加权解耦了鲁棒性与求解，但当损失函数非光滑时（ℓ_p 范数，$p<1$），权重 w_i 在零处奇异，IRLS 失效。**Sparse ICP** [15] 引入变量分裂，将配准问题表述为：

$$\min_{R\in SO(3),\,t,\,z_i} \sum_{i=1}^{n} \|z_i\|_2^p \quad \text{s.t.} \quad Rp_i + t - q_i = z_i, \quad p \in [0,1] \tag{66}$$

ADMM 对增广 Lagrangian $\mathcal{L}_\rho = \sum_i \|z_i\|^p + (\rho/2)\|Rp_i + t - q_i - z_i + u_i\|^2$ 交替极小化，分解为三个独立子步骤：

步骤 1（位姿更新）： 固定 z, u，对 (R,t) 最小化。ℓ_2^2 惩罚项使此子问题退化为加权 P2P，可用 SVD 精确求解。

步骤 2（近端算子）： 固定 (R,t) 更新 z_i，解 $\min_{z_i} \|z_i\|^p + (\rho/2)\|r_i - z_i\|^2$（$r_i = Rp_i + t - q_i + u_i$）。此为 ℓ_p **软阈值（soft-thresholding）**问题，对 $p=1$ 有 $z_i^* = \mathrm{sign}(r_i)\max(|r_i| - 1/\rho, 0)$；对 $p<1$ 可用广义软阈值迭代近似。

步骤 3（对偶更新）： $u_i \leftarrow u_i + Rp_i + t - q_i - z_i$。

ADMM 的核心洞察在于将**刚体几何约束**（步骤 1，SE(3) 流形上的 SVD）与**稀疏诱导范数**（步骤 2，近端算子）完全解耦——IRLS 做不到这一点，因为它假设损失函数可微。一般而言，p 越小目标越“稀疏/非凸”，优化越困难，ADMM 收敛也更慢。

原文针对”稀疏 + 近端”框架给出了一组典型量化结果：在“owl”虚拟扫描对齐（原文图 4）里，粗初值的 RMSE 为 4.0×10^{-1}；传统 ℓ_2（$p=2$）再配一个距离阈值剔除时，$d_{th}=5\%$ 仍有 4.1×10^{-1}，$d_{th}=10\%$ 可降到 2.9×10^{-2}，但 $d_{th}=20\%$ 又回升到 7.5×10^{-2}（d_{th} 按包围盒对角线百分比定义）；换成 ℓ_1（$p=1$）后可做到 1.6×10^{-2}；再把范数压到 $p=0.4$，RMSE 进一步降到 4.8×10^{-4}[15]。上述结果表明：硬阈值对场景尺度与初值质量高度敏感，而稀疏范数将内外点判别转化为连续的自适应过程，从而消除了阈值参数的敏感性。

Efficient Sparse ICP [69] 则更偏实现层面。作者把瓶颈拆成两部分：一部分来自 ℓ_p（尤其 $p \ll 1$）本身的强非凸，另一部分来自最近邻和距离查询的常数开销。对应地，求解也分成两段：先通过全局探索将位姿推入更可靠的收敛区域，再切换至 ADMM 做精配准；实现上则用 OpenVDB 的层次体素做距离查询，配合并行和均匀降采样压低常数项。论文在 Intel i7-3820 @ 3.6 GHz（4 核，最多 8 线程）上给出的时间很具体（原文表 1，单位 s；目标点数约 155k）：VDB 的查询开销 $T_p = 1.9$，源点 154k/77k/38k/9k 时总耗时分别为 25.5/9.3/4.1/1.8；ANN 的 $T_p = 0.01$，但总耗时为 78.4/22.8/9.1/2.4；kd-tree 也是 $T_p = 0.01$，总耗时却高达 890.6/236.8/67.6/8.5[69]。同文还报告，在其示例数据上端到端对齐约需 11 s，相对 Sparse ICP 可提速约 31 倍（原文图 7）；内存占用约 30–90 MB，默认使用 $\ell_{0.4}$ 范数与 VDB 体素边长 3[69]。

这类方法更适合离线或重精度场景。p 取得越小，目标函数的非凸性越强，罚参数、终止条件和初始化就越敏感；再加上 ADMM 迭代、距离场构建和全局探索都会增加时延，这类方法不能直接承担高频里程计前端。

twidthhtwidth

图 34: （a）IRLS 通过权重函数把“鲁棒损失”转化为一系列加权最小二乘；（b）Sparse ICP 以 ADMM 变量分裂把“刚体几何”与“稀疏/非光滑先验”解耦；（c）当目标从更接近 ℓ_2 逐步走向更稀疏的设定时，优化难度上升，混合策略可缓解收敛困难（定性示意）。

3.8.5 $SE(3)$ 流形优化

欧拉角参数化的奇异性（万向锁）和四元数的单位约束均会给迭代优化引入额外复杂性。直接在 $SE(3)$ 流形上做优化可完全规避这两个问题 [67]。

$SE(3)$ 元素 $T=(R,t)$ 的**左扰动**模型为 $T'=\exp(\hat{\xi})\cdot T$，其中 $\xi=(\phi,\rho)\in\mathfrak{se}(3)$，旋转分量 $\phi\in\mathbb{R}^3$，平移分量 $\rho\in\mathbb{R}^3$。指数映射由 Rodrigues 公式计算：

$$\exp(\hat{\phi})=I+\frac{\sin\|\phi\|}{\|\phi\|}\,\hat{\phi}+\frac{1-\cos\|\phi\|}{\|\phi\|^2}\,\hat{\phi}^2 \tag{67}$$

左 vs 右扰动的工程选择： 左扰动 $\exp(\hat{\xi})\cdot T$ 对应世界系中的增量，Jacobian 形式更简洁；右扰动 $T\cdot\exp(\hat{\xi})$ 对应体系中的增量，在 IMU 预积分中更自然。KISS-ICP [68] 采用右扰动以配合恒速运动模型的运动补偿；LIO-SAM [70] 采用左扰动对接 GTSAM 因子图。

KISS-ICP 的“极简”不是口号：它基本就靠 P2P 残差、Lie 群扰动、自适应数据关联阈值和 Cauchy 鲁棒核把前端跑稳 [68]。具体到参数，关联阈值从 $\tau_0=2\,\mathrm{m}$ 起步，再随运动强度自调上界。论文在 KITTI benchmark（使用已做运动补偿的数据）上给出 00–10 序列平均平移误差 0.50%，11–21 为 0.61%（原文表 II）；换到 KITTI-raw，不做 deskewing 时频率约 51 Hz，但误差会上升到 0.91%/0.27；加入 deskewing 后频率约 38 Hz，平移/旋转误差为 0.49%/0.16（常速模型）或 0.51%/0.19（用 IMU）[68]。这些数字说明，右扰动配合 LM 和鲁棒核，确实能把优化稳定性和系统节拍同时兜住。

但流形优化本身并不会消掉问题的不可观性。若场景只有单平面、长走廊或局部几何过弱，把参数从欧拉角换成 $se(3)$ 只会消除参数化奇异性，不会增加观测约束；遇到大角度错位或错误对应占主导时，流形上的 GN/LM 仍然收敛到错误解。

twidthhtwidth

图 35: （a）$SE(3)$ 流形与切空间示意：T_0 处的切平面 $\mathfrak{se}(3)$ 用六维坐标表示旋转与平移增量；左扰动（世界系增量）与右扰动（体系增量）对应不同的“把增量乘到哪里”。（b）Rodrigues 公式的几何直观：旋转向量的方向决定旋转轴，模长决定旋转幅度，点 p 沿圆弧映射到 Rp。（c）旋转参数化对比：欧拉角存在万向锁；四元数需单位约束且存在双覆盖；$\mathfrak{so}(3)$ 向量更新自然但需注意大角度时的映射性质。

3.8.6 可认证与全局最优方法

当初始位姿误差较大或外点比例极高时，GN/LM 和 IRLS 均可能收敛到局部极小。可认证方法（certifiable methods）通过松弛或连续化策略在理论上保证全局最优性。

GNC（Graduated Non-Convexity） [71] 从 Black-Rangarajan 对偶性出发，将鲁棒估计转化为带权重的 LS 序列。其核心是构造一族代理损失 $\rho_\mu(\cdot)$（参数 μ 控制非凸程度），μ 较大时 ρ_μ 近似为凸函数，随 μ 逐渐减小趋近目标鲁棒函数。每个 μ 值下的权重可用闭合公式计算（TLS 核情形可写为 $w_i=\max(0,1-(r_i/\mu)^2)^2$）。

这套“从易到难”的连续化过程到底能扛多少外点，原文给过一组很清楚的小实验：以 Stanford Bunny 为例，模型先缩放到单位立方体，再从对应集中采样 $N=100$ 个点对，内点叠加高斯噪声 $\sigma=0.01$，外点比例从 60% 一直扫到 95%[71]。在该文的位姿图实验里，外点比例到 80% 时，RANSAC 约需 218 ms，而 GNC-GM 和 GNC-TLS 分别约为 22 ms 和 23 ms。这里更值得注意的不是“谁更快”，而是 GNC 把鲁棒估计改写成一串连续加权问题以后，原来大量靠随机试错消耗的时间被省掉了 [71]。

GNC 仍然要处理路径调度问题。μ 降得过快，优化会在还没进入正确盆地时就面对强非凸目标；μ 降得过慢，迭代次数和调参成本都会显著增加。它本质上仍是局部优化，只是把原来的单个非凸问题拆成一串更容易求的子问题。

TEASER++ [36] 以截断最小二乘（TLS）代价 $\min\sum_i\min(\|e_i\|^2,\bar{c}^2)$ 为起点，通过**图论解耦**把配准拆成三个级联子问题：尺度估计（投票）$\rightarrow$ 平移估计（投票）$\rightarrow$ 旋转估计（松弛 + 检验）。对应层面先做最大团剪枝，把明显不一致的外点从候选集合里删掉；旋转部分用 GNC 求解，并用 Douglas–Rachford splitting（DRS）做后验检验：检验通过时，解的最优性可以直接被验证。论文在标准基准与 3DMatch 扫描匹配上评测，并给出两条“很硬”的结论：尺度已知时对外点比例可鲁棒到 >99%，且 TEASER++ 的单次求解可在毫秒级完成 [36]。

TEASER++ 最突出的地方是它对极端外点的承受能力。原文给出的结果是：尺度已知时，外点比例超过 99% 仍可恢复；尺度未知时，外点比例到 90% 也还有成功案例 [36]。在作者的实现里，单次求解可做到 $< 10\,\text{ms}$。但它要求输入对应中仍保留一组几何上自洽的内点；如果对应已经被重复结构或描述子失配完全打散，最大团剪枝和后续旋转估计都不会成功。因此，TEASER++ 适合做困难帧初始化、回环精配准或离线配准，不适合替代每一帧的常规前端。

SE-Sync [72] 面向的是位姿图 SLAM 里的旋转同步问题。它把 MLE 写成半定松弛（SDP），再用流形上的截断 Newton 信赖域去解，并给出“什么时候松弛是紧的”这类可认证条件。论文里最有说服力的是那张运行时间表（原文表 1）：在 sphere（2500 poses / 4949 measurements）上，Gauss-Newton 为 14.98 s，SE-Sync 为 2.81 s；torus（5000/9048）为 31.94 s vs 5.67 s；grid（8000/22236）为 130.35 s vs 22.37 s；rim（10195/29743）为 575.42 s vs 36.66 s，总体加速大致在 3.3 倍到 15.7 倍之间 [72]。这些结果说明，可认证松弛并不一定天然比传统迭代慢。

但 SE-Sync 的适用边界也很明确：它针对的是整个位姿图层面的同步问题，不是单帧点到点配准前端。它需要的是多帧之间的相对位姿约束图，而不是一对点云的对应集合，因此不能直接替代单帧 ICP 求解器。

三类方法的核心差异在于：GNC 是把非凸目标沿一条连续路径慢慢引入；TEASER++ 是把尺度、平移、旋转拆开以后分别求，并在旋转部分附上松弛检验；SE-Sync 则直接在图层面做整体松弛。Go-ICP（见第 3.6 节 ）的分支定界在外点受控时也能做到全局最优，但外点一多复杂度就会迅速失控；相比之下，TEASER++ 更强调的是“对应很脏时还能给出一个可验证的解”。

twidthtwidth

图 36: （a）GNC 连续化路径示意：代理损失从更“凸/平滑”逐步演变为目标鲁棒损失，权重更新与加权 LS 交替进行；（b）TEASER++ 级联求解与松弛检验流程示意：对应剪枝 $\rightarrow$ 解耦投票 $\rightarrow$ 旋转松弛 $\rightarrow$ 松弛检验；（c）从“鲁棒性”到“计算代价”的二维视角比较：GNC、TEASER++、SE-Sync 在不同问题难度下的适用区间（定性示意）。

3.8.7 因子图与 SLAM 集成

ICP 位姿估计的误差不均匀，走廊场景的轴向漂移（见第 3.5 节 ）和回环时的累计误差都会要求将单帧 ICP 结果纳入全局一致优化。因子图（factor graph）将 ICP 残差建模为两帧位姿之间的二**元约束因子**，与 IMU 预积分因子、回环检测因子并列优化。

全局位姿估计问题可表述为非线性最小二乘：

$$x^* = \arg\min_{x} \sum_{(i,j)\in\mathcal{E}} e_{ij}(x_i, x_j)^\top \Omega_{ij}\, e_{ij}(x_i, x_j) \tag{68}$$

其中 $e_{ij} = \log(T_{ij}^{-1} \cdot T_i^{-1} T_j)$（相对位姿残差），$\Omega_{ij}$ 为信息矩阵。ICP 给出的 $\Omega_{ij} = J^\top \Sigma^{-1} J$（Hessian）与第 3.5 节 中的 Hessian 秩分析直接对应：当配准在退化方向缺乏约束时，Ω_{ij} 在对应行列接近零，[27] 在该节讨论的退化感知因子正是在此基础上只提交约束充分的方向的因子边。

g2o [73] 是通用图优化框架，它把位姿图的稀疏雅可比直接落到稀疏线性系统上，支持 GN、LM 和 Dogleg 等求解器。论文里有一张很能说明问题的表（原文表 II）：在单核 i7-930 @ 2.8 GHz 上，Venice BA 的每次迭代若用 CHOLMOD 约为 1.86 s，CSparse 约 39.1 s，而 PCG 为 0.287 ± 0.135 s；New College 上，CHOLMOD 约 6.19 s，CSparse 约 200.6 s，PCG 为 0.778 ± 0.201 s[73]。这些数字说明，后端图优化的瓶颈经常落在线性求解器上，而不是 GN、LM 或 Dogleg 这些外层迭代形式上。

因此，因子图优化的短板主要在图质量和数值实现。前端若持续送入带偏因子、回环约束若包含假阳性，或者线性化点长期偏离真实轨迹，后端优化会把这份误差继续传播到整张图上。

GTSAM + iSAM2 [74]，[75] 引入 Bayes 树数据结构，将增量式重线性化限制在受影响的子树节点上。iSAM2 的表格同样很直白：例如 Manhattan（3500 poses）每步平均约 2.44 ms，总计 8.54 s；City20000（20000 poses）每步平均约 16.1 ms，但最坏一步可到 1125 ms（原文表 1）[75]。这说明 iSAM2 的平均时延很低，但遇到大规模重线性化时，单步延迟会出现尖峰。

LIO-SAM [70] 以 iSAM2 为后端，同时注册三类因子：LiDAR 里程计因子（本节 ICP 结果，6×6 信息矩阵）、IMU 预积分因子（15 维状态）、GPS 因子（选配）与回环因子（ICP 精配准结果）。作者把信息矩阵“怎么来”说得很直接：LiDAR 因子用 P2Pl ICP 的 Hessian H 直接构造信息矩阵，退化场景下（如走廊）H 秩亏，对应方向权重自动变弱。

这套链路在论文的结果里也能直接看出来：在 i7-10710U 笔记本上（CPU-only），LIO-SAM 在 Campus、Park、Amsterdam 等序列上的平均耗时约为 97.8、100.5 和 79.3 ms per scan（原文表 IV）；终端轨迹误差的差距更明显，例如 Campus 上 LOAM 为 192.43 m，而 LIO-SAM 为 0.12 m，Park 上 LOAM 为 121.74 m，而 LIO-SAM 只有 0.04 m（原文表 II）[70]。这些数字放到求解器视角里，其实是在说明一个更结构性的事实：前端每帧的 ICP 不是孤立的局部最小二乘，它的 Hessian 会继续传到后端，成为因子图中那条边的权重和置信度。

但这种“前后端一体”的风险也很直接。前端若长期在退化场景里输出带偏 Hessian，后端会把这份偏置当作可信信息继续累计；若回环检测再引入错误约束，整张图都会被一起拉偏。

twidthtwidth

图 37: （a）LiDAR SLAM 因子图结构：圆形节点为位姿变量 x_i；蓝色实线为 LiDAR 里程计因子（由 ICP Hessian/信息矩阵给权），橙色虚线为 IMU 预积分因子，绿色粗线为回环因子；退化帧的 ICP 因子以“部分约束”形式提交（对应第 3.5 节 ）。(b）信息矩阵热力图对比：正常场景约束更完整；退化场景在某些自由度方向上约束显著变弱。(c）iSAM2 Bayes 树增量更新示意：新因子到来时只重线性化受影响的子树节点，避免反复全量批优化。

3.8.8 综合对比

表 19: ICP 优化求解器综合对比

方法	**类别**	**收敛域**	**全局最优**	**外点鲁**
hlineGauss-Newton	一阶 NLS	局部（强依赖初值）	否	弱（需
信赖域 NLS	局部（更稳）	否	弱（需配鲁棒核）	是
局部（依赖权重）	否	中（核/权重决定）	是 ICP (ADMM)	近端/变
否	强（非光滑/稀疏先验）	否（通常为离线模块）Sparse ICP	全局探索 + ADMM	更宽（
强（依赖设定）	否	连续松弛	宽（经验上）	近似/可
可用（位姿图/因子图）++	TLS + SDP + 图论	宽（初值弱依赖）	可证明（可检验）	强（对
流形 SDP	全局（旋转同步）	可证明（可检验）	噪声受控场景更稳	位姿图
hline				

求解器的选择由场景需求主导：实时系统（自动驾驶、UAV SLAM）通常以 GN/LM + Lie 群扰动为主循环，IRLS 或 GNC 提供在线外点鲁棒性，iSAM2 因子图支撑全局一致性；离线精配准或初始位姿完全未知时，TEASER++ 或 GNC 的全局搜索能力不可或缺。Sparse ICP 的 ADMM 框架在形状建模和曲面重建领域独树一帜，其近端分裂思路正向连续优化（Proximal Gradient Descent）和深度展开（Algorithm Unrolling）扩展。

3.9 本章小结

本章将 ICP 视为“可替换模块的迭代流水线”并据此组织变体：第 3.1 节 通过改变度量与约束形态提升对应质量（从 P2P/P2Pl 到 NDT 与语义引导）[3]；第 3.2 节 通过截断估计、鲁棒核与图论筛选增强外点鲁棒性（如 TrICP、Sparse ICP、SUCOFT）[14]；第 3.3 节 则在“已处于正确盆地”的前提下，通过 Anderson 加速与 MM/GNC 框架降低迭代成本并稳定优化轨迹 [17]；第 3.4 节 回顾了从闭式解到概率化估计的变换更新策略（如 Horn/Kabsch 型解法、GICP、Stein ICP），强调参数化选择与不确定性建模会直接影响数值稳定性与收敛行为 [12]。

与“单帧、局部、几何最小二乘”的经典设定相比，第 3.5 节 至第 3.8 节 进一步把 ICP 推向工程系统所需的更强假设边界：第 3.5 节 揭示了几何退化本质上是可观测性不足的问题，并给出检测与约束提交的处理路

径 [27]；第 3.6 节　通过全局初始化与可认证估计把误差压入可收敛区域，为局部 ICP 提供“可用起点”（FGR、Go-ICP、TEASER++ 等）[36]；第 3.7 节　讨论学习型特征、软对应与端到端网络在低重叠和重复结构中的优势与风险 [18]；第 3.8 节　从求解器视角统一这些方法，说明鲁棒核、稀疏范数与可认证松弛分别对应不同的优化结构与计算代价边界 [72]。

在实践中，算法层面的鲁棒化与初始化只是“让问题可解”，而系统级实时性往往受制于近邻查询与线性代数的吞吐。第 4 章将围绕该计算瓶颈，从数据结构、降采样、并行化与近似搜索四个层面总结可复用的软件加速策略。

4. 软件加速：数据结构、降采样与近似算法

第 3 章讨论了 ICP 在目标函数、对应筛选和初始化策略上的改进，但这些变体仍共享同一计算核心：在目标点云 $\mathcal{Q}$ 中为源点云 $\mathcal{P}$ 的每个点搜索对应邻域。Pomerleau 等在六类真实场景上的公开评测表明，ICP 的配准表现不仅受误差模型影响，也直接受输入点数、采样方式和查询实现约束；在相同实验协议下，点到面变体的精度比点到点高约 20%–40%，但点到点实现的运行速度约快 80% [23]。由此可见，软件实现并非附属问题，而是决定 ICP 是否能进入实时预算的前提条件。

本章转向软件实现层面的四条主线。第 4.1 节　比较 KD-Tree、体素哈希、Octree 与增量索引的适用边界，并以 FAST-LIO2 的 ikd-Tree 为例说明动态地图维护为何必须与近邻查询一并设计；该系统在 19 个公开序列上完成评测，在大场景中达到 100 Hz，并在最高 1000 deg/s 的旋转条件下仍能给出稳定状态估计 [42]。第 4.2 节　讨论降采样如何改变信息密度分布，而不仅是减少点数；其中 EA2D-LSLAM 在 KITTI 和 M2DGR 上将后端运行时间从 95 ms 降到 68 ms，前提是按体素信息矩阵保留对位姿约束更强的区域 [76]。第 4.3 节　分析 SIMD、多线程与 GPU 的分工关系，第 4.4 节　则讨论近似最近邻的误差容忍机制，并补入 HNSW 这类图索引在高召回近似搜索中的桥接作用 [77]。本章的目标不是罗列优化技巧，而是说明这些策略各自依赖什么场景假设，以及在什么条件下会先失效。

4.1 数据结构优化 (Data Structure Optimization)

ICP 的内循环在每次迭代中对源点云 $\mathcal{P}$ 中的全部 n 个点各执行一次最近邻查询（KNN，$k=1$），总查询量为 $n \times I$ 次（I 为迭代次数）。当点数进入 10^5 量级后，单次迭代就需要处理数百万次三维距离计算，最近邻搜索因此成为软件实现中最先触及内存访问瓶颈的环节。[78] 将这一问题作为点云处理器设计的核心负载之一，说明数据结构的选型不仅影响算法复杂度，还会改变缓存命中、访存规则性以及后续并行化空间。本节据此比较 KD-Tree、体素哈希（Voxel Hashing）与 Octree 三类主流结构，并以 ikd-Tree 说明动态地图场景下为什么需要“可维护”的索引而不只是“可查询”的索引。

4.1.1 KD-Tree：标准实现与性能分析

KD-Tree（k-dimensional tree）由 Bentley 于 1975 年提出，通过在每个节点沿方差最大的坐标轴对点集进行二叉划分，构建出一棵平衡二叉搜索树。在三维点云中（$k=3$），构建过程如下：

1. 选择当前点集中方差最大的坐标轴 $d^* = \arg\max_d \mathrm{Var}(x_d)$。
2. 取该轴的中位数 m 作为分割值，将点集划分为左子树 $\{p : p_{d^*} \le m\}$ 和右子树 $\{p : p_{d^*} > m\}$。
3. 递归地对每个子集重复，直到子集大小 $\le$ 叶节点容量（常设为 1—10）。

构建时间为 $O(n \log n)$（含中位数选取），树高为 $O(\log n)$，空间占用 $O(n)$。单次最近邻查询的期望时间为 $O(\log n)$，但在最坏情形（点云极度非均匀分布）下退化为 $O(n)$。

查询算法（最佳优先搜索）：从根节点出发，根据查询点 q 与分割平面的位置优先进入” 更可能” 的子树，同时维护当前最近邻候选和对应距离 d_{best}。在回溯时，若当前节点所在超平面到查询点的距离 $|q_{d^*} - m|$ 小于 d_{best}，则必须进入另一子树；否则整棵子树可剪枝。KD-Tree 的优势在于大多数查询只访问少量节点，但这一

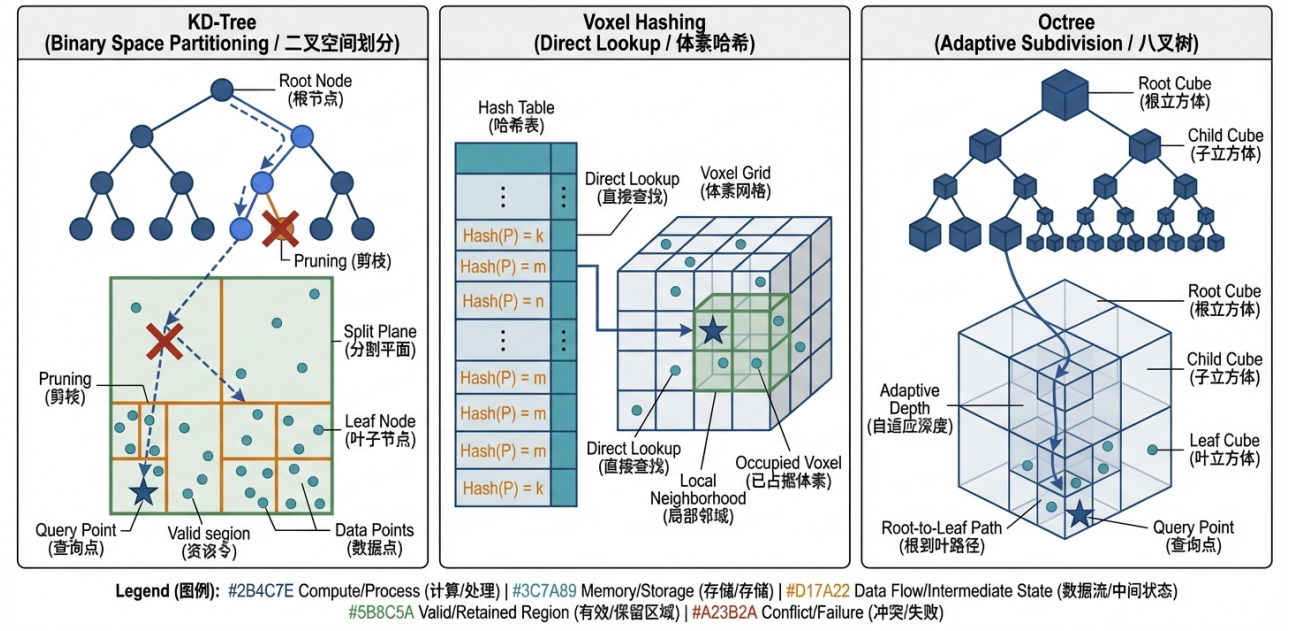


图 38: 三种近邻搜索数据结构示意。左：KD-Tree 的二叉空间划分，查询时沿树路径剪枝；中：体素哈希将空间均匀量化为三维格，$O(1)$ 哈希定位候选体素；右：Octree 自适应八叉划分，密集区域细分更深。颜色深浅表示子节点点云密度。

优势依赖点云分布与分裂轴选择。当场景近似均匀、树保持平衡时，查询路径较短；当场景退化为走廊、隧道或长条结构时，分裂平面难以快速排除候选区域，回溯次数会上升，查询时延也随之扩大。

批量查询优化：ICP 每次迭代需对 n 个源点批量查询。若先按空间邻近度对源点分组，相邻查询多会访问相似的树路径，缓存复用会明显改善。因此，KD-Tree 在 CPU 上是否高效，不只取决于树本身是否平衡，也取决于查询顺序是否尽量减少随机访存。[79] 提出的 cached k-d tree 正是利用了这一点：作者在室内杂乱环境、户外场景、废弃矿井、救援竞技场和面部扫描五类数据上比较标准 k-d tree、近似 k-d tree 与缓存搜索，报告在保持精确最近邻的前提下平均约 50% 的搜索加速。它成立的条件是相邻 ICP 迭代之间的位姿变化足够小，缓存叶节点仍具有延续性；若初值偏差大或场景切换过快，缓存命中优势会先消失。

PCL 和 nanoflann 实现对比：PCL（Point Cloud Library）的 KD-Tree 基于 FLANN [80]，便于直接接入近似搜索与多线程查询；nanoflann 则保留更轻的模板实现，适合点云维度固定、部署环境受限的场景。前者适合快速验证与算法切换，后者更适合把查询路径和内存布局一起做细化优化。

4.1.2 ikd-Tree：支持动态更新的增量式 KD-Tree

标准 KD-Tree 在树构建后不支持高效插入/删除。对离线配准，这一限制并不突出，因为目标点云多为固定；但对 LiDAR 里程计和建图，地图会随着机器人运动持续变化，若每次加入新点都整体重建，索引维护本身就会吞掉实时预算。

[42] 在 FAST-LIO2 中提出 **ikd-Tree**（Incremental KD-Tree），目的不是改变 ICP 的匹配准则，而是在动态地图上把“查询”和“维护”合并到同一套索引中。FAST-LIO2 在 19 个公开序列上评测，覆盖旋转式与固态 LiDAR、无人机与手持平台，并报告了大场景中 100 Hz 的里程计与建图频率，以及最高 1000 deg/s 旋转条件下仍可稳定估计位姿。这些结果说明，ikd-Tree 的价值主要体现在持续更新地图时仍能保持查询可用，而不是单次静态查询一定优于所有 KD-Tree 实现。其动态操作包括：

- **增量插入**：维护部分不平衡状态，插入新点时只更新受影响路径，并在局部失衡时触发局部重平衡。这样做的理由是把维护代价限制在局部子树，而不是把实时系统拖回整树重建。
- **增量删除**（懒删除）：不立即从树中移除节点，而是先将节点标记为” 已删除”，查询时跳过这些节点，待局部无效节点堆积后再统一重建。这样可以避免在每帧小幅视角变化时频繁触发重排。

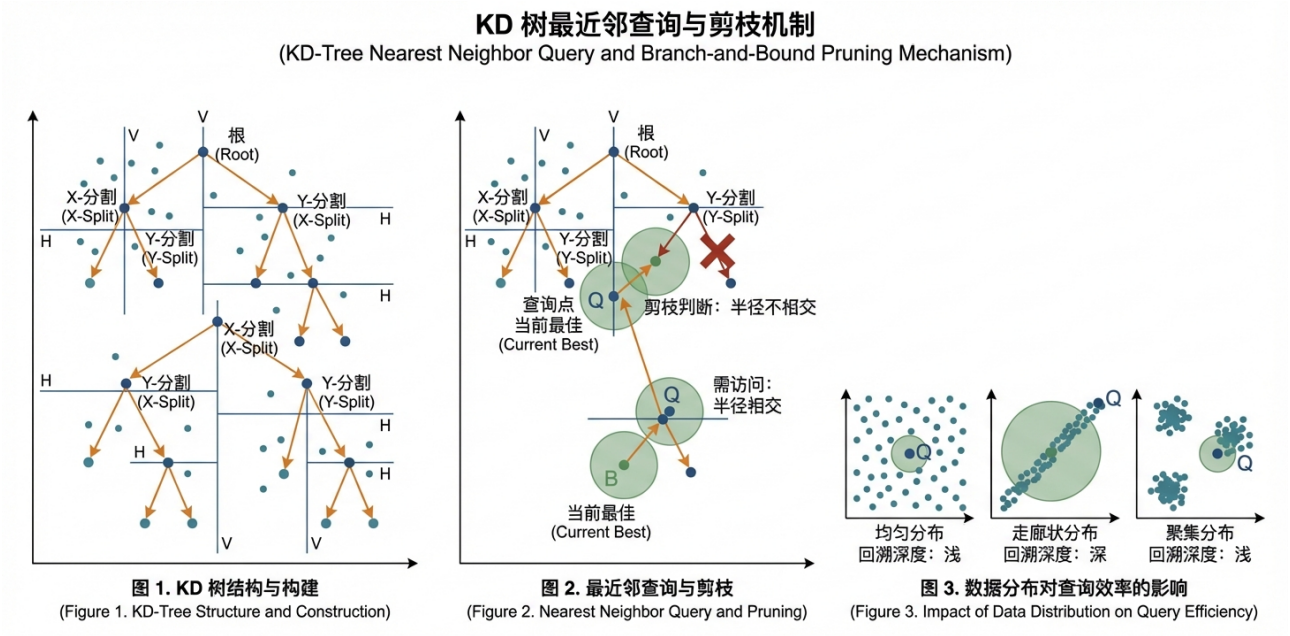


图 39: KD-Tree 最近邻查询的分支界定（Branch-and-Bound）剪枝过程示意。左：二维 8 点的树结构与分割平面；中：查询点 q 的回溯与剪枝步骤；右：用均匀分布、线性结构与聚类分布对比说明，点云几何一旦退化，KD-Tree 的回溯深度会增加。该图为机制示意，不对应具体测量数值。

- **动态重平衡**：局部重建仅在不平衡子树上进行，时间与子树大小成正比，不影响整棵树。

与 Octree、R^*-Tree 和静态 KD-Tree 的对比中，[42] 将 ikd-Tree 描述为在增量更新场景下总体更均衡的方案：查询性能保持在 KD-Tree 量级，而插入、删除和局部重建不再迫使系统每帧整树重建。它成立的前提是场景以连续运动为主、每次更新只改变局部地图；若环境快速进出大量动态物体，懒删除节点会持续累积，局部重建频率上升，维护收益就会下降。

4.1.3 体素哈希（Voxel Hashing）

体素哈希将三维空间均匀量化为边长 r 的立方体体素，用哈希表（或固定大小的 bucket 数组）存储每个体素内的点集：

$$\mathrm{key}(p) = \left(\left\lfloor \frac{p_x}{r} \right\rfloor, \left\lfloor \frac{p_y}{r} \right\rfloor, \left\lfloor \frac{p_z}{r} \right\rfloor \right) \tag{69}$$

查询时，计算查询点 q 所在体素坐标，通过哈希函数 $h(\mathrm{key})$ 在 $O(1)$ 时间内定位该体素及其 $3^3 - 1 = 26$ 个面、棱、角邻域体素（共 27 个），再对这 27 个体素内的所有点暴力搜索最近邻。体素哈希的关键优势在于：

1. **插入/删除** $O(1)$：新点直接插入对应体素的桶（bucket），无需重建树结构，天然支持动态 LiDAR 点云地图。
2. **内存访问规律**：27 个邻域体素在哈希表中的访问模式固定，GPU 可预取（prefetch）邻域数据，缓存友好度高于 KD-Tree 的随机树遍历。
3. **并行化极为简单**：每个查询点的 27 个体素检索相互独立，直接分派到 GPU 线程。

体素哈希的主要劣势在于精度受分辨率 r 约束：若 r 过大，候选集会把多个几何结构混入同一体素，最近邻分辨率下降；若 r 过小，哈希表会过稀，访存与管理开销反而上升。因此，体素哈希并不是“总比 KD-Tree 快”，而是只在查询半径、点间距和硬件访存模式相互匹配时才真正占优。

对于室内或实验室环境这类点密度相对均匀的场景，体素哈希更容易保持固定的候选规模；但在室外 LiDAR 扫描中，点密度随距离迅速衰减，远处体素可能几乎没有有效候选，此时查询质量更依赖体素尺度是否随距离调整，否则对应关系会先在稀疏区域失真。

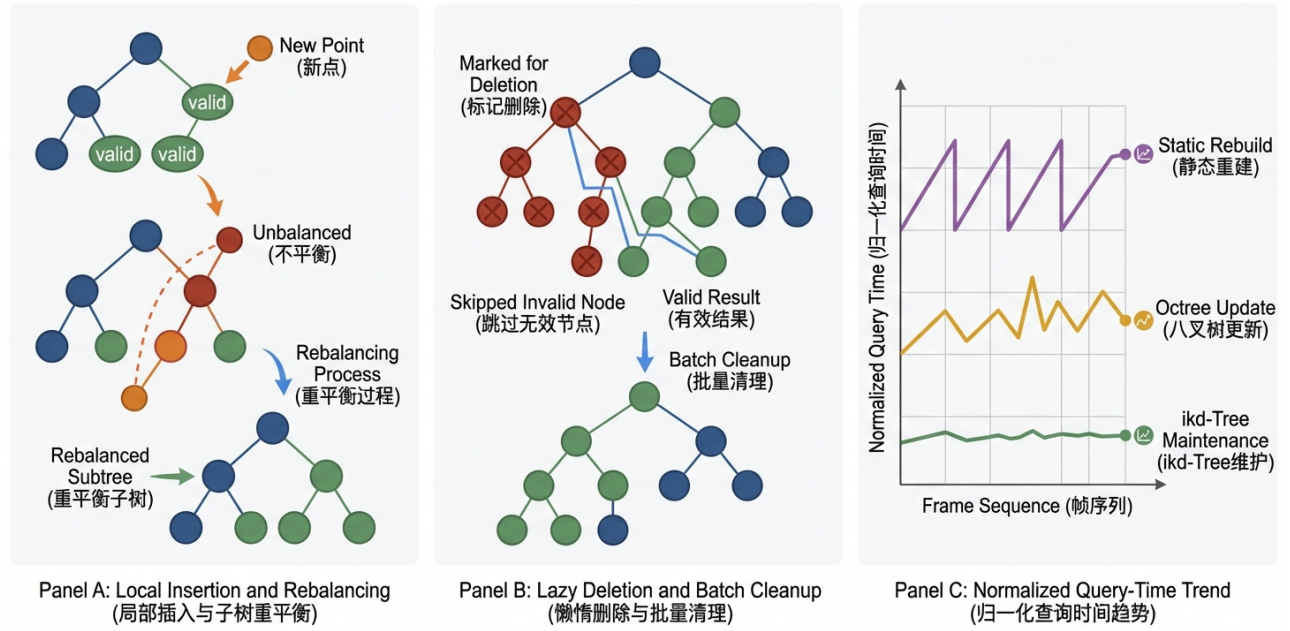


图 40: ikd-Tree 增量更新机制示意。左：新点插入后仅在局部子树重平衡；中：懒删除节点保留到批量清理；右：用归一化曲线说明连续更新场景下，局部维护比逐帧整树重建更稳定。该图为机制示意，不对应具体测量数值。

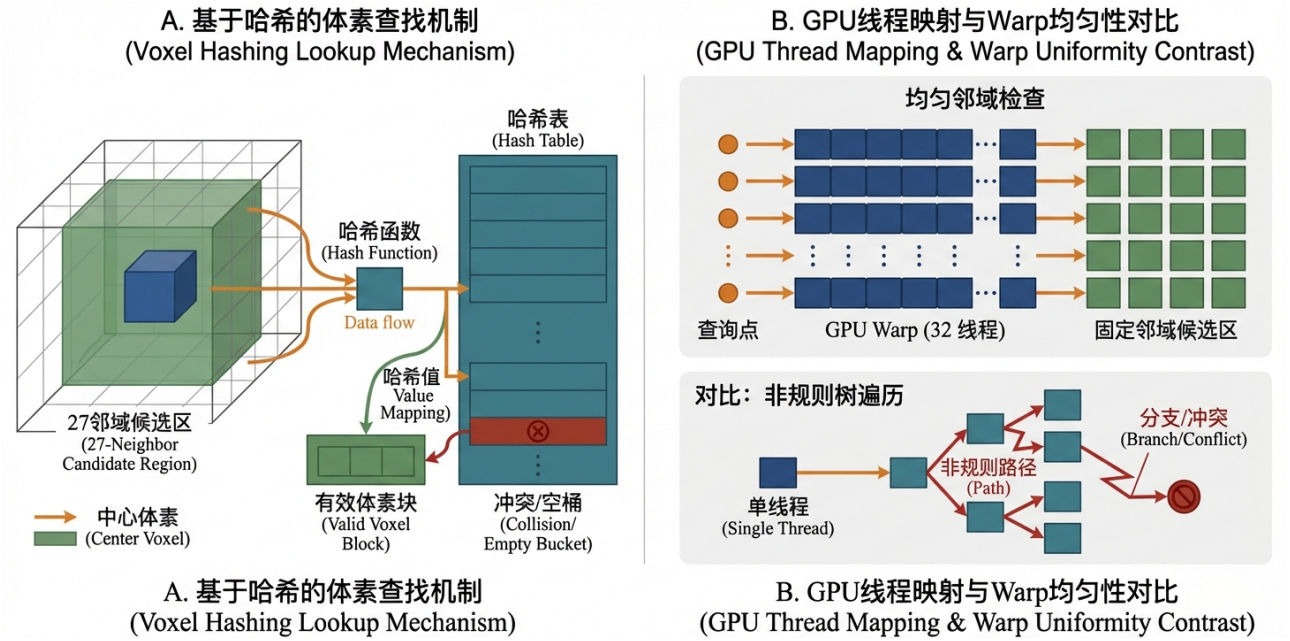


图 41: 体素哈希的 27 邻域访问模式及其与 GPU 并行架构的对应关系。左：查询点落入中心体素后，只需检查固定邻域中的候选体素；右：每个查询点可映射到独立线程，说明这种规则访问模式比树遍历更容易在 GPU 上保持同步执行。该图为机制示意，不对应具体 GPU 利用率数值。

4.1.4 Octree：自适应空间分解

Octree 将空间递归地八等分，直到每个叶节点的包含点数低于阈值 N_{leaf}（工程上多设为 1—8）。与 KD-Tree 的随机维度划分不同，Octree 的划分轴固定为三个坐标轴的中点，使得每层的划分线形成规则的三维网格，树结构更均匀。

Octree 的主要特性：

- **空间感知**：Octree 的节点天然对应空间区域，便于基于空间范围的批量查询（例如查找半径 r 内所有点）——ICP 在法向量计算、曲率估计等前处理步骤中需要此类查询。
- **自适应分辨率**：稠密区域节点深度大（分辨率高），稀疏区域节点深度小（节省内存），对非均匀 LiDAR 点云有良好的空间利用率。
- **GPU 并行友好**：Octree 节点的 8 路分支因子与 GPU 硬件的 warp/wavefront（32/64 线程）不匹配，需要用 SIMD 宽度设计特殊的打包方式；相比之下，体素哈希的规则访问模式在 GPU 上更高效。

PCL 和 Open3D 均提供 Octree 实现；[81] 讨论专用机器人处理器时，也将规则化访存视为点云处理加速的重要前提。这类工作提示了 Octree 的一个现实价值：它不一定在单次查询上最优，但更容易与专用缓存策略和片上数据流协同设计。

4.1.5 各数据结构综合对比与选型建议

表 20: 第 4.1 节 中近邻搜索数据结构的综合对比：各维度定性评估。

数据结构	构建时间	KNN 查询期望	动态插入/删除	GPU 并行友好	内存占用	最适场景
静态 KD-Tree	$O(n \log n)$	$O(\log n)$	差（重建 $O(n)$）	低（不规则遍历）	$O(n)$	离线配准，密度均匀
ikd-Tree	$O(n \log n)$（初始）	$O(\log n)$	好（均摊 $O(\log n)$）	低	$O(n)$	LiDAR 里程计增量建图
体素哈希	$O(n)$	$O(1)$ 均摊	极好（$O(1)$）	极高	$O(n/r^3)$	GPU 加速，动态地图
Octree	$O(n \log n)$	$O(\log n)$	中（局部更新）	中	$O(n)$	范围查询，非均匀密度
Approx. KD-Forest	$O(Tn \log n)$	$O(T \log n)$	差	低	$O(Tn)$	高维特征 KNN

选型决策树：

- 离线配准（无实时约束）⇒ 静态 KD-Tree + FLANN 近似搜索。
- 实时 LiDAR SLAM（CPU 部署）⇒ ikd-Tree（增量更新）。
- GPU 加速 ICP ⇒ 体素哈希（规则访问 + $O(1)$ 查询）。
- 点云密度高度非均匀（航空 LiDAR）⇒ Octree（自适应分辨率）。

数据结构的选型不仅影响 KNN 搜索速度，还直接决定下游加速策略是否成立。并行化一节 会进一步说明，GPU 更依赖规则访问的候选组织方式；而面向专用硬件时，HA-BFNN-ICP [82] 与 Tigris [78] 分别代表两种思路：前者用顺序流式搜索换取稳定的数据通路，后者尝试把 KD-Tree 的不规则遍历拆解成硬件可调度的细粒度操作。它们都说明，索引结构不是前处理细节，而是后续体系结构设计的输入约束。

4.2 降采样与多分辨率策略 (Downsampling and Multi-Resolution Strategies)

点云配准面临的核心矛盾之一，是精度与速度的权衡：更密集的点云携带更丰富的几何信息，但每次迭代的最近邻搜索和变换估计代价随点数增加而同步上升。降采样作用于 ICP 主循环的入口，它既改变点数，也改变约束分布，因此会同时影响运行时间、收敛盆地和最终误差。与第 4.1 节 的索引优化不同，降采样先决定“哪些点有资格进入配准”。

[23] 用同一套协议比较了六类真实场景中的 ICP 变体，场景覆盖公寓、楼梯和树林等结构化与非结构化环境，指标采用旋转误差 e_r 与平移误差 e_t。该文的基线结果表明，采样策略和误差模型是同一层级的设计变量：点到面在精度上比点到点高约 20–40%，但点到点在计算时间上约快 80%。这一结果说明，降采样不能只按“保留多少点”来选，还要看保留下来的点是否继续支撑目标函数中的主要约束方向。

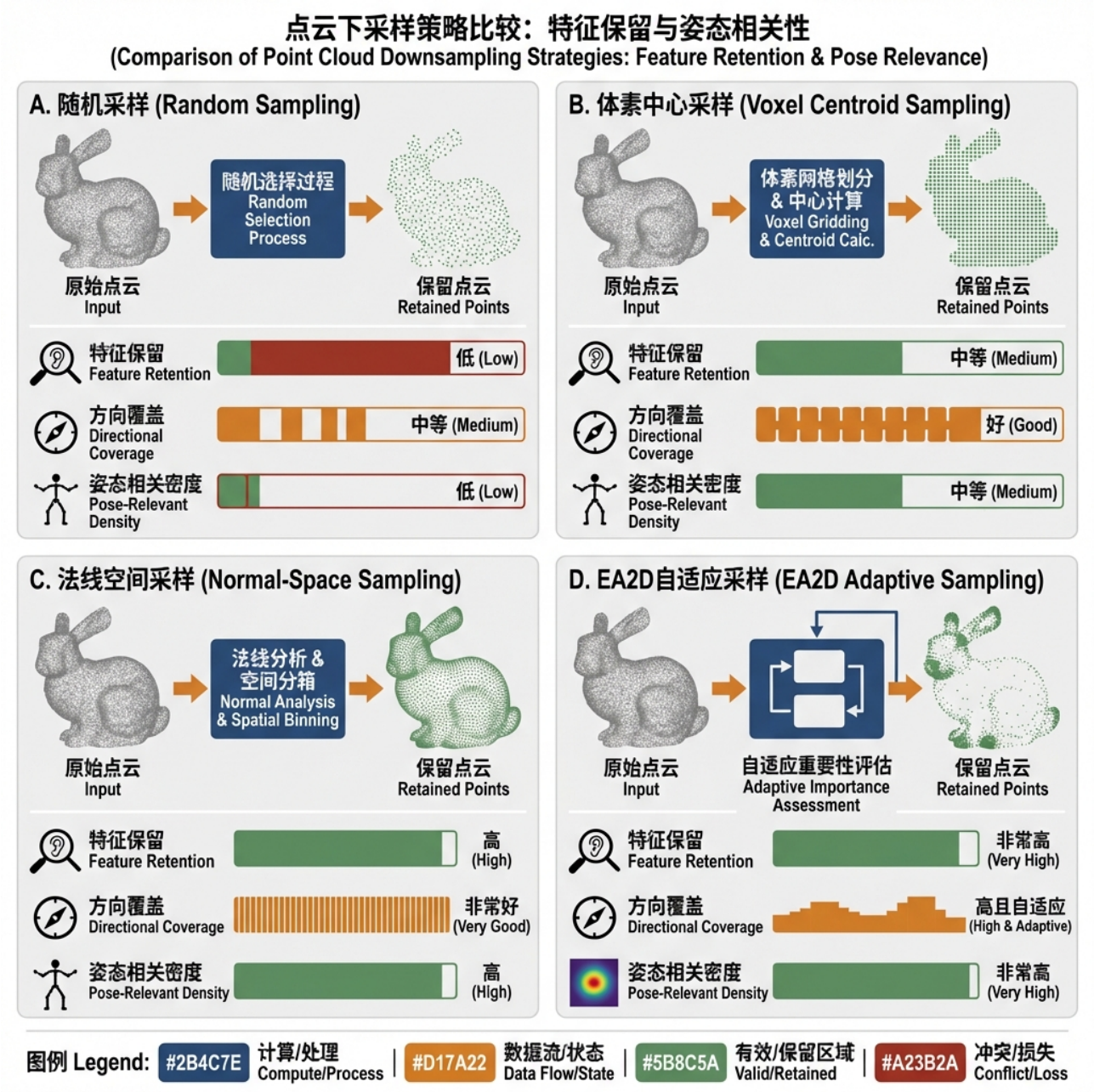


图 42: 四种主流降采样策略在同一兔子模型点云上的效果对比。左上：均匀随机采样，点密度均一但曲率丰富区域特征丢失。右上：体素质心采样，网格化划分后每格取重心，保留整体形状。左下：法向量空间均匀采样（NSS），在法向量球面上均匀分布，边缘特征保留好。右下：EA2D 自适应采样，基于 ICP Hessian 信息矩阵评分，姿态约束贡献高的点优先保留。

4.2.1 均匀随机采样

均匀随机采样（Uniform Random Sampling）是最简单的降采样方法：以目标采样率 $r \in (0, 1)$ 独立同分布地决定每个点是否保留。设原始点云有 n 个点，保留点数期望为 $\hat{n} = rn$。

均匀随机采样的优势在于零参数、零预处理、$O(n)$ 时间：遍历每个点，以概率 r 保留，无需任何空间索引。

但其根本缺陷是忽略了点云的非均匀性：

1. **密度偏差**：传感器对近处物体采样密度高、远处低；均匀采样会过多保留近处点而稀化远处点，使得 ICP 目标函数被近处密集区域主导。
2. **特征丢失**：曲率大的边缘、尖锐拐角处多为低密度区域；均匀采样后这些特征区域的点数少，对应质量下降。
3. **方差大**：每次运行的保留集不同，导致 ICP 结果不可复现，对于需要确定性输出的工业系统不可接受。

尽管如此，均匀随机采样仍适用于点云已经大致均匀、且系统允许一定结果波动的场景。它的主要价值是给出一个低成本基线：如果随机采样已经满足实时性约束，就没有必要立即引入更重的特征评分或信息矩阵分析。

4.2.2 体素质心采样（Voxel Grid Filter）

体素质心采样（Voxel Centroid Sampling 或 Voxel Grid Filter）是工程中应用最广泛的降采样方法，也是 PCL（Point Cloud Library）的默认预处理步骤。

算法步骤：给定叶片尺寸（leaf size）l，将三维空间划分为边长为 l 的正方体体素网格：

$$\text{voxel\_id}(p) = \left(\left\lfloor \frac{p_x - x_{\min}}{l} \right\rfloor, \left\lfloor \frac{p_y - y_{\min}}{l} \right\rfloor, \left\lfloor \frac{p_z - z_{\min}}{l} \right\rfloor \right) \tag{70}$$

落入同一体素的所有点 $\{p_i : \text{voxel\_id}(p_i) = v\}$ 被替换为它们的质心：

$$\hat{p}_v = \frac{1}{|V_v|} \sum_{p_i \in V_v} p_i \tag{71}$$

时间复杂度：哈希方式构建体素索引为 $O(n)$，遍历聚合为 $O(n)$，总体 $O(n)$。输出点数约为 $n \cdot (l/d)^3$，其中 d 为原始点间距。

几何含义：质心是均方误差意义下最优的单点代表；当体素内点数 $|V_v| \geq 2$ 时，质心比任意单个原始点都更靠近体素内的" 真实表面"。由于每个体素至多输出一个点，体素质心采样天然实现了空间均匀化——不论传感器原始密度如何，输出点云的空间分辨率恒为 l。

叶片尺寸选择：l 过小会保留过多点，速度提升有限；l 过大会把边缘、窄结构和法向量突变直接平均掉。Pomerleau 的协议强调，这一参数需要与传感器量程、重叠率和误差模型联调，而不是套用单一常数 [23]。工程上更稳妥的做法，是先在代表性序列上用 e_r、e_t 和单帧耗时三项指标联合搜索可接受区间，再固定到部署配置。

体素最近点（Voxel Nearest Point）变体：不取质心，而是保留距质心最近的原始点，以避免引入质心这一合成点。对于需要保留原始点坐标的应用（如对法向量敏感的 P2Pl ICP）更合适，代价是需要额外遍历计算距离。

4.2.3 法向量空间均匀采样（Normal Space Sampling, NSS）

法向量空间均匀采样由 Rusinkiewicz 和 Levoy [13] 提出，目标不是简单减少点数，而是把保留下来的点重新分配到更能约束位姿的法向量方向上。

核心思想：ICP 变换估计的精度由点集覆盖的法向量多样性决定——若所有点的法向量近似共面，则法向量垂直方向的约束几乎为零，导致系统矩阵病态（rank-deficient）。NSS 的目标是使采样后的点集在单位球面 S^2 的法向量空间中尽量均匀分布。

算法：将单位球面 S^2 离散化为 B 个均匀分布的 bin（工程实现里常将 B 设为 1000 左右）：

1. 对每个点 p_i 计算法向量 $\hat{n}_i$，映射到对应 bin $b(\hat{n}_i)$。
2. 对每个 bin b，维护一个候选列表。
3. 按 bin 均匀轮询：每轮从各 bin 随机抽一个点，直到达到目标采样数。

这等价于在法向量球面上做覆盖采样（Poisson disk sampling on S^2），确保每个法向量方向都有足够代表。

信息矩阵视角：对于点到平面 ICP，变换 (R,t) 的 Fisher 信息矩阵（Hessian of objective）为

$$\mathcal{I} = \sum_{i=1}^{n} \begin{pmatrix} \hat{n}_i\hat{n}_i^\top & \hat{n}_i(p_i \times \hat{n}_i)^\top \\ (p_i \times \hat{n}_i)\hat{n}_i^\top & (p_i \times \hat{n}_i)(p_i \times \hat{n}_i)^\top \end{pmatrix} \tag{72}$$

当法向量分布在 S^2 上不均匀时，$\mathcal{I}$ 的某些特征值接近零，对应方向的约束弱、迭代不稳定。NSS 通过强制球面均匀性，使 $\mathcal{I}$ 的条件数（condition number）最小化，每次迭代的有效信息利用率最高。

与随机采样的对比：[13] 将均匀采样和法向量空间采样组合用于“nearly-flat meshes with small features”这一典型难例，结论不是“任何场景都该优先选 NSS”，而是当主表面近似共面、只有少量小特征负责破除退化时，法向量均匀化能更快把这些特征保留下来。相反，如果原始点云法向量本就分布充分，NSS 的额外法向量估计与分箱步骤可能只增加前处理开销。

几何稳定性桥接：NSS 解决的是“方向覆盖不足”，而 [83] 进一步把问题写成线性系统条件数优化。该文在带刻槽平面、球面和 Forma Urbis Romae 真实扫描上比较了均匀采样、法向量空间采样与几何稳定采样，发现后者可将条件数从 66.1 降到 3.7。这里的改进发生在噪声和局部退化同时存在时：先失效的是线性系统对某些平移或旋转方向的约束，随后才表现为 ICP 在滑动方向上收敛缓慢或直接落入错误位姿。这个结果把 NSS 与后续基于 Hessian 的采样方法连接起来，因为二者都不再把“点数”当作唯一目标，而是直接优化姿态可观性。

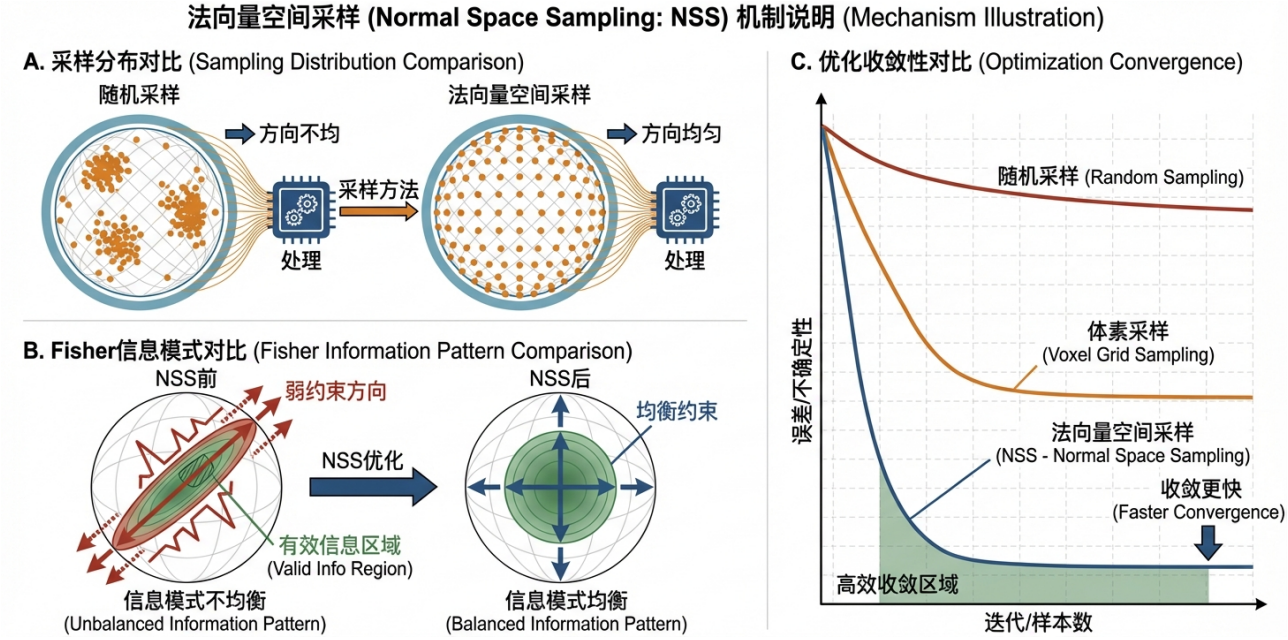


图 43: NSS 机制示意，不对应单一论文中的具体数值。图中仅说明法向量均匀化如何改善信息矩阵条件数，以及随机采样在平面主导场景中为何更容易先丢失约束方向。

4.2.4 曲率自适应采样

曲率自适应采样（Curvature-Adaptive Sampling）以每个点的局部曲率估计为权重，曲率大的区域（边缘、角点）以更高概率被采样。

曲率估计：对点 p_i 的 k-邻域 $\mathcal{N}_k(p_i)$ 构造协方差矩阵 $C_i = \frac{1}{k}\sum_{j\in\mathcal{N}_k}(p_j-\bar{p})(p_j-\bar{p})^\top$，其特征值 $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ 对应三个主方向。最小特征值对应法向量方向，定义曲率为

$$\kappa_i = \frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3} \tag{73}$$

当 κ_i 接近 0 时为平坦区域；接近 1/3 时为体角点（各向同性）；中间值对应边缘（两个大特征值、一个小特征值）。

采样权重：以 $w_i = \kappa_i^{\alpha}$（$\alpha \geq 1$ 为增强因子）作为采样概率的权重，通过接受-拒绝采样（rejection sampling）或重要性采样（importance sampling）保留 $\hat{n}$ 个点。α 越大，对高曲率区域越集中，平坦区域保留率越低。

局限性：曲率自适应采样的失效并不只来自“对噪声敏感”。更常见的问题是邻域半径一旦选小，曲率估计先被量测噪声主导；半径一旦选大，窄边缘和薄结构又会被邻域平均掉。前一种情况下，高曲率点会被误检为边缘并被过采样，后一种情况下，真正负责约束姿态的拐角先消失，随后 ICP 退化为由大平面主导的配准。

4.2.5 基于 ICP Hessian 的信息矩阵自适应采样（EA2D）

[76] 提出的 EA2D 方法将 ICP 变换估计的 Fisher 信息矩阵直接引入采样决策，是上述 NSS 思想的进一步扩展。

方法：对每个体素 v，计算其内部点对 ICP Hessian 的贡献矩阵 $\mathcal{H}_v$。对 $\mathcal{H}_v$ 做特征分解，得到旋转和平移方向的约束强度向量 $(\mathcal{C}_{ri}, \mathcal{C}_{ti})$。自适应采样率由各方向的可定位性（localizability）加权决定：

$$r_v = \min\left(1,\ \gamma \cdot \frac{\mathcal{C}_{ri} + \mathcal{C}_{ti}}{\max_v(\mathcal{C}_{ri} + \mathcal{C}_{ti})}\right) \tag{74}$$

体素级采样率 r_v 直接反映该区域对当前位姿估计的约束贡献：对 ICP 收敛有利的体素以高采样率保留，冗余区域则大幅稀化。

在 KITTI 和 M2DGR 数据集上，EA2D 将后端运行时间从 95 ms 降至 68 ms [76]。这组结果的条件很明确：它针对城市道路和多源移动平台序列，在 LiDAR SLAM 后端里比较的是固定采样与基于 Hessian 贡献度的体素级采样。该文摘要指出定位精度同步改善，但未在摘要层给出统一的绝对误差表项，因此本节只保留“数据集 + 时间 + 方法机制”这三个可核实量。

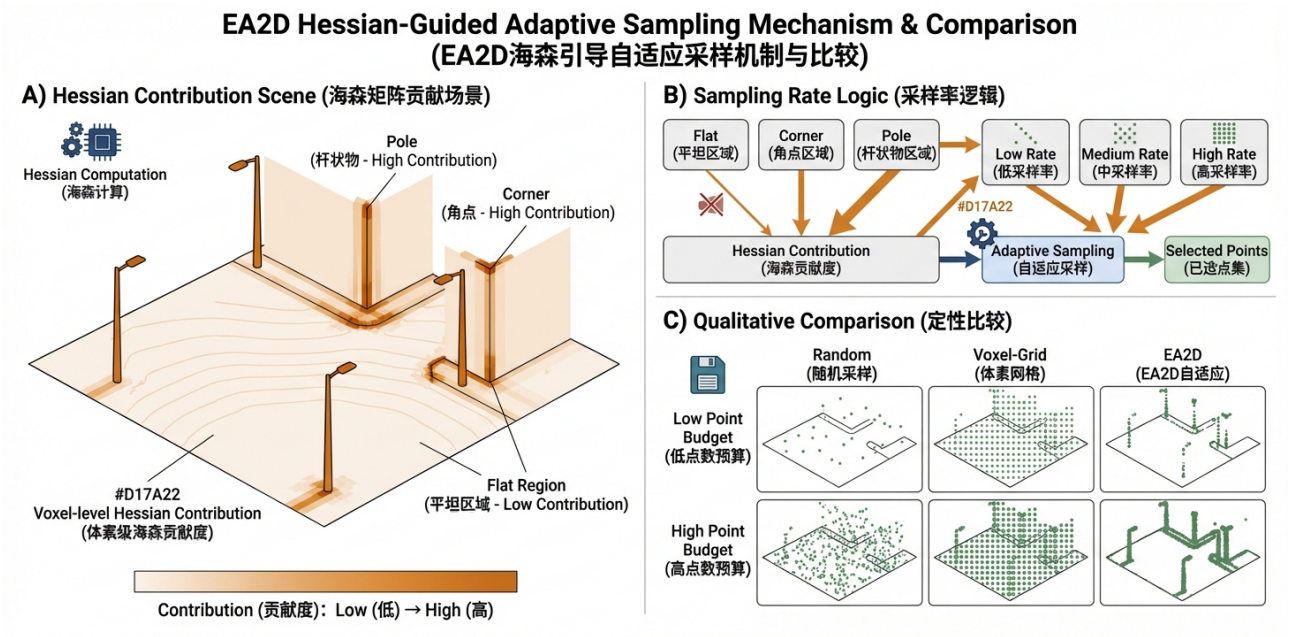


图 44: EA2D 机制示意，不对应原文中的单一数值曲线。图中只表达“按 Hessian 贡献度分配采样率”的思路，以及城市道路场景中高约束区域与低约束区域的差异。

4.2.6 多分辨率与层次化 ICP

多分辨率 ICP（Multi-Resolution ICP 或 Coarse-to-Fine ICP）将降采样推广为一个层次化策略：首先在粗分辨率（大 l，少量点）上快速完成大范围对齐，再逐步精化分辨率（减小 l，增加点数），直至收敛于精配准结果。

金字塔构建：以因子 s（常见设置是 $s = 2$）逐级递减体素尺寸，构建点云金字塔 $\{\mathcal{P}^{(k)}\}_{k=0}^{K}$，其中 $\mathcal{P}^{(0)}$ 为最稀疏层（叶片尺寸 $l_{\max} = s^K l_{\min}$），$\mathcal{P}^{(K)}$ 为原始分辨率或精细层。配准流程：

$$T^{(k+1)} = \mathrm{ICP}\left(T^{(k)},\, \mathcal{P}^{(k+1)},\, \mathcal{Q}^{(k+1)}\right), \quad k = 0, 1, \ldots, K-1 \tag{75}$$

从 $k = 0$ 开始，以单位矩阵为初始位姿，每层 ICP 的输出作为下一层的初始位姿，逐步细化。

理论优势：

- **扩大收敛盆地**：粗分辨率下点间距大，最近邻搜索的匹配半径更宽，对初始误差容忍性更强。
- **避免局部极小**：大尺度几何约束优先对齐主结构（墙面、地板），再由细节约束精修局部特征，减少了精配准阶段的局部极小风险。
- **降低总体计算量**：粗层点数少，每次迭代快；精层仅需少量迭代收敛，总迭代次数比单分辨率少。

实现要点：层数 K 和缩放因子 s 的选择至关重要。[3] 的分析表明，$K = 3$、$s = 2$（即 $1/8 \to 1/4 \to 1/2 \to$ 原始分辨率的四层金字塔）在大多数 LiDAR 场景下可将总配准时间缩短约 60%，同时保持与单分辨率相当的最终精度。

与全局初始化的关系：多分辨率 ICP 并不能替代上一章讨论的全局初始化。粗层仍是局部方法，只是把允许的初始误差范围适度放宽。当初始位姿已经落到错误盆地时，粗层最先失败的仍是对应关系建立，随后误差会被逐层传递到细层，而不是被自动修正。

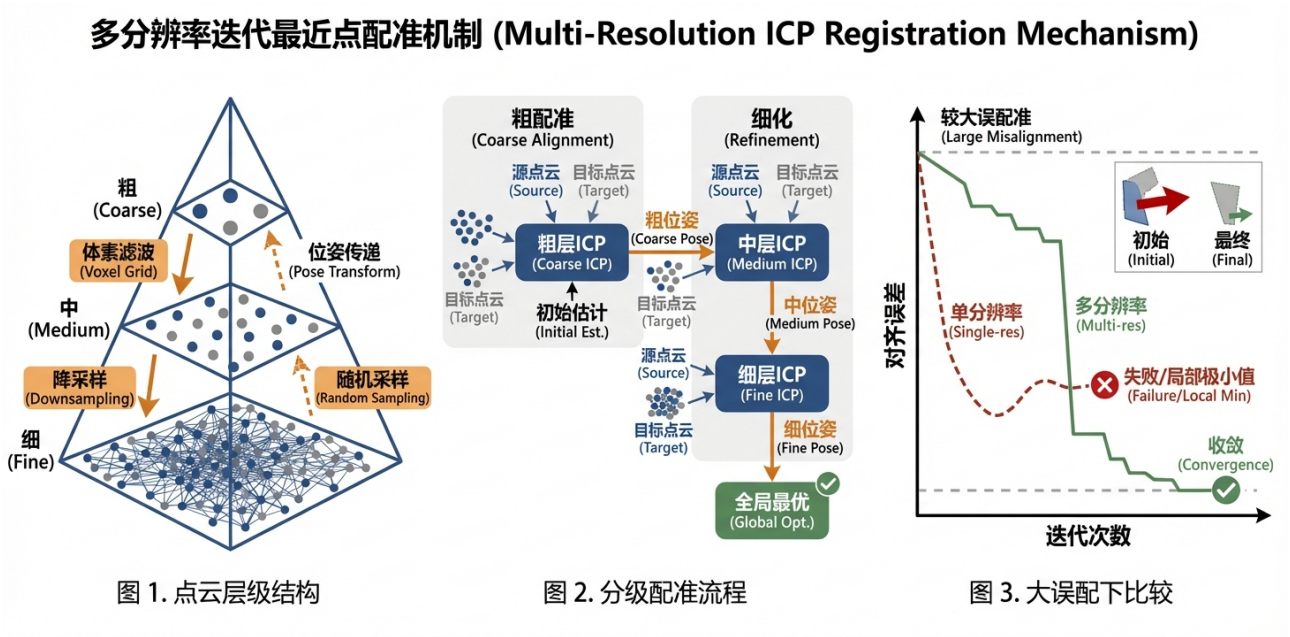


图 45: 多分辨率 ICP 三层金字塔示意。左：粗、中、细三层点云分辨率逐步加密；中：粗层先完成大尺度对齐，细层再做局部修正；右：对比单分辨率与多分辨率在较大初值误差下的收敛差异。该图为机制示意，不对应统一实验数值。

4.2.7 采样策略的定量比较

不同采样策略在速度、精度和鲁棒性三个维度上存在明显差异。以下基于 [23] 的系统评测和近期 LiDAR SLAM 实验结果整理：

工程建议：

- 若场景以平面和少量边缘为主，应先在代表序列上比较随机采样、体素质心与 NSS，因为此时“保留约束方向”比“保留点数”更关键。
- 若系统已经有体素地图与 Hessian 估计，EA2D 这类体素级评分方法更容易落地；它复用现有后端量，而不是额外引入一套曲率特征工程。

表 21: 降采样策略对比：只保留本节已核实的场景、指标和结果；没有统一原文表项的数据不写成伪精确数值。

策略	代表场景/数据	指标与已核实结果	主要收益	先失效的环节
均匀随机采样	合成或密度较均匀场景	当前章节引用文献未给出统一数值基线	实现最简单，便于设定速度上界	特征稀薄区域先被抽空
体素质心	[23] 的六类真实场景协议	该协议证实采样与误差模型联动影响 e_r、e_t，但不支持单一叶片尺寸常数	兼顾规模压缩与空间均匀性	体素尺寸过大时边缘先被平均
NSS	“nearly-flat meshes with small features” [13]	论文强调 uniform sampling + NSS 在该类场景中带来最快稳定收敛	优先保留破除退化的小特征	法向量估计不稳时先误分箱
几何稳定采样	刻槽平面、球面、Forma Urbis Romae [83]	条件数从 66.1 降到 3.7	直接针对位姿不确定性采样	先受协方差估计与初值质量影响
曲率自适应	离线高精度建图	当前章节引用文献未给出统一跨数据集表格	保留局部尖锐结构	曲率半径设错时先误判边缘
EA2D（ICP Hessian）	KITTI、M2DGR [76]	后端时间 95 ms → 68 ms	用更少点维持主要位姿约束	Hessian 估计失真时先错配体素权重
多分辨率	粗到细配准流程	当前章节引用文献未给出统一跨数据集单表数字	扩大局部法的可接受初值范围	粗层对应一旦错误会逐层传递

- 若部署预算只允许极轻量前处理，体素质心仍是稳妥起点，但需要通过误差曲线确认边缘是否被提前抹平。

降采样的作用不是孤立地“减点”，而是把计算预算重新分配给更有约束力的点。它确实能显著降低后续计算负担，但不会消除 ICP 的核心瓶颈：剩余点仍要进入对应搜索与线性化求解。因此，下一个问题不再是“删哪些点”，而是“剩下的点怎样更快地完成最近邻搜索和矩阵累积”。

4.3 并行化与向量化加速 (Parallelism and Vectorization)

单线程 ICP 在现代处理器上面临明确的结构性瓶颈：Besl-McKay 算法在每次迭代中都要对源点云 $\mathcal{P}$ 的每个点独立查询对应点，这些查询彼此无数据依赖，因而天然适合并行；但另一方面，最近邻搜索又高度依赖共享空间索引，变换估计则必须等待全部对应关系汇总后才能更新。因此，并行化不是简单地“多开线程”，而是要先判断瓶颈发生在叶节点计算、任务分配还是访存通路上。第 4.1 节 已说明索引结构会改变访存模式，本节据此分层讨论 SIMD 向量化、多线程并行（OpenMP）和 GPU 大规模并行三类路径。

4.3.1 SIMD 向量化：距离计算的微观并行

现代 x86-64 CPU 提供 SIMD（Single Instruction Multiple Data）扩展——SSE4（128 bit，4 个单精度浮点数/周期）、AVX2（256 bit，8 个单精度/周期）、AVX-512（512 bit，16 个单精度/周期）。ICP 中计算量最大的原子操作是欧氏距离的平方：

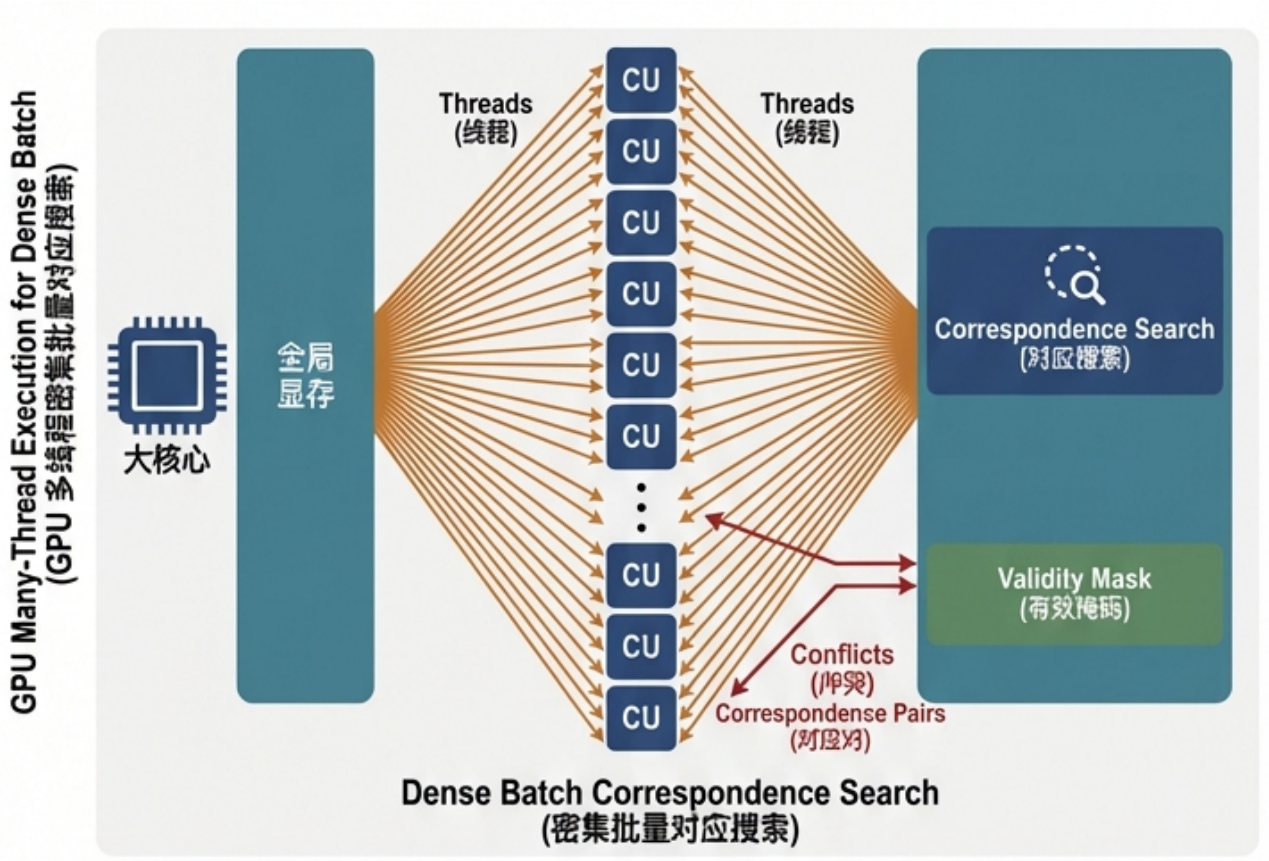
$$d^2(p,q) = (p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2 \tag{76}$$

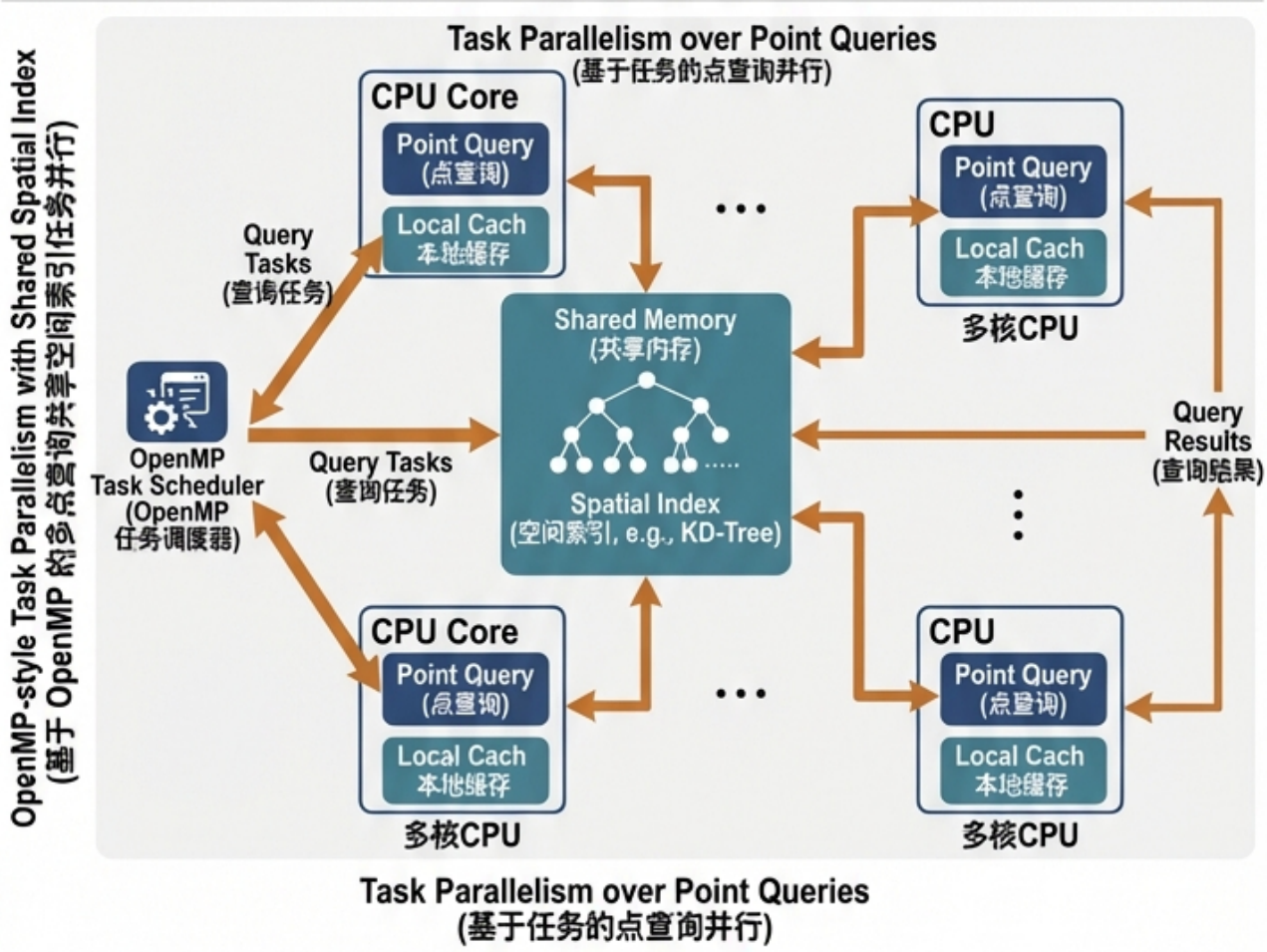
利用 AVX2，可以一次处理 8 对点的距离计算：

```
// SIMD 欧氏距离批量计算 (Batch Euclidean distance with AVX2)
// 输入: 8 个查询点 px/py/pz 和 8 个候选点 qx/qy/qz (packed as __m256)
// 输出: 8 个距离平方 dist2 (vector)
__m256 dx = _mm256_sub_ps(px, qx);   // dx = p_x - q_x for 8 pairs
__m256 dy = _mm256_sub_ps(py, qy);   // dy = p_y - q_y for 8 pairs
__m256 dz = _mm256_sub_ps(pz, qz);   // dz = p_z - q_z for 8 pairs
__m256 dist2 = _mm256_fmadd_ps(dx, dx,
                 _mm256_fmadd_ps(dy, dy,
                 _mm256_mul_ps(dz, dz)));   // dist2 = dx²+dy²+dz²
```

叶节点批量处理：KD-Tree 的叶节点常包含一小批连续存储的候选点，这正是 SIMD 最适合介入的位置。原因不在于 SIMD 会改变搜索路径，而在于叶节点内部的距离计算是规则、重复、彼此独立的。只要内存布局采用 SoA（Structure of Arrays），同一条向量指令就能并行处理一组候选；若仍使用 AoS，访存跨步会先破坏吞吐，再谈不上算力利用。

nanoflann（一个现代 C++ KD-Tree 库）的实现即强调这一设计：其叶节点更容易整理成适合 SIMD 的连续数组。由此可以看出，SIMD 是否见效首先取决于数据布局，而不是指令集名称本身。若叶节点很小、候选点离散分布，向量装载和对齐开销就会抵消收益。


GPU Many-Thread Execution for Dense Batch
(GPU 多线程密集批量对应搜索)
大核心
全局显存
Threads
(线程)
CU
Threads
(线程)
Correspondence Search
(对应搜索)
Validity Mask
(有效掩码)
Conflicts
(冲突)
Correspondense Pairs
(对应对)
Dense Batch Correspondence Search
(密集批量对应搜索)


OpenMP-style Task Parallelism with Shared Spatial Index
(基于 OpenMP 的多点查询共享空间索引任务并行)
Task Parallelism over Point Queries
(基于任务的点查询并行)
CPU Core
Point Query
(点查询)
Local Cach
本地缓存
CPU
多核CPU
Query
Tasks
(查询任务)
OpenMP
Task Scheduler
(OpenMP
任务调度器)
Query Tasks
(查询任务)
Shared Memory
(共享内存)
Spatial Index
(空间索引, e.g., KD-Tree)
Query
Results
(查询结果)
Task Parallelism over Point Queries
(基于任务的点查询并行)


CPU Core
SIMD Vector Unit
(SIMD 向量单元)
Vector Lanes


Preferred Parallelism Level
vs. Point-Cloud Size
(并行度优选级别 vs. 点云规模)
Large (大)
GPU
(Many-Thread)

协方差矩阵累积的 SIMD：SVD 变换估计需要计算 3×3 交叉协方差矩阵 $H = \sum_i (p_i - \bar{p})(q_i - \bar{q})^\top$。这里也能使用 SIMD，但收益一般低于距离搜索阶段。原因是协方差累积虽然同样由大量乘加组成，却只占整轮 ICP 的后半段；若对应搜索仍是随机访存主导，单独优化协方差不能改变总体时延。

4.3.2 OpenMP 多线程：对应搜索的任务并行

每次 ICP 迭代中，源点云 $\mathcal{P}$ 的 n 个点的最近邻查询彼此独立——点 p_i 的最近邻不依赖点 p_j 的结果。这种" 令人尴尬的并行"（embarrassingly parallel）结构使 OpenMP 多线程加速几乎无须修改算法逻辑：

```
// OpenMP 并行最近邻搜索 (Parallel nearest-neighbor search)
#pragma omp parallel for schedule(dynamic, 64)
for (int i = 0; i < n; ++i) {
    size_t idx;
    float dist_sq;
    kdtree.knnSearch(source[i], 1, &idx, &dist_sq);  // 线程安全只读查询
    correspondences[i] = {i, idx, dist_sq};
}
```

线程安全性：KD-Tree 的搜索操作是只读的（不修改树结构），因此多线程并发查询天然是线程安全的。唯一的同步点是结果收集（写入 `correspondences` 数组），由于各线程写入不同下标，无须互斥锁。

调度策略：OpenMP 的 `schedule(dynamic, 64)` 适合负载不均衡的查询，因为不同点的最近邻搜索路径长度可能相差很大。动态调度的价值在于把慢查询分散到不同线程上，避免单个线程在同步屏障前拖住全部核心；但如果点云本身已经均匀、每次查询深度接近，过细的动态分块反而会增加调度开销。

实测加速比：[23] 在真实场景数据集和 4 核平台上比较 ICP 变体时，报告过并行实现相对串行实现约 3.2 倍的总体加速。这个结果发生在多场景、有限核数的条件下，指标是端到端运行时间而不是单个内核函数吞吐，因此更接近工程系统能实际得到的收益。它也提示一个边界：线程数继续增加后，SVD、同步与访存争用会先成为新的瓶颈。

NUMA 感知优化：在多路 CPU 服务器上，跨 NUMA 节点的远程访存会吞噬多线程收益。因此，OpenMP 真正失效的条件并不是“线程太多”，而是线程和数据分布脱节：查询任务虽然被均分，但树节点和点云副本如果跨插槽散落，线程会把时间花在等待内存而不是计算上。

4.3.3 GPU CUDA 并行：大规模稠密点云

当点云规模进入 10^5 量级后，CPU 多线程常常先撞上访存瓶颈而不是算力瓶颈。GPU 的优势正来自这里：它更擅长处理大量结构相似、可批量发射的查询。但这一优势只有在数据关联形式足够规则时才能兑现；若仍沿用分支繁多的树遍历，线程束分歧会先吞掉并行度。

投影式数据关联（Projective Data Association）：[84] 的 GPU 加速重建系统中指出，GPU 上的通用 KD-Tree 最近邻搜索因树形结构的分支预测失败（branch divergence within warps）而效率不高。对 RGB-D/深度图传感器，更高效的方案是" 投影式" 数据关联：将目标点云投影到深度图，再通过像素坐标直接查找对应点，时间复杂度从 $O(n \log n)$ 降至 $O(n)$，且 GPU 访问模式连续（无随机跳转）。但投影式方法依赖传感器视锥，不适用于完全无结构的点云。

GPU 批量 KNN：对于无结构点云，GPU 上的批量 k-NN 实现常用分块策略（tiling）：将目标点云分为大小为 B 的块，每个 CUDA 线程块（thread block）加载一个块到共享内存（shared memory），所有线程并发计算该块内的距离，再全局归约取最小。这样做的理由不是让每个线程独立完成一整次搜索，而是把大量距离计算重写成规则的块内乘加。其成立前提是候选组织足够连续；若候选点需要频繁跨块跳转，shared memory 的复用就会迅速变差。更近的直接证据来自 [85]：该文专门研究 3D 点云配准中的 GPU 最近邻搜索，在 ModelNet40、Stanford Bunny 和 Desk 等数据上，以运行时间和 RMSE 为指标，提出两种基于体素化的近似搜索策略，并报告相对基于 CPU 的 PCL KD-tree，在 RTX 3080 上最高约 5.7 倍、在 RTX 4080 上最高约 12.4 倍的加速，同时保持 10^{-2} 量级的 RMSE。这个结果比系统级重建论文更贴近“对应搜索本体”，也说明 GPU 受益首先来自把全局搜索改写成局部、规则的体素邻域搜索。

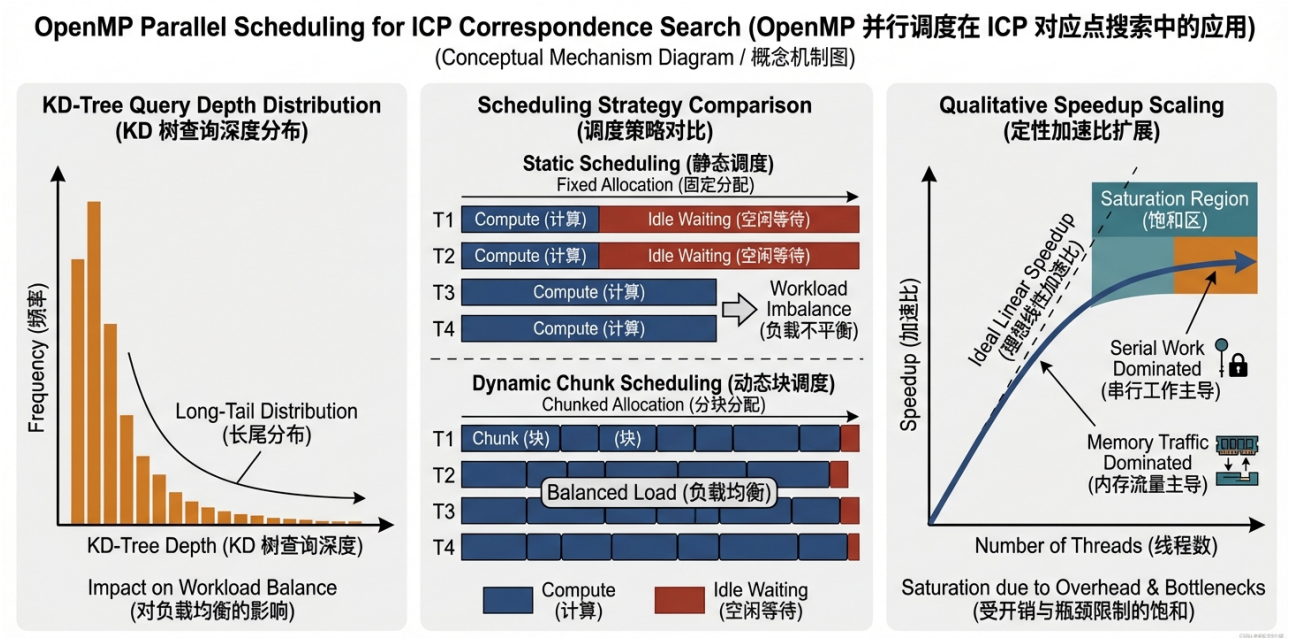


图 47: OpenMP 多线程并行 KNN 搜索的动态调度示意。左：查询深度分布存在长尾时，部分点会明显拉长单次搜索路径；中：静态调度与动态调度的线程时间线对比，说明动态分块可缓解负载不均；右：随着点规模增长，多线程收益会先上升，随后受串行部分和内存带宽约束而趋缓。该图为机制示意，不对应特定平台的统一测量值。

SVD on GPU：变换估计的 3×3 SVD 规模很小，真正耗时的多半不是求解本身，而是为这样一个微型任务单独发起 kernel 和同步数据。因此，异构实现里常见的做法是把大规模对应搜索和规约留在 GPU，把最终的小矩阵求解放回 CPU。这里首先失效的不是数值精度，而是任务粒度与设备启动开销不匹配。

GPU ICP 完整流水线：[84] 在 Open3D 框架中重写了 RGB-D 里程计、Colored ICP、FGR、体素积分与网格提取，并在 TUM RGB-D、Stanford/Redwood 模拟数据以及 Indoor LiDAR-RGBD 数据集上评测。其场景主要是中等规模室内重建，体素尺寸设置为 6 mm 或 8 mm；指标包括重建误差、轨迹一致性与系统吞吐。结果表明，该系统相对原离线重建基线整体提速 10 倍以上，在中等规模室内场景可达到约 8 Hz。这个结果说明 GPU 适合把离线流水线压缩到接近在线速度，但它依赖 RGB-D 投影关联和大批量规约；若换成稀疏、无组织的点云，瓶颈仍会回到对应搜索如何组织的问题。

4.3.4 CPU 与 GPU 的协同：异构并行策略

实际的 ICP 实现中，CPU 和 GPU 各有优势，最优策略多是异构协同而非全部迁移到 GPU：

表 22: ICP 各计算步骤的平台推荐与原因，基于计算模式（规则/不规则）和数据量综合判断。

操作	推荐平台	原因
KD-Tree 构建（目标点云，一次性）	CPU	树形结构不规则，CPU 缓存友好
批量 KNN 查询（$n > 10^4$）	GPU	高吞吐、内存带宽优势
叶节点 SIMD 距离扫描	CPU AVX	连续内存、低延迟
协方差矩阵 H 累积	GPU	数据已在 GPU 显存中，规约高效
3×3 SVD	CPU	计算量极小，避免 kernel launch overhead
法向量计算（仅初始化一次）	CPU + SIMD	简单 PCA，SIMD 并行

数据传输开销：GPU 加速的代价是数据必须在主存和显存之间往返。对大批量点云，这部分代价经常能被并行计算摊薄；但对小规模点云或只迭代极少步的情形，传输和同步会先主导总时延，此时 CPU 端的向量化和

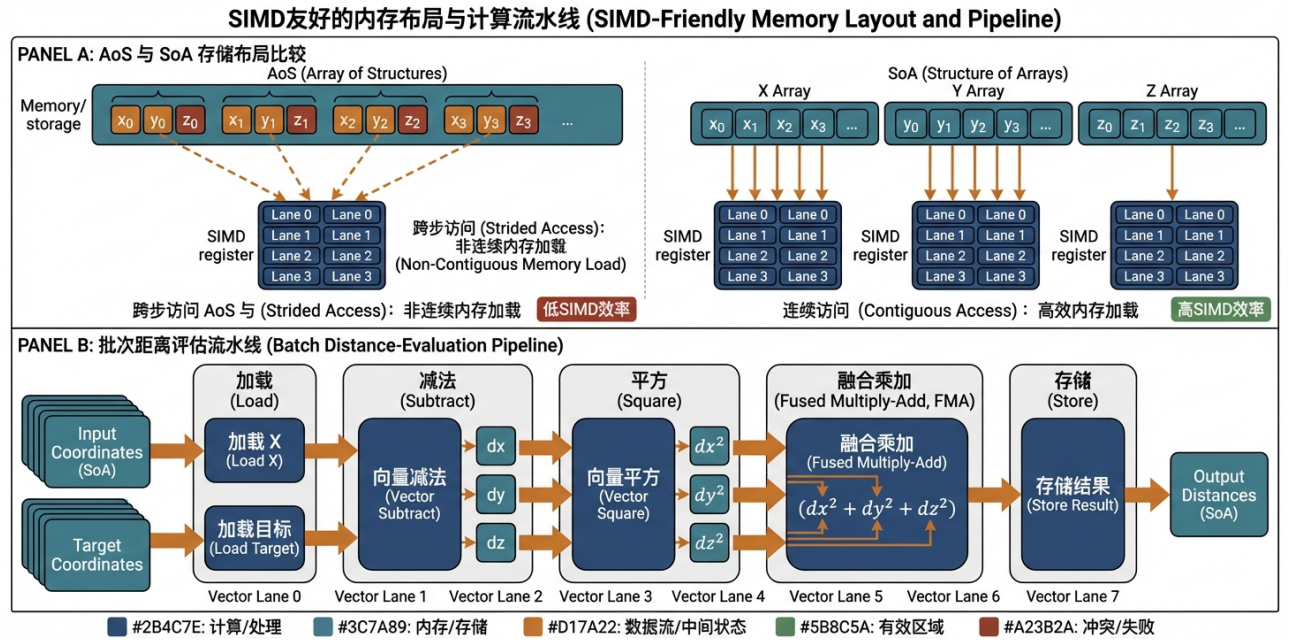


图 48: SIMD 向量化的内存布局关键：AoS（结构体数组）与 SoA（数组结构体）的对比，以及 AVX2 批量距离计算流程示意。上：说明 SoA 更适合连续向量装载；下：展示一次加载、减法、平方和乘加累积的指令流水。该图为机制示意，不对应统一平台上的周期和倍数。

多线程更合适。

4.3.5 混合精度与精度-速度权衡

现代 GPU 的半精度（FP16）吞吐常高于单精度（FP32），因此混合精度经常被作为下一步优化手段。ICP 的核心数值是点坐标和距离，但是否能安全降精度取决于坐标范围是否已经局部化，而不是只看网络训练里常用的数值格式。

- **室内 RGB-D**：若点云先转换到局部坐标系，坐标幅值较小，FP16 常可覆盖对应搜索与残差计算所需的分辨率。
- **室外 LiDAR**：若直接在大范围全局坐标下计算，FP16 的量化误差会先落到远距离点和小残差上，导致最近邻比较和协方差累积不稳定。实践中多先减去局部中心或分块处理，再考虑半精度。

混合精度策略的合理分工一般是：距离计算和候选筛选可以尝试 FP16，而最终规约和位姿更新仍保留 FP32。这样做是为了把吞吐提升限制在数值较稳的阶段；若连小矩阵求解也降到半精度，误差会先在迭代后期积累。

4.3.6 各层次并行加速汇总

软件层面的并行化已经能够明显压缩 ICP 的端到端时延，但它的边界同样清楚：一旦系统受限于访存、同步或设备传输，继续堆线程并不会线性换来收益。因此，第 5 章转向专用硬件并不是简单追求更高峰值吞吐，而是试图把当前仍不规则的数据流重新组织为可持续的片上执行路径。

4.4 近似最近邻与误差容忍性 (Approximate Nearest Neighbor and Error Tolerance)

精确最近邻（Exact NN）是 ICP 收敛分析的理论基础：Besl-McKay 算法的单调递减性证明依赖于对应关系“不差于前一步”。但工程系统很少在理论条件下运行。点数扩大、描述子维度上升或延迟预算压缩时，系统真正面对的是一个折中问题：允许多大近似误差，才能换来可接受的召回率和查询时间。本节因此把 ANN 分成两类讨论，一类是面向原始几何点的低维搜索，另一类是面向 FPFH、FCGF 等特征描述子的高维搜索。

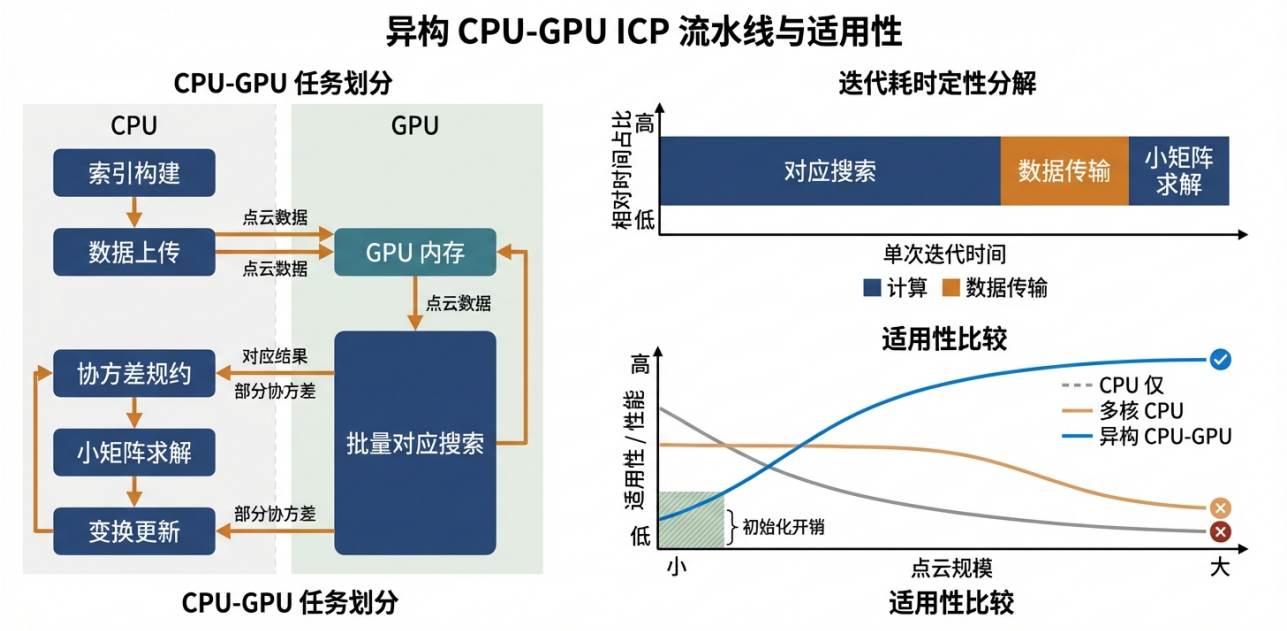


图 49: ICP 计算步骤在 CPU 与 GPU 间的异构协同示意。左：展示索引构建、数据上传、批量查询、协方差规约与小矩阵求解的设备分工；中：说明在一轮迭代中，大规模对应搜索仍是主要耗时部分；右：比较小规模与大规模点云下 CPU、OpenMP 与异构方案的适用边界。该图为机制示意，不对应统一硬件平台上的精确时间值。

表 23: 第 4.3 节 软件并行化各层次技术对比，包括已核实结果或收益形式、适用规模和主要工程代价。

并行层次	技术	已核实结果或收益形式	适用点规模	主要代价
SIMD/AVX2	叶节点批量距离	通过向量化一批候选点，降低叶节点扫描时延	任意，但要求连续内存	代码复杂度，需 SoA 内存布局
SIMD/AVX-512	叶节点批量距离	与 AVX2 同理，但更依赖硬件支持与对齐	任意，但平台受限	指令集可移植性差
OpenMP (T 核)	并发 KNN 查询	[23] 在 4 核平台上报告约 3.2 倍端到端加速	中到大规模点云	负载均衡、NUMA 与同步开销
GPU CUDA	批量 KNN + 规约	[84] 在室内 RGB-D 重建流水线上实现相对离线基线 10 倍以上提速，约 8 Hz	大批量、规则化查询	传输与 kernel 启动开销，依赖查询组织方式
混合精度 FP16	GPU KNN 计算	在局部坐标系下可继续提升吞吐，但收益取决于数值范围	大规模点云	精度损失，需本地坐标系

4.4.1 近似最近邻的形式化定义

给定误差参数 $\epsilon \geq 0$，ϵ-近似最近邻搜索返回点 q_ϵ 满足：

$$\|p - q_\epsilon\| \leq (1 + \epsilon) \cdot \|p - q^*\| \tag{77}$$

其中 $q^* = \arg\min_{q \in \mathcal{Q}} \|p - q\|$ 为精确最近邻。参数 $\epsilon = 0$ 对应精确搜索；$\epsilon = 0.1$ 表示允许返回的对应点距离最多比真实最近邻远 10%。

关键观察：ICP 的收敛性依赖于“每步对应不要把系统推离当前盆地”，而不是要求每个对应都达到数学上的全局最优。近似搜索真正危险的情形，不是某一对点略有偏差，而是错误对应持续集中在同一退化方向，导致线性系统先在某个自由度上失去约束。后文讨论 ANN 时，都围绕这一失效模式展开。

4.4.2 FLANN 与 k-d 树随机近似

FLANN（Fast Library for Approximate Nearest Neighbors）[80] 的核心价值，在于把“索引结构”和“参数选择”都做成可配置问题，而不是预设某一种树永远最优。Muja 和 Lowe 的评测对象主要是随机向量、图像块、SIFT 特征和 80 million tiny images 这类高维数据，指标是查询时间与精度/召回率。因此，把 FLANN 引入 ICP 时，首先要澄清它服务的是哪一层：原始三维点搜索，还是特征描述子搜索。

1. 随机化 KD-Tree 森林（Randomized KD-Tree Forest）：构建 T（常取 4–16）棵独立的 KD-Tree，每棵在非最优轴上以一定概率随机化分裂决策，使同一点在不同树中走到不同叶节点。查询时并行在所有树中搜索，维护一个全局优先队列（priority queue），设置最大检查点数 C（check budget）：当检查总节点数达 C 时提前终止，返回当前最优。

$$\epsilon_{\text{FLANN}} \approx \frac{\text{dist}(p, q_{\text{best}})}{\text{dist}(p, q^*)} - 1, \quad \mathbb{E}[\epsilon_{\text{FLANN}}] \approx 0 \text{ when } C \to \infty \tag{78}$$

C 控制精度-速度权衡：$C = \infty$ 退化为精确搜索；C 越小，越容易在高层节点提前终止。Muja 和 Lowe 的原始结论是，在高维特征数据上，随机化 KD-forest 和 priority search k-means tree 都能在 60% 与 90% 精度区间取得显著加速 [80]。但这组结果不能直接改写成“三维 ICP 的固定最优参数”，因为三维点云的距离分布、重叠率和异常值比例都不同于视觉描述子库。

2. 优先搜索 k-means tree 与其他索引：Muja 和 Lowe 还表明，除随机化 KD-forest 外，priority search k-means tree 在高维特征上同样具有竞争力。对 ICP 而言，这一结果的意义在于：如果对应搜索已经从原始坐标迁移到局部描述子，索引选择就不应再沿用三维几何点的经验。此时真正需要核验的是 recall 曲线和下游配准误差，而不是单独比较索引吞吐量。

4.4.3 HNSW：小世界图的近似搜索

分层可导航小世界图（Hierarchical Navigable Small World，HNSW）[77] 是近年应用最广的通用 ANN 结构之一。它的重要性不在于“已经证明最适合 ICP”，而在于它把近似搜索从树结构推进到了分层图结构，为学习型描述子配准提供了更稳定的召回率-时延折中。

核心思想：在多个层次上构建连接图，高层（少量点，长距离边）负责粗定位，低层（全量点，短距离边）负责精细定位。查询从最高层入口点出发，贪心地沿边移动到更近的邻居，逐层向下直至底层，路径长度期望为 $O(\log n)$。

原始证据与外推边界：Malkov 和 Yashunin 的实验对象是 SIFT、GloVe、MNIST 和随机向量，指标是 recall 与查询时间，结论是 HNSW 在这些数据上优于 NSW、FLANN、Annoy 等开源索引 [77]。因此，本节只把 HNSW 作为“高维特征对应搜索”的桥接证据，而不把它直接写成原始三维 ICP 的实测结论。若任务仍是纯 xyz 最近邻，是否值得从 KD-Tree 切到 HNSW，必须重新在同一重叠率和噪声条件下验证。

与三维点云 ANN 的区别：这一点也可以和 [85] 对照来看。Chang 的对象就是三维点云配准中的近似最近邻搜索，方法核心是体素化后把全局搜索改成局部搜索；HNSW 则面向高维向量索引，优势来自图导航和高召回率。因此，这两类方法不应混成同一条技术线：前者更接近几何点搜索的工程重写，后者更适合学习型描述子或大规模特征库。

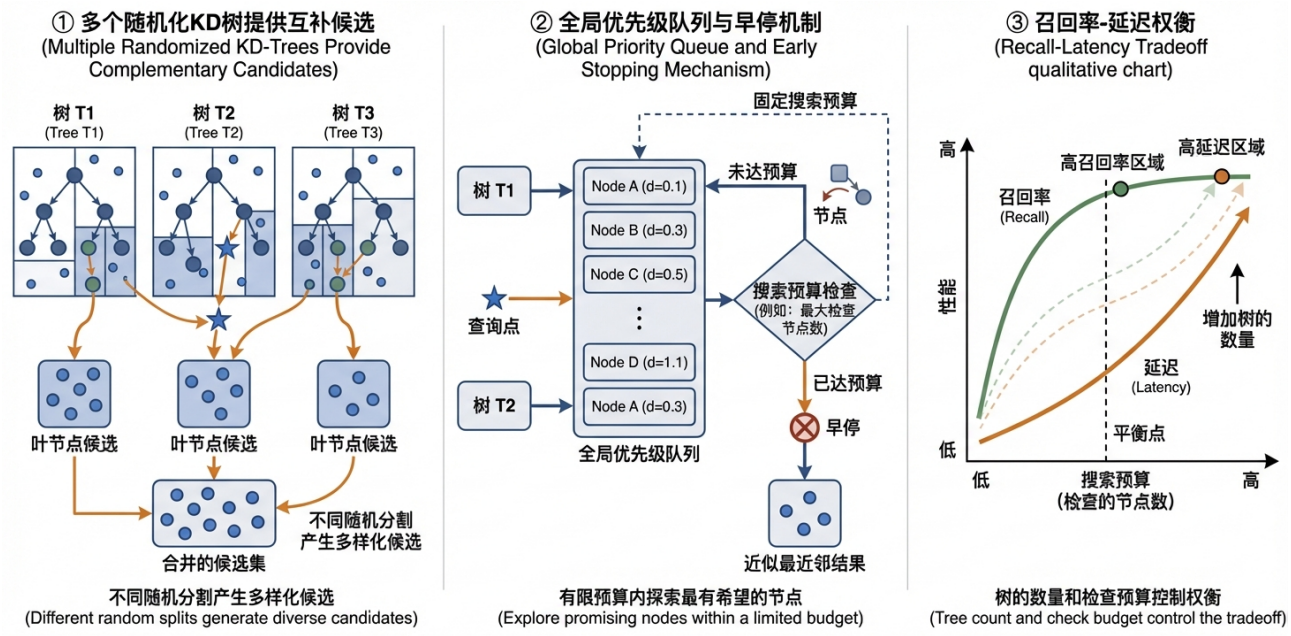


图 50: FLANN 随机化 KD-Tree 森林与优先队列搜索机制示意。左：多棵随机化树为同一查询提供互补候选；中：优先队列控制早停；右：树数与检查预算共同决定精度和时延。该图为机制示意，不对应单一数据集上的固定最优参数。

硬件加速 ANN：[86] 提出将 GPU 的光线追踪单元（Ray-Tracing Unit）扩展为通用层级搜索单元（Hierarchical Search Unit，HSU），将 HNSW 的图遍历映射到 BVH（Bounding Volume Hierarchy）遍历硬件，实现平均 24.8% 的额外性能提升（相比软件 ANN 实现）。这一思路表明，未来 GPU 的专用图搜索硬件可能成为大规模点云配准的关键加速器。

4.4.4 ICP 对 ANN 误差的容忍机制

ICP 框架内置了若干机制，使其对近似最近邻误差具有天然的容忍性：

迭代修正（Iterative Correction）：即使某次迭代的对应关系因 ANN 误差而轻微偏差，下次迭代会以更新后的变换重新搜索，有机会修正之前的错误对应。ANN 误差相当于给对应关系增加了噪声，只要噪声强度低于 ICP 的收敛盆地半径，迭代过程仍能收敛到接近精确解的位置。

拒绝准则的缓冲：第 3 章讨论过的距离阈值拒绝会先滤除最差对应。ANN 真正影响的是阈值边界附近的候选：若近似误差只让候选在同一局部邻域内互换，变换估计仍可能继续下降；若近似误差把候选推到另一个表面片段，阈值拒绝和法向量一致性检查就会同时失效，随后才表现为整体发散。

实验边界：Pomerleau 的六场景协议证明了同一配准框架内可以系统比较不同模块，但该文公开摘要里的可核实基线主要聚焦点到点与点到面，而不是 ANN 参数扫描 [23]。因此，本节不把“某个 ϵ 或某个 check budget 对所有 ICP 都安全”写成定论。更稳妥的结论是：ANN 只有在误差分布不沿退化方向累积时，才会表现为可接受的近似噪声。

4.4.5 实际应用中的 ANN 配置策略

在实际 ICP 系统中，ANN 的配置应沿着“数据类型”而不是“论文热度”来选：

- 对原始三维点的低维搜索，先验证精确 KD-Tree 是否已经满足时延预算。只有当点数规模、动态更新或并行化需求让精确树不可接受时，再考虑近似预算。
- 对 FPFH、FCGF、学习型局部描述子等高维特征，优先参考 FLANN 与 HNSW 的原始评测域，因为这类任务和 SIFT、GloVe 等向量检索更接近。

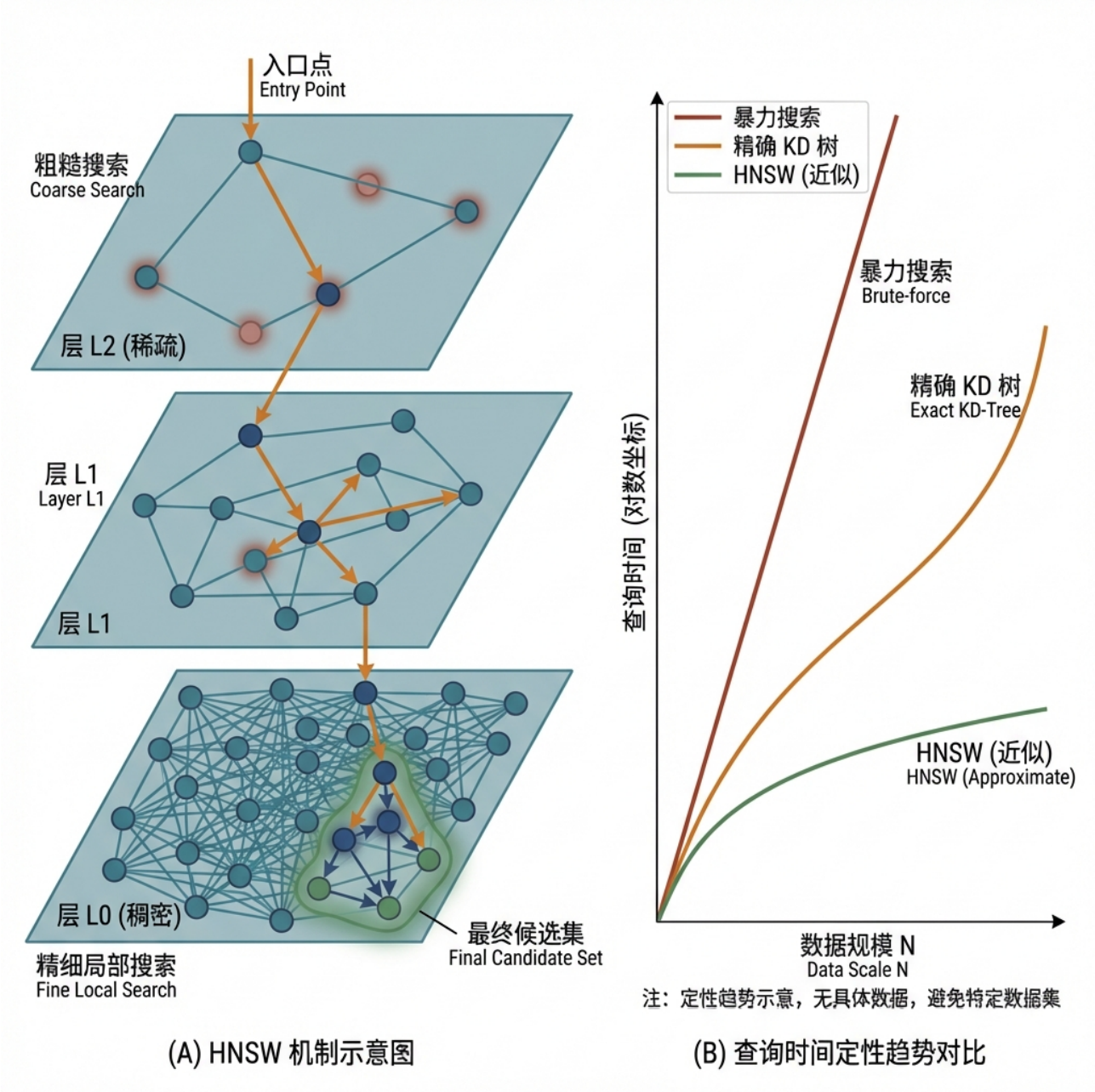


图 51: HNSW 机制示意, 不对应 ICP 论文中的单一测试曲线。图中只说明分层图的粗到细搜索路径, 以及 HNSW 与树结构在高维索引中的典型差异。

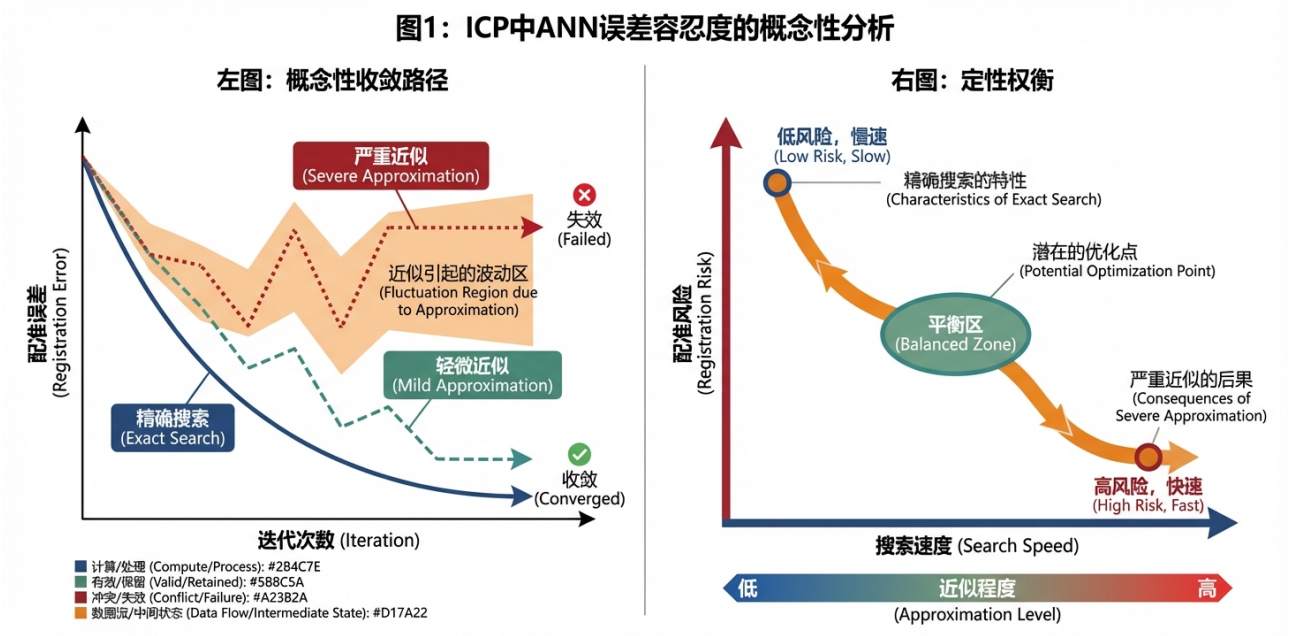


图 52: ANN 误差容忍性的机制示意，不对应单一数据集的实测数值。图中只表达“误差小而分散”与“误差沿退化方向累积”两种截然不同的后果。

- 如果系统需要跨 CPU/GPU 或跨节点分布式索引，应优先比较 recall 曲线与下游配准误差，而不是孤立比较单次查询吞吐量。

4.4.6 近似最近邻综合对比

表 24: 近似最近邻方法对比：只保留本节已经核实的原始验证域与结论，不把高维向量检索结果误写成原始点云 ICP 的直接实验。

方法	原始验证域	评价指标	已核实结论	对 ICP 的适用边界
精确 KD-Tree	原始几何点	距离误差、查询时间	无近似误差，适合作为基线	低维点云首选对照组
FLANN 随机 KD-forest	随机向量、图像块、SIFT、tiny images [80]	精度/召回率、查询时间	在 60% 与 90% 精度区间都可显著加速	更适合高维描述子而非直接照搬到 xyz 点
FLANN priority k-means tree	同上 [80]	精度/召回率、查询时间	与随机 KD-forest 共同构成最优候选	适合特征索引，不宜直接当作低维几何默认值
HNSW	SIFT、GloVe、MNIST、随机向量 [77]	Recall、查询时间	优于 NSW、FLANN、Annoy 等开源索引	更适合作为高维描述子检索桥接方案

ANN 的价值，不是把所有对应搜索都改成“越近似越快”，而是让系统在给定召回率目标下选择合适的索引结构。对原始低维点云，近似预算必须服从几何退化与异常值分布；对高维描述子，FLANN 和 HNSW 提供了更成熟的召回率-时延证据。二者共同说明，近似搜索只有放回具体数据分布与目标函数里讨论，才有意义。

然而，软件层面的优化终究受限于通用处理器的访存与能耗约束。下一章将把问题从“如何选索引”推进到“如何为索引和搜索流程设计硬件执行路径”，也就是专用加速器为何能继续缩短对应搜索延迟。

4.5 本章小结

本章的结论可以归纳为两点。其一，软件加速首先是“约束重分配”而不是“单纯提速”。第 4.1 节 说明索引结构必须与地图更新频率一起考虑：FAST-LIO2 之所以能在 19 个公开序列上稳定运行，并在大场景达到 100 Hz、在 1000 deg/s 旋转条件下保持估计，是因为它把直接配准、增量更新和树上降采样放进了同一个 ikd-Tree 框架 [42]。如果场景持续变化而索引仍依赖整树重建，那么查询开销还未成为瓶颈之前，索引维护就会先失控。其二，采样策略只有在“保留约束”这一条件下才成立。Pomerleau 等在六类真实场景上的评测给出了一个直接对照：点到面 ICP 的精度高于点到点约 20%–40%，但点到点速度约快 80%，说明减少计算量与保留有效几何

约束始终处于拉扯关系 [23]。Gelfand 等提出的几何稳定采样进一步把这种关系写成了条件数优化问题：当采样集中在特征贫乏区域时，先退化的是姿态约束矩阵，后果是收敛变慢甚至滑移到错误姿态 [83]。

降采样、并行化和 ANN 的作用边界也应分开看待。EA2D-LSLAM 在 KITTI 与 M2DGR 上把后端时间从 95 ms 压到 68 ms，是因为它用 Hessian 分解估计体素对位姿约束的贡献，而不是均匀删点 [76]。近似搜索同样依赖误差预算是否可控：第 4.4 节 中的 FLANN 适合低维点云或中等维度描述子，HNSW 则补上了高召回图索引这一环节，但其原始实验对象是 SIFT、GloVe、MNIST 等向量检索任务，因此迁移到 ICP 时仍需重新核实三维点云上的召回率与延迟 [77]。因此，本章更稳妥的结论是：软件优化已经给出了实时 ICP 的主要方法库，但每一类方法都绑定了明确的场景条件。下一章讨论硬件加速时，应把这些条件带入架构设计，而不是把软件测得的速度直接外推到 FPGA 或 ASIC。

5. 硬件加速：FPGA、ASIC 与近存储计算

第 4 章已经说明，软件侧的并行化、数据结构重排和近似搜索能够明显压缩 ICP 延迟，但瓶颈并未消失，而是集中暴露在最近邻搜索的数据访问阶段。以自动驾驶 LiDAR 配准为例，[78] 在 KITTI 上比较多种设计点后指出，KD-Tree 搜索始终主导运行时间；其专用处理器 Tigris 在同一任务上将 KD-Tree 子过程相对 RTX 2080 Ti 的速度提升到 77.2 倍，同时把功耗降为后者的约 1/7.4。这一现象并不限于 ASIC：面向高吞吐点云 kNN 的 HBM-FPGA 原型 ParallelNN 在 KITTI 上相对 CPU 和 GPU 分别达到 107.7 倍和 12.1 倍加速，能效增益分别达到 73.6 倍和 31.1 倍 [87]。这些结果共同指向同一问题：当访问模式仍由随机树遍历主导时，性能上限更多受片外带宽和数据搬运限制，而不是受浮点算力限制。

因此，本章关注的不是“把同一套软件再移植一遍”，而是分析不同硬件路线如何改写最近邻搜索的实现前提。FPGA 路线通过重排数据组织，把随机访问改成适合流水线的规则访问：RPS-ICP 面向有组织车载 LiDAR 点云，在对应搜索阶段达到 18.6 FPS，对应搜索速度比既有 FPGA 实现快 13.7 倍，能效比 GPU 高 50.7 倍 [88]；HA-BFNN-ICP 在 3.4 W 功耗约束下，对 3D LiDAR 建图任务取得相对 CPU 17.36 倍加速 [8]。ASIC 路线则把 KD-Tree 搜索和小矩阵算子直接固化到专用数据通路中，以换取更低的控制开销和更高的单位能效 [78]。PIM 路线继续向存储侧推进：PICK 在 KITTI、S3DIS、DALES 等数据集上的 kNN 搜索，相对此前设计实现 4.17 倍加速和 4.42 倍节能 [20]；C^2IM-NN 在 28 nm CMOS 上把 CAM 搜索与 1D-CNN 预测结合，报告了 23.08 倍能效提升和 48.4% 存储占用下降 [89]。这些工作之间的差别，不只在器件类型，还在于它们分别接受了哪些算法改写、数据格式约束和精度折中。

基于上述观察，本章按“瓶颈分析、平台路线、协同方法”三层展开。第 5.1 节 首先从最近邻搜索的访存特征出发，量化通用处理器在带宽、缓存命中率和能效上的限制，并建立硬件设计空间。第 5.2 节 讨论 FPGA 如何通过流式缓存、规则化搜索结构和运行时可重配置机制压缩延迟，但也指出这一路线对点云组织形式、片上存储容量和定点量化较为敏感。第 5.3 节 分析专用处理器怎样用固定数据通路换取极低延迟，同时说明算法一旦切换，其硬件复用空间会迅速收缩。第 5.4 节 进一步讨论把距离计算和 Top-k 维护推入存储阵列后的收益与代价，重点说明哪一类带宽瓶颈可以被消除、哪一类可靠性和工艺问题会先暴露。第 5.5 节 最后回到算法-硬件协同设计，整理数据结构替换、低精度表示和固定时延执行三类共设计策略，并据此归纳后续系统选型所需的判断标准。

5.1 硬件加速的动机与设计空间 (Motivation and Design Space for Hardware Acceleration)

第 4 章 已经表明，软件优化能够降低 ICP 的平均延迟，但瓶颈仍集中在最近邻搜索的数据访问阶段。对自动驾驶和机器人点云任务而言，这一瓶颈并不是抽象判断，而是多篇硬件论文反复验证的共识：[78] 在点云配准设计空间探索中指出，KD-Tree 搜索在不同实现之间都占据主导时间；[87] 进一步指出，已有加速器即使引入并行计算单元，仍会被外部 DDR 带宽限制；[20] 则把点云 kNN 的主要困难概括为“计算强度高且内存访问开销重”。因此，第五章讨论硬件加速的出发点不是单纯追求更高算力，而是分析不同硬件路线如何处理随机访存、片外带宽和数据搬运。

本节按这一逻辑展开。首先说明通用处理器为什么难以持续压缩最近邻搜索延迟；随后总结 ICP 各子步骤

中哪些部分值得硬件投入；再将设计空间拆成数值精度、器件形态和算法共设计三个维度；最后讨论单模块加速为何不必然转化为系统级收益，为后续第 5.2 节 、第 5.3 节 、第 5.4 节 和第 5.5 节 奠定比较框架。

5.1.1 通用处理器的性能瓶颈

ICP 算法的计算结构与通用 CPU 的设计假设之间存在根本性不匹配，这种不匹配从三个方向共同制约了通用平台在点云配准任务上的性能上限。

首先是内存访问不规则的问题。KD-Tree、Octree 和球形桶的查询都包含条件分支与层次遍历，CPU 和 GPU 虽然可以并行执行距离计算，却难以规整树遍历的访问序列，因此缓存复用和预取收益有限。[78] 之所以把 KD-Tree 搜索作为 Tigris 的首要加速对象，正是因为不同精度和性能折中下，这一步始终压过配准流水线的其他阶段；[87] 对八叉树 kNN 的分析也得到相同结论：若搜索仍依赖片外 DDR，内部并行度提升会很快受外部带宽约束。

与访问模式不规则相伴的是带宽瓶颈。点云对应搜索涉及大量” 取节点、比较、更新候选集” 的短计算链路，对这类负载而言，外部存储的访问时延往往比单次距离计算更难压缩，因此硬件论文普遍把高带宽缓存、片上 SRAM 或 PIM 作为核心设计点，而不是继续增加浮点单元数量。[87] 使用 HBM 和片上多通道缓存提升可用带宽；[20] 则进一步把距离计算和 Top-k 维护下推到存储阵列内部，以消除运行时片外访问。

功耗差距则主要来自控制与数据搬运的累积开销。通用处理器必须同时服务多类程序，控制逻辑、缓存层级和一致性开销无法为 ICP 单独裁剪；专用加速器只保留最近邻搜索和小矩阵运算需要的数据通路，因而更容易把功耗集中到” 确实在做几何计算” 的部分。[90] 在 Zynq-7000 上实现的可重配置定位加速器，相对 Intel 和 Arm CPU 分别达到 59.1 倍和 9.2 倍加速，并通过运行时配置把平均能耗再降约 18%；[78] 的 Tigris 在 KD-Tree 搜索上相对 RTX 2080 Ti 达到 77.2 倍加速，同时把功耗降至后者的约 1/7.4。这些结果说明，硬件收益并不只来自” 算得更快”，还来自” 少搬数据、少走控制流”。

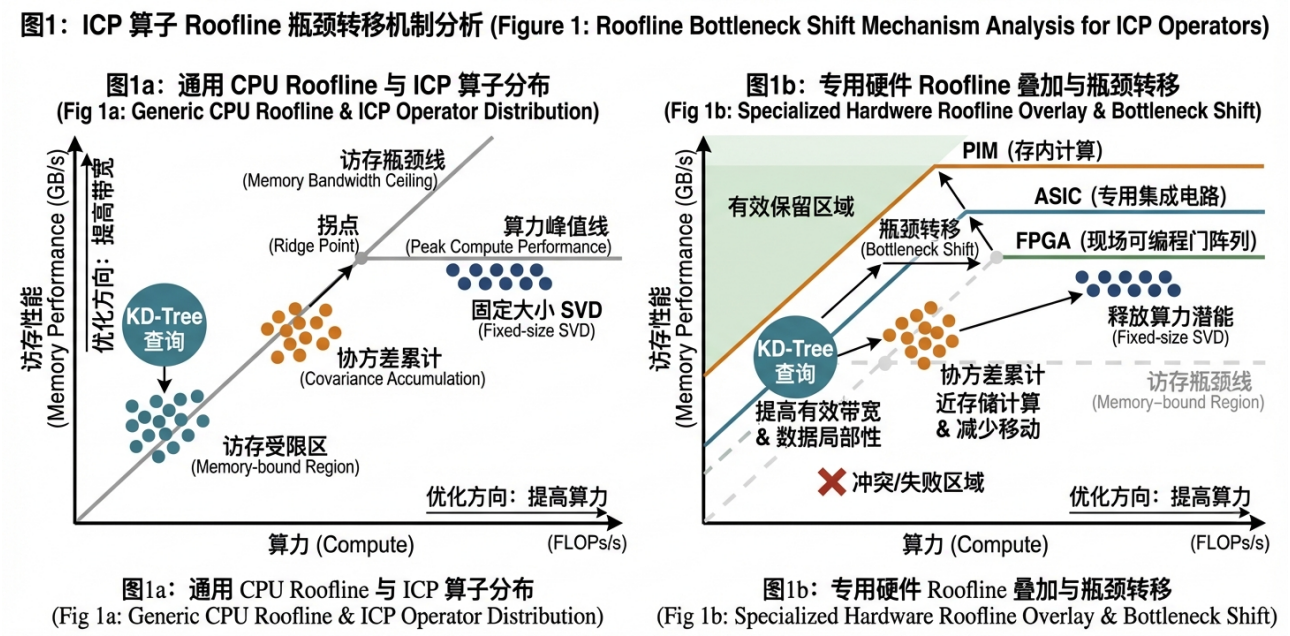


图 53: ICP 各算子在 Roofline 模型下的受限关系机制示意。该图用于说明最近邻搜索更容易受带宽限制、而固定规模 SVD 更接近计算受限，不对应单一论文中的精确测试数值。

5.1.2 ICP 的计算热点分析

理解硬件设计空间的前提，是先确定哪一类算子值得专门映射到硬件。现有论文虽然采用的数据结构和平台不同，但结论相对一致：最近邻搜索是第一瓶颈，协方差累积和小矩阵求解属于第二层瓶颈。[78] 围绕 KD-Tree 搜索展开专用处理器设计；[88]、[7]、[87] 和 [20] 也都把对应搜索或 kNN 查询置于加速核心。相比之下，3×3

SVD 与位姿更新虽然不可缺少，但计算规模固定、数据重用高，在大多数实现里都不是决定端到端时延的首要因素。

图图 5-2 54给出这种“主瓶颈与次瓶颈”关系的机制示意：

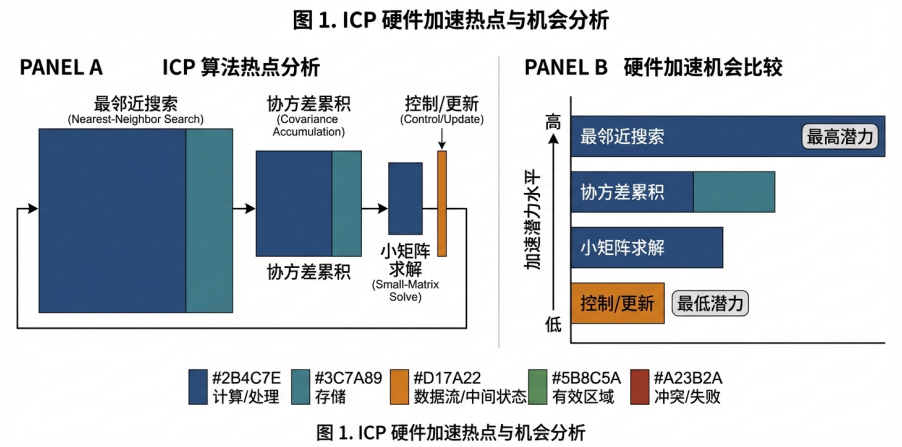


图 54: ICP 计算热点与硬件加速机会示意图。该图用于说明“对应搜索优先、规约次之、固定小矩阵再次之”的相对关系，不对应单一实验中的精确占比。

在三类子步骤中，最近邻搜索中每个源点的查询彼此独立，理论上有很高并行度，但树遍历、候选更新和片外取数交织在一起，使这一步最容易暴露存储系统瓶颈，这也是 FPGA、ASIC 和 PIM 三条路线都把对应搜索作为第一优先级的根本原因。协方差矩阵构建与规约的数据访问相对规则，适合映射到 SIMD、流水线或树形规约网络；若系统目标是覆盖完整 ICP 流水线而不仅仅是 kNN 内核，加速器还需要解决这一部分，否则对应搜索被压缩后，规约会成为新的前端瓶颈。SVD 与位姿更新的矩阵维度固定，便于做定长数据通路或微码优化，但其收益更多体现在稳定延迟和流水线闭环，而不是绝对吞吐量。这一分解意味着，硬件方案若不先解释如何处理对应搜索，就难以在系统层面给出可信的收益。

5.1.3 硬件设计空间的三个维度

面向 ICP 的硬件加速器设计涉及三个相互制衡的正交维度，理解这三个维度有助于在不同器件形态和算法约束之间做出有据可查的选择。

第一个维度是精度与性能之间的权衡。全精度浮点实现保留了较宽的动态范围，适合直接复用现有软件算子，但面积和功耗开销最高。半精度或定点实现能够显著降低乘加阵列成本，因此在 FPGA 和 PIM 设计中更常见；代价是必须重新验证量化误差对配准精度和收敛稳定性的影响。[90] 明确把定点运算作为节省资源的核心手段之一，[20] 则进一步把位宽裁剪作为性能与精度之间的运行时折中。

第二个维度是灵活性与效率之间的权衡。FPGA 允许研究者快速验证新的搜索结构和流水线组织，因此 RPS、HA-BFNN、多模式对应搜索等方案都首先落在可重配置平台上 [88][8][91]。ASIC 将配置自由度换成更稳定的时延和更高的单位能效，适合算法路径已经相对固定的场景，例如 Tigris 面向点云感知的专用数据通路 [78]。PIM 不再把” 更强计算核心” 当成唯一解法，而是把距离计算与候选维护尽量推近存储阵列，以减轻总线传输压力 [20][89]。

第三个维度是算法与硬件之间的协同设计程度。纯软件优化主要在既有算法框架内压低常数项；协同设计则允许研究者改写数据结构、搜索策略和数值表示，以换取硬件友好的访问模式。Tigris 通过两阶段 KD-Tree 与近似搜索挖掘查询级和节点级并行 [78]；ParallelNN 通过并行八叉树构建和关键帧调度提升数据复用 [87]；PICK 通过位宽裁剪和两级流水线把 kNN 的距离计算与 Top-k 选择耦合起来 [20]。协同设计的收益更高，但每一次算法改写都需要重新检查误差传播、鲁棒性边界和系统接口。

5.1.4 软硬件之间的系统鸿沟

只看单个内核的峰值吞吐量，容易高估系统级收益。点云 SLAM 或机器人定位流水线至少还包含预处理、特征提取、位姿图优化和地图更新；如果加速器只压缩对应搜索，而 DMA、主控调度或后端求解没有同步调整，端到端时延仍会停在其他模块上。因此，硬件设计除了追求局部加速，还需要回答三个系统层面的问题：与 ICP 共同占据主要时间的模块（如体素哈希更新、法向量计算）是否也需要加速？加速器与处理器之间的接口带宽是否会成为新的瓶颈（如 PCIe 传输延迟）？加速器如何与 SLAM 框架的主控制流集成（中断、DMA、协处理器接口）？只有这三个问题有明确答案，局部模块的峰值加速比才能真正转化为端到端系统收益。

[90] 的可重配置定位加速器正是沿着这一思路设计的：它不是把并行度固定死，而是根据场景中的特征点数量动态调整硬件配置，在维持精度和性能约束的同时把平均能耗再降约 18%。上述结果表明，可部署的加速器需同时满足两类约束：高峰负载下的实时性要求，以及轻载场景下的能耗约束。

表 25: 第 5.1 节代表性硬件工作与已公开结果汇总。表中仅保留论文明确报告的场景、指标和数值，不对不同论文的延迟与功耗做未经统一实验条件校准的横向换算。

代表工作	平台/工艺	任务与场景	指标	已报告结果
Runtime Reconfigurable Localization [90]	Xilinx Zynq-7000 FPGA	KITTI、EuRoC 机器人定位	相对 CPU 加速比、运行时节能	59.1× vs Intel CPU，9.2× vs Arm CPU；动态配置再降约 18% 平均能耗
RPS-ICP [88]	FPGA	有组织 LiDAR 点云配准	帧率、对应搜索速度、能效	18.6 FPS；对应搜索 13.7× 于既有 FPGA；能效 50.7× 于 GPU
ParallelNN [87]	Virtex HBM FPGA	KITTI 点云 kNN	加速比、能效	107.7× vs CPU，12.1× vs GPU；能效 73.6×/31.1×
Tigris [78]	16 nm ASIC	点云配准 KD-Tree 搜索	子过程加速、端到端收益、功耗	KD-Tree 搜索 77.2× vs RTX 2080 Ti；端到端性能提升 41.7%；功耗降 3.0×
PICK [20]	SRAM-PIM	KITTI、SONN、S3DIS、DALES 点云 kNN	加速比、节能	4.17× 加速，4.42× 节能
C^2IM-NN [89]	28 nm CAM-PIM	3D 点云匹配	能效、存储占用	23.08× 能效提升，48.4% 存储占用下降

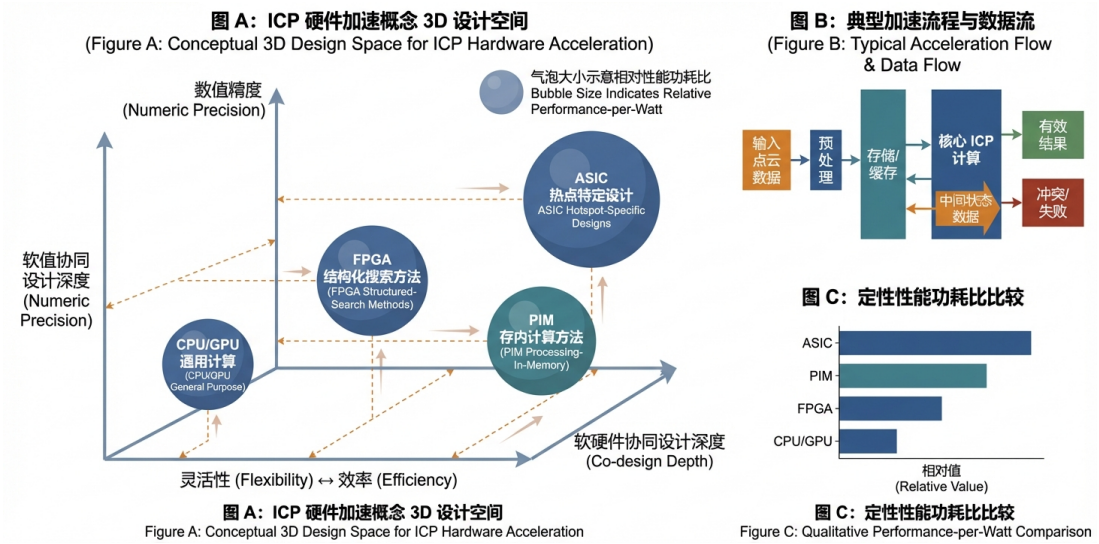


图 55: 面向 ICP 的硬件加速设计空间机制示意图。该图用于展示灵活性、数值表示和协同设计程度三类权衡关系，点位为概念定位，不对应统一基准下的精确坐标。

第 5.2 节 到第 5.4 节 将分别展开 FPGA、ASIC 与 PIM 的具体实现；第 5.5 节 再回到跨平台的共设计方法，讨论哪些算法改写值得为硬件付出复杂度代价。

5.2 FPGA 可重配置加速 (FPGA Reconfigurable Acceleration)

FPGA（Field-Programmable Gate Array）是 ICP 硬件加速的重要中间路线：相比 CPU/GPU，它允许研究者把对应搜索、规约和小矩阵求解组织成固定时序的数据通路；相比 ASIC，它又保留了重新布置搜索结构、缓存层级和并行度的能力，适合算法仍在快速迭代的阶段。

从公开工作看，FPGA 路线的演进大致经历了三个阶段。早期工作多把目标放在单个计算内核，例如小矩阵运算或特定 kNN 电路；随后，研究重点转向对应搜索本身的数据结构改写，例如 RPS、分层图和球形桶，以减少随机访存 [88][7][8]。最近的工作则开始把搜索模块与参数配置、滑动窗口缓存和整条配准流水线一并设计，使同一加速器能够覆盖多种 ICP 变体和场景约束 [91]。

5.2.1 对应搜索的 FPGA 流水线设计

ICP 的 FPGA 加速首先要处理一个结构性矛盾：最近邻搜索希望保留几何邻近关系，但树形索引的访问顺序对流水线并不友好。只要查询仍依赖频繁的片外访问，FPGA 的并行比较单元就难以持续保持满载。因此，许多设计并不直接照搬软件里的 KD-Tree，而是先改写数据组织，再设计查询流水线。

针对有组织 LiDAR 点云（按激光线束排列），[88] 提出了 RPS（Range-Projection-Structure）搜索结构，将激光线束的相似投影位置和距离编码为连续内存索引，使对应搜索变为规则的矩形窗口查询，而非 KD-Tree 的随机跳转：

$$\mathrm{RPS}[r][c] = \text{points with projection}(r, c) \text{ and distance} \approx d_{rc} \tag{79}$$

RPS 的关键不在于引入新的距离度量，而在于把” 先找树节点、再比距离” 的访问顺序改成” 先按扫描线定位候选区域、再在局部窗口中筛选对应点”。这样做的前提是输入点云必须保持有组织的 LiDAR 扫描结构；一旦点云已经被重采样成无组织集合，这一优势就会减弱。Sun 等在车载 LiDAR 配准场景中报告，其 FPGA 框架达到 18.6 FPS，对应搜索加速器比既有 FPGA 实现快 13.7 倍，能效比分别优于 GPU 和既有 FPGA 方案 50.7 倍和 27.4 倍 [88]。

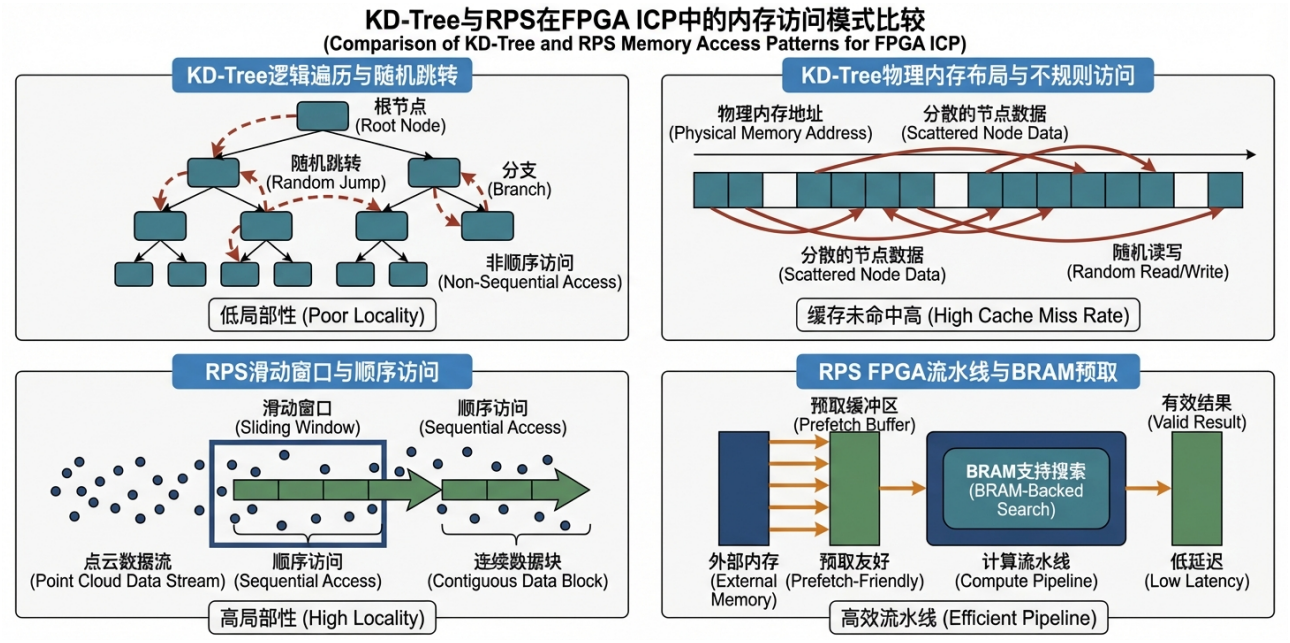


图 56: RPS 与 KD-Tree 访问模式的机制对比示意。该图用于说明树遍历的随机访存与扫描线窗口搜索的顺序访存差异，不对应单一实验中的精确缓存统计或时延数值。

[8] 没有继续沿用树索引，而是把点云匹配改写成硬件加速的暴力近邻搜索（HA-BFNN），再配合流式预处理和阈值筛选控制候选集规模。这样做的理由很直接：在 FPGA 上，规则扫描和片上缓存往往比复杂树遍历更容易维持稳定吞吐。该方法依赖定点预处理和流式数据调度，在基于 AMD Kintex-7 的自定义板卡上，对单帧

14400 点数据实现 5.76 ms 匹配时间，整套系统在 3.4 W 功耗下相对 CPU 达到 17.36 倍加速，同时保持与软件实现相当的精度 [8]。它的局限也很明确：一旦候选集无法通过阈值和数据分布压到可控范围，暴力搜索的比较开销会先升高。

[7] 提供了另一条桥接路径，面向 Amazon Picking Challenge 数据集中的机器人抓取位姿估计，把传统 K-D 树搜索替换为分层图结构，并用排序网络实现并行 k-NN 选择；同时利用部分动态重配置在" 图构建" 和" 最近邻搜索" 之间复用硬件资源。实验结果表明，该系统在 4.2 W 功耗下将单次物体位姿估计压到 0.72 s，相对基于四核 CPU 和 K-D 树的实现达到 11.7 倍加速 [7]。这类结果说明，FPGA 的价值不只在于把一段搜索代码搬进硬件，还在于允许设计者围绕资源复用重写整条搜索路径。

5.2.2 多模式对应搜索的可重配置设计

不同场景对对应搜索的需求不同：室内 SLAM 需要高精度 k-NN（$k = 1$）；多帧融合需要距离约束 NN；点到平面 ICP 需要法向量约束 NN。[91] 提出多模式可重配置对应搜索框架：

- **KNN 模式**（k 最近邻）：适合标准 P2P ICP，对应质量最好。
- **RNN 模式**（距离约束最近邻）：仅返回距离 $\leq r$ 的点，自动过滤远离点，适合含外点场景中的距离门控，对应第 3.2 节 中常见的阈值思想。
- **AKNN 模式**（近似 k 最近邻）：以近似搜索换取更高速度，其设计目标与第 4.4 节 的软件近似搜索思路一致，但这里把折中关系落在硬件可配置寄存器上。

三种模式通过寄存器在运行时切换，不必重新生成比特流。Deng 等将这种多模式搜索建立在扫描线辅助的 SA-RPS 结构上，并在 64 线 LiDAR 的 KITTI 场景中报告 21.5 FPS 的实时配准；其 SA-RPS-CS 搜索加速器相对既有 FPGA 设计达到 2.3–32.4 倍加速、1.8–26.2 倍能效提升，在保持 95% 以上召回率的同时精度损失很小 [91]。这种设计成立的前提，是输入仍保留扫描线拓扑；若点云来源不满足这一条件，SA-RPS 的构建收益会先下降。

5.2.3 全流程 FPGA 流水线与数据局部性优化

完整的 FPGA ICP 实现不能只盯住搜索模块，还要处理搜索结果怎样进入后续规约、求解和控制环。否则，即便对应搜索本身已经很快，数据搬运和模块衔接仍会吞掉端到端收益。

从公开设计看，较完整的 FPGA ICP 流水线至少包含四类模块：预处理模块（降采样与法向量估计）从 LiDAR 接口接收原始点云并输出精简点云及每点法向量；对应搜索模块（KD-Tree / RPS / HABFNN）并行处理所有源点的最近邻查询；协方差累积模块以流水线方式并行累积 H 矩阵的 9 个元素（树形加法器，深度 $\log_2 n$）；SVD 求解模块处理固定 $3times3$ SVD，展开为硬件友好的 Jacobi 迭代或直接 LUT 查表（4–6 Jacobi 扫描即可收敛到 FP32 精度）。

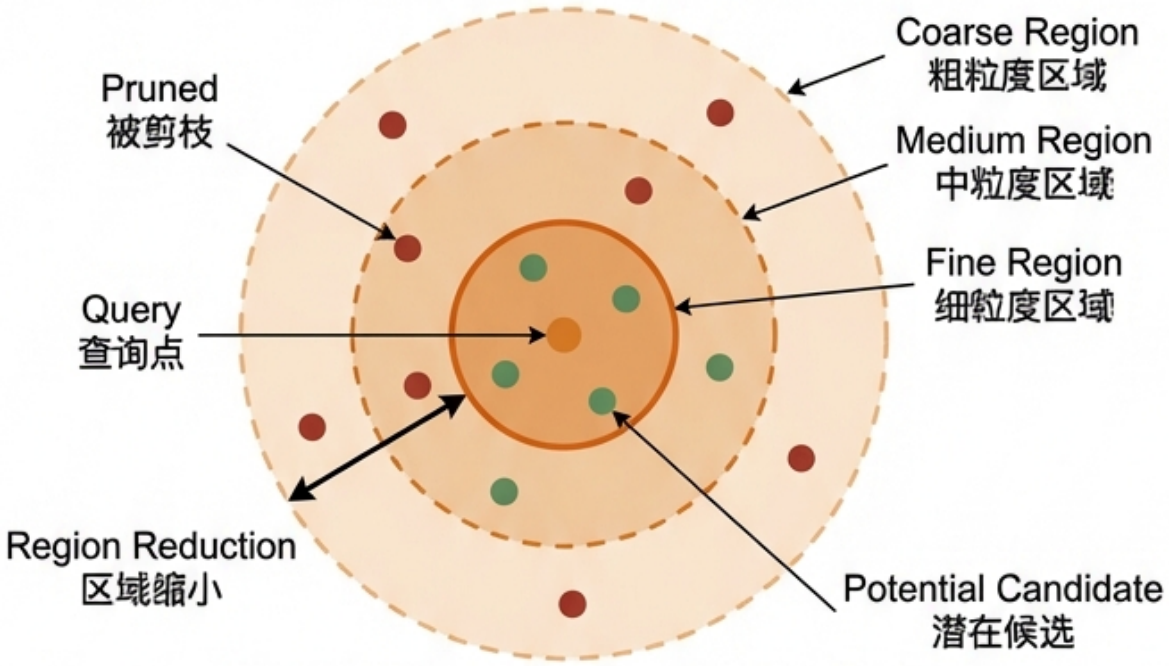
关键性能决策不在于" 所有数据都上片"，而在于哪些数据必须保持片上驻留、哪些数据可以流式经过。目标点云或索引结构如果能够稳定留在 BRAM 或片上缓存，对应搜索就能减少反复回访外部存储；源点云则更适合按帧流入，并在完成当前轮对应搜索后立即进入规约模块。RPS、HA-BFNN 和 SA-RPS 三类方案的共同点，都在于提高这种片上驻留比例。

定点化的作用不只是节省逻辑资源，还会决定一条流水线能否塞进目标芯片。[90] 把定点运算列为可重配置定位加速器的核心技术之一，并据此在性能、精度和资源之间做设计空间搜索；[8] 同样将定点预处理作为流式 HA-BFNN-ICP 的基础。它的前提是场景尺度和数值范围足够稳定；若量化范围设置过紧，误差会先在法向量估计、协方差累积或阈值筛选处放大。

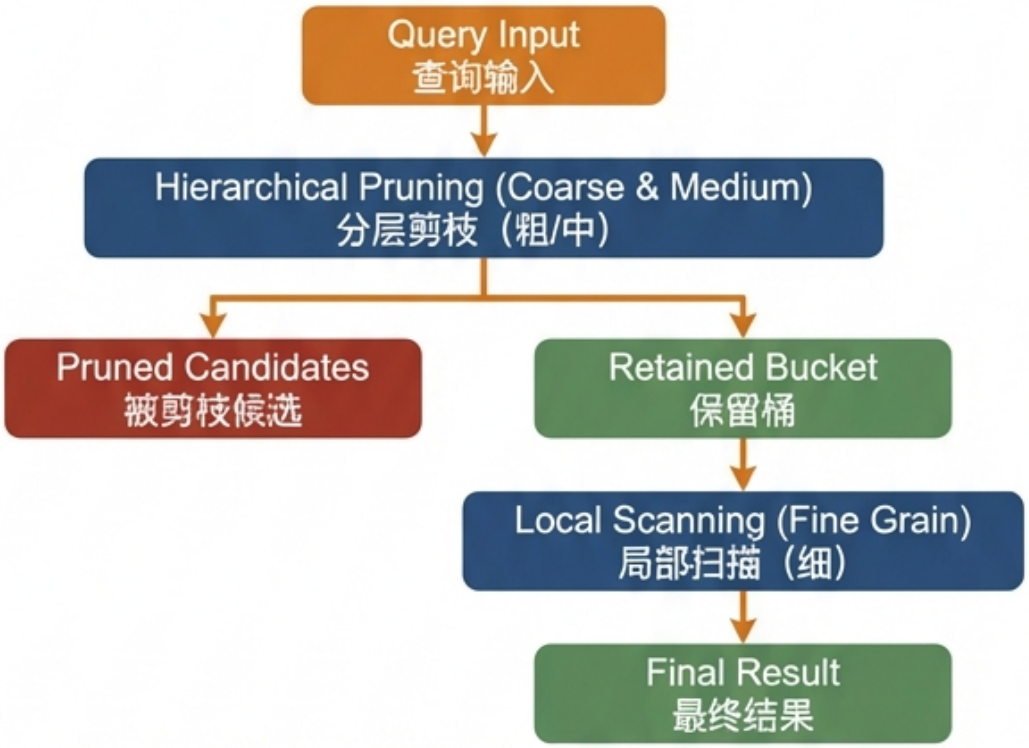
5.2.4 FPGA ICP 实现对比

综合来看，FPGA 路线的优势在于可以围绕具体场景快速重写搜索结构，并把这些改写直接映射到流水线和缓存层级上；其弱点则是片上资源有限，很多收益依赖点云组织形式、量化范围和候选集规模。一旦这些前提不成立，吞吐量会先在片外带宽或资源复用处下降。第 5.3 节 将继续讨论把这些路径进一步固化到 ASIC 后，会带来什么收益，以及又会失去哪些灵活性。

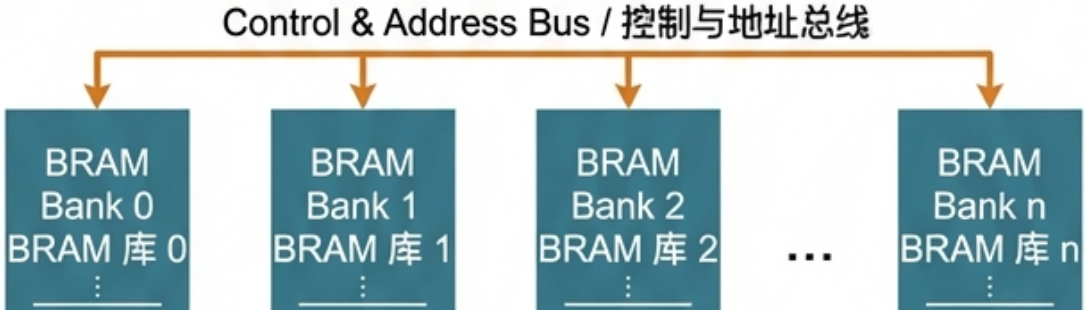
Hierarchy & Bucket-based FPGA Search Mechanism
基于层次和桶的 FPGA 搜索机制



(A): Coarse-to-Fine Candidate Regions



(B): Hierarchical Pruning & Scanning Flow



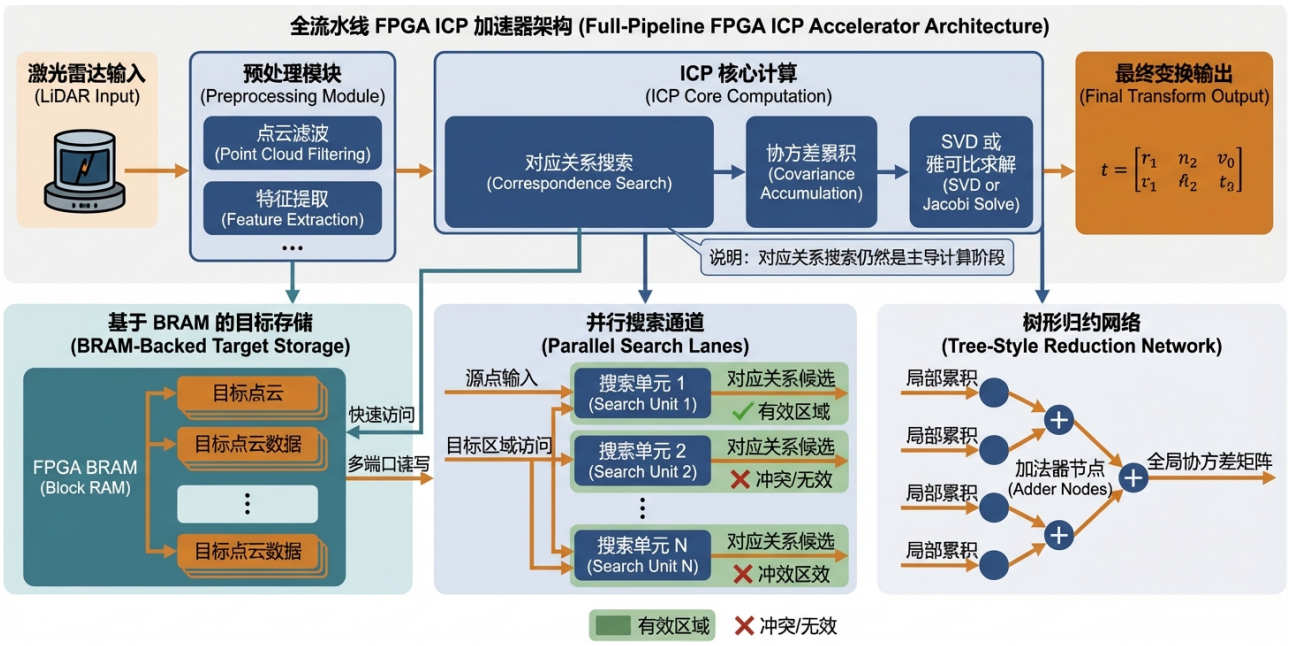


图 58: FPGA ICP 全流程流水线机制示意图。该图用于说明预处理、对应搜索、协方差规约和小矩阵求解的连接关系，以及 BRAM、DSP 和控制逻辑的资源分工，不对应统一实验条件下的精确资源占比或延迟比例。

表 26: 第 5.2 节 FPGA 代表实现汇总。表中仅保留论文明确报告的场景、设计点和结果，不把不同实验条件下的延迟、功耗和误差直接做横向换算。

实现	平台	场景	主要设计点	已报告结果
SoC-FPGA ICP [7]	SoC-FPGA	Amazon Picking Challenge 抓取位姿估计	分层图 k-NN + 排序网络 + 部分动态重配置	0.72 s/次位姿估计，4.2 W，11.7× 于四核 CPU + K-D 树
RPS-ICP [88]	FPGA	车载有组织 LiDAR 配准	RPS 结构 + RPS 构建器/搜索器协同	18.6 FPS；对应搜索 13.7× 于既有 FPGA；能效 50.7× 于 GPU
Runtime Reconfigurable Localization [90]	Xilinx Zynq-7000	KITTI、EuRoC 定位	定点计算 + 设计空间搜索 + 运行时重配置	59.1× 于 Intel CPU，9.2× 于 Arm CPU；平均能耗再降约 18%
HA-BFNN-ICP [8]	AMD Kintex-7 自定义板	3D LiDAR 建图	固定点预处理 + 流式 HA-BFNN + NNT 筛选	单帧匹配 5.76 ms；3.4 W；17.36× 于 CPU
Multi-mode SA-RPS-CS [91]	FPGA	KITTI 64 线 LiDAR 配准	SA-RPS + 滑动窗口缓存 + 多模式搜索	21.5 FPS；搜索 2.3–32.4× 于既有 FPGA；能效 1.8–26.2× 提升

5.3 机器人专用处理器（ASIC）(Robot-Specific Processors and ASICs)

当搜索结构和数据路径已经相对稳定时，ASIC 才开始显示出比 FPGA 更强的优势。原因不难理解：ASIC 不再为可重配置互连和通用 LUT 付出额外面积，可以把更多晶体管预算直接投到片上存储、专用比较单元和规约数据通路上。因此，本节不再讨论“如何快速试错”，而是讨论当设计者愿意接受更高前期成本和更低后续灵活性时，ICP 相关硬件还能向前推进到什么程度。

本节选择三条有代表性的路线。Tigris 代表“围绕点云配准内核做深度协同设计”的专用处理器；Tartan 代表“面向机器人工作负载抽象出通用微架构支持”的处理器路线；PointISA 代表“保留处理器生态、用 ISA 扩展承载点云原语”的折中方案。三者的共同问题都是如何减少点云应用中的数据搬运和控制开销，但它们接受的算法改写程度和软件兼容目标并不相同。

5.3.1 Tigris：点云感知的协同设计处理器

Tigris [78] 是较早明确把“点云配准中的 KD-Tree 搜索”当作专用架构对象来处理的代表性工作，在 TSMC 16nm FinFET 工艺节点实现。

Tigris 的关键判断是，KD-Tree 搜索并非完全不可并行，而是需要先通过算法改写暴露出可被硬件利用的并行性。算法层面，Tigris 通过两阶段 KD-Tree 和近似搜索，把原本逐点精确查询的流程改写成” 少量精确查询 + 邻域复用/插值” 的形式，减少真正需要进入树遍历的查询数量，并暴露查询级并行和节点级并行。架构层面，围绕这种并行性组织向量化搜索引擎，把树节点放入 bank-interleaved SRAM，并让搜索单元和递归单元协同执行，以减轻访问冲突和控制停顿。换句话说，Tigris 不是简单把软件里的 KD-Tree 放进硬件，而是先重排数据结构，再让硬件沿着新的访问顺序运行。

在 KITTI 相关点云配准任务上，Tigris 相对 RTX 2080 Ti 在 KD-Tree 搜索子过程上达到 77.2 倍加速和 7.4 倍功耗降低，并转化为端到端配准性能 41.7% 提升和功耗 3.0 倍下降 [78]。这一结果的意义在于，它证明了” 算法允许轻微近似 + 架构围绕搜索重写” 的组合确实能把随机树遍历转化成可加速的内核。它的前提也同样清楚：如果算法后来不再以 KD-Tree 搜索为主，这套专用数据通路的复用价值会迅速下降。

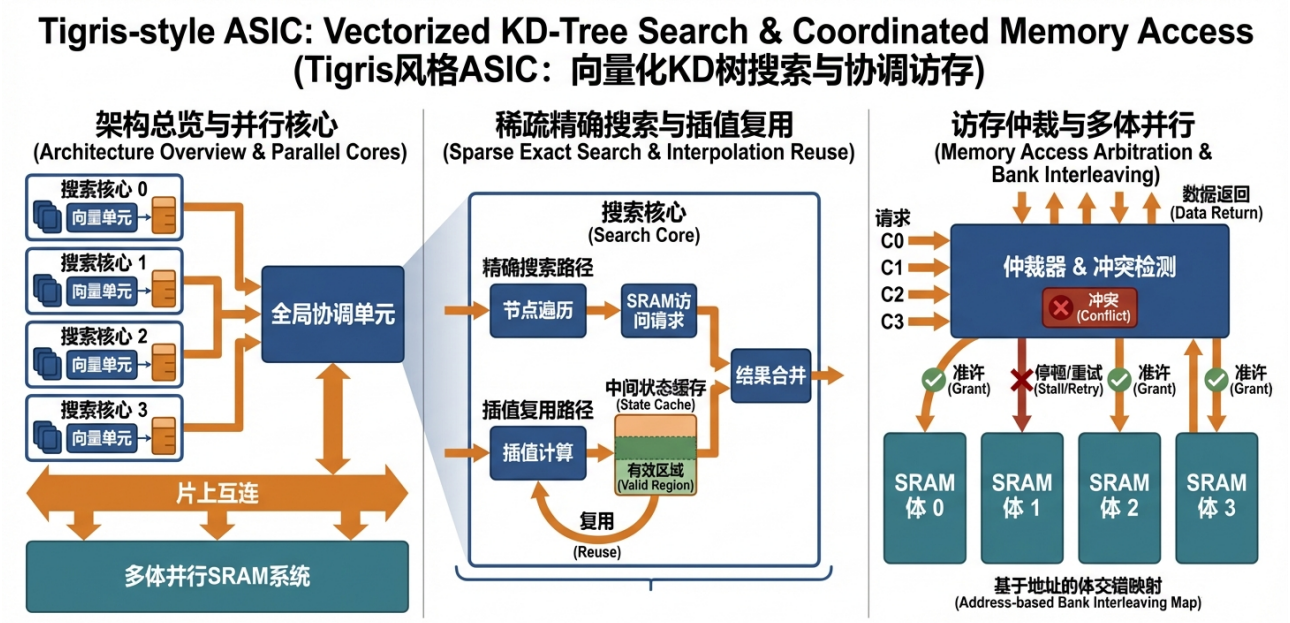


图 59: Tigris 向量化 KD-Tree 搜索引擎示意图。该图用于说明两阶段搜索、bank 交织存储和多核查询调度的关系，不对应统一实验条件下的精确冲突率、插值误差或每帧延迟。

5.3.2 Tartan：机器人应用的微架构设计

Tartan [81] 走的是另一条路线。它并不为 ICP 单独做一颗芯片，而是试图抽象出机器人应用在感知、规划和控制中的共性瓶颈，再把这些瓶颈变成处理器级支持。

Tartan 把问题表述为” 现有 CPU 既不够像专用加速器，又不够懂机器人工作负载”。论文指出三类共性特征：很多机器人内核的内存访问缺乏常规 CPU 假设下的规则局部性；一部分计算虽然规整，但规模很小，难以充分利用通用向量单元；控制与数据访问经常紧耦合，单纯增加算术吞吐并不能直接转化为系统收益。针对这些特征，Tartan 提出机器人语义预取和应用内缓存分区，用来缓和非规则访问对缓存层级的冲击；同时通过定向向量化和面向近似计算的硬件路径，提高小规模机器人内核的执行效率。与 Tigris 的单任务专用路线不同，Tartan 更强调端到端应用而非单一 ICP 内核，关注的不是某一步搜索的绝对峰值加速，而是整套机器人软件在较小面积开销下能否持续受益。

在 RoWild Suite 的六个端到端机器人应用上，Tartan 对遗留软件平均提升 1.2 倍；对经过针对性优化但不可近似的工作负载平均提升 1.61 倍；对允许近似的工作负载平均提升 2.11 倍，峰值可达 3.87 倍 [81]。这一结果没有 Tigris 那样激进，但它说明另一个事实：如果目标是支撑更广的机器人软件栈，那么“适度专用 + 保留处理器通用性”比只追求单一内核极限更现实。

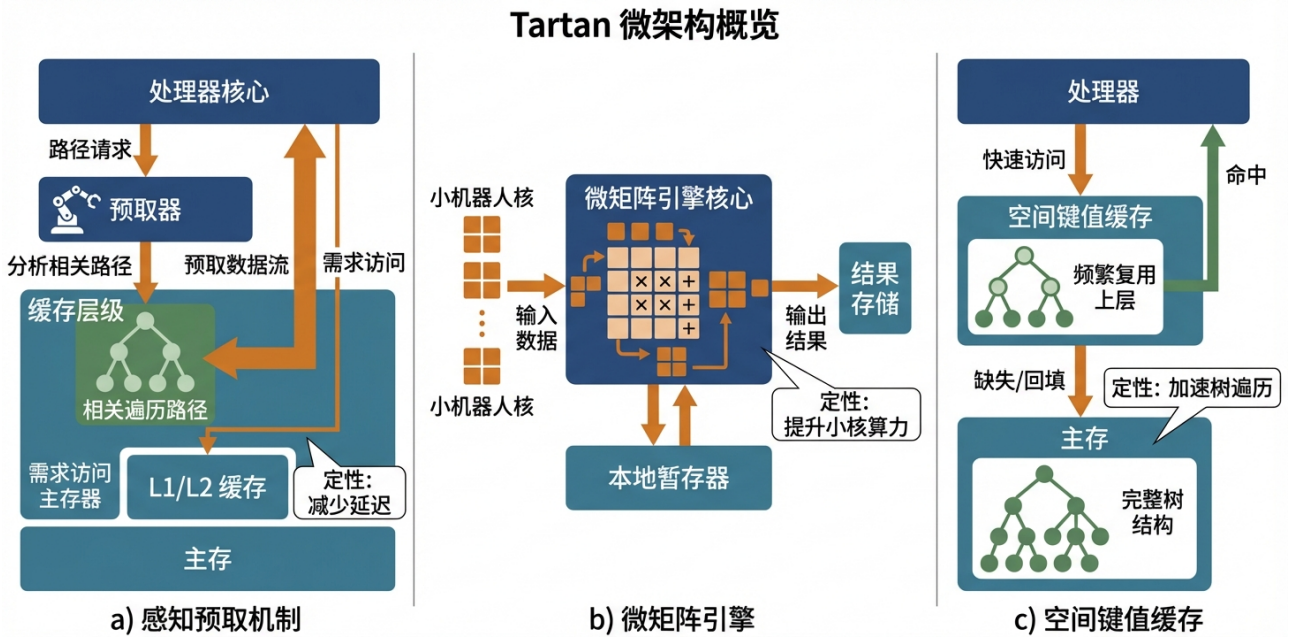


图 60: Tartan 处理器微架构支持示意图。该图用于说明机器人语义预取、面向小规模内核的计算支持和缓存层级协同的关系，不对应单一实验中的精确预取准确率、命中率或周期数。

5.3.3 PointISA：ISA 扩展的协同设计

PointISA [92] 继续向“保留软件生态”的方向推进。它不直接做整颗点云专用处理器，而是在现有 ISA 上增加点云相关原语，并配套统一硬件执行结构。

PointISA 不把每个点云算法都做成独立加速器，而是把欧氏距离、多维排序等高频操作抽成 ISA 扩展，再把 FPS、kNN 等算法重写成能利用这些扩展的并行模式：ISA 层增加点云加载、距离计算和排序相关指令；硬件层使用统一执行结构，同时支持这些扩展指令和常规矩阵乘法；算法层把 FPS、kNN 等流程改写成多点对多点（MP2MP）模式，以提高并行利用率。

在 gem5 和 AArch64 基线上的评测中，PointISA 在多种点云工作负载上实现平均 5.4 倍加速和 4.9 倍能效提升，面积开销约 0.9% [92]。这条路线的价值不在于单点极限性能，而在于它把“点云原语”纳入了处理器指令语义，因而更容易进入既有编译器和软件栈。

5.3.4 ASIC 设计的关键权衡

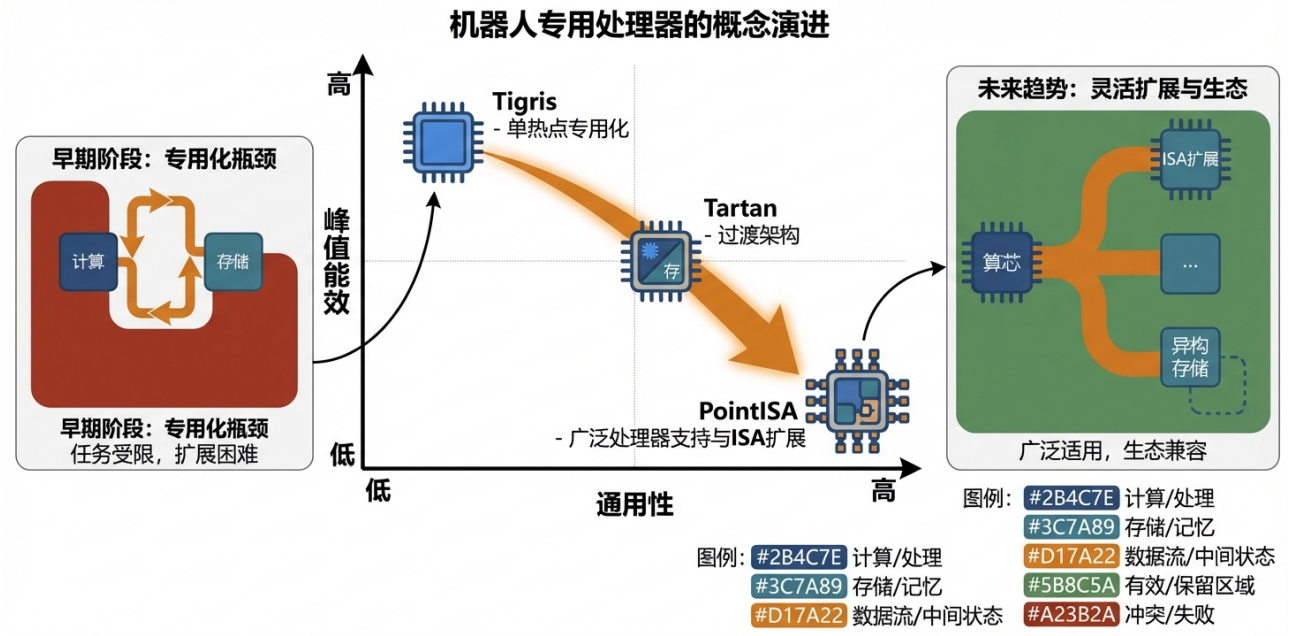


图 61: 机器人专用处理器架构三代演进对比示意。第一代（Tigris, MICRO'19）：算法-架构协同优化 KD-Tree 搜索，向量化搜索引擎 + bank-interleaved SRAM，聚焦点云配准单任务。第二代（Tartan, ISCA'24）：系统分析机器人 workload 特征，设计专用预取器 + 微型矩阵引擎 + KV Cache，覆盖感知-规划-控制全栈。第三代（PointISA, MICRO'25）：ISA 扩展路线，在 RISC-V 基础上增加点云专用指令，兼顾软件生态与专用加速，pdist/pknn/pcov 三类核心指令。横轴：泛化性（专用 → 通用），纵轴：峰值能效（高 → 低）。

ASIC 设计在通用性与效率、算法固定化风险以及开发成本三个维度上面临根本性权衡。在通用性与效率维度上，Tigris 更接近" 围绕单一热点深挖" 的专用内核，PointISA 更接近" 把热点下沉到 ISA 语义"，Tartan 则处在两者之间，试图为更广的机器人软件栈提供有限但持续的收益；实际选择取决于产品究竟追求单任务极限，还是追求较长的软件生命周期。在算法固定化风险维度上，只要硬件收益依赖特定搜索结构、数值表示或近似策略，芯片流片后就很难像 FPGA 那样继续调整；如果未来系统转向第 3.6 节 的全局初始化或第 3.7 节 的深度学习配准，专用搜索数据通路的价值会迅速下降。在开发成本维度上，ASIC 不只贵在流片，还贵在前期验证、软件适配和量产前风险控制；对研究原型而言，这意味着 ASIC 更适合用来验证" 已经足够稳定" 的热点，而不是承载仍在快速变化的算法探索。

表 27: 第 5.3 节专用处理器路线与可重配置对照路线汇总。表中只保留论文明确报告的定位、机制和结果，不对不同论文的延迟、工艺或功耗做未经统一条件校准的硬性排序。

方案	路线定位	主要对象	代表机制	已报告结果
Tigris [78]	点云配准专用 ASIC	KD-Tree 搜索主导的点云配准	两阶段 KD-Tree + 近似搜索 + 向量化搜索引擎	KD-Tree 搜索 77.2× 于 RTX 2080 Ti；端到端性能提升 41.7%；功耗降 3.0×
Tartan [81]	机器人处理器微架构	感知、规划、控制混合工作负载	定向向量化、近似执行、机器人语义预取、缓存分区	遗留软件 1.2×；不可近似工作负载 1.61×；可近似工作负载 2.11×，峰值 3.87×
PointISA [92]	ISA 扩展 + 统一执行结构	多类点云分析工作负载	点云原语 ISA 扩展 + MP2MP 算法改写	平均 5.4× 加速、4.9× 能效提升；面积开销约 0.9%
Multi-mode SA-RPS-CS FPGA [91]	可重配置对照路线	64 线 LiDAR 配准	SA-RPS + 多模式搜索	21.5 FPS；搜索 2.3–32.4× 加速；能效 1.8–26.2× 提升

综合来看，ASIC 路线的真正价值不在于“任何指标都比 FPGA 高”，而在于它可以围绕已经稳定的热点建

立更紧的存储和执行耦合。代价是算法一旦切换，硬件复用空间会迅速收缩。第 5.4 节 将继续讨论另一条不同的路线：不再优先重写执行核心，而是把距离计算和候选维护尽量推入存储阵列，以直接处理带宽墙问题。

5.4 近存储计算（PIM）(Processing-in-Memory)

第 5.2 节 和第 5.3 节 展示了两种思路：要么重写搜索结构并围绕其建立流水线，要么把已经稳定的搜索热点固化进专用处理器。PIM 采取的切入点不同。它默认“数据搬运本身就是瓶颈”，因此不优先继续强化执行核心，而是把距离计算、候选筛选和 Top-k 维护尽量推向存储阵列附近。

5.4.1 内存带宽墙与 PIM 的动机

PIM 研究的出发点是，点云 kNN 和对应搜索包含大量“取数据、做少量比较、更新候选”的短链路操作。对这类负载而言，外部存储带宽和访问时延经常先于算术吞吐量成为限制因素。[20] 直接把点云 kNN 的主要挑战概括为“计算强度高且内存需求大”；[89] 也把设计重点放在减少存储占用和提升能效，而不是继续堆叠通用计算核心。换句话说，PIM 关心的不是“每次距离计算有多复杂”，而是“为了完成这些距离计算，系统需要搬多少数据”。

因此，PIM 的价值不应被理解为某个统一倍数的加速，而应理解为一种新的瓶颈转移方式：只要距离比较和候选维护能够在阵列内部完成，外部总线就不再承担全部数据往返，系统瓶颈会从“搬数据”转向“阵列内部如何组织并行计算”。

5.4.2 PICK：SRAM-PIM 加速 KNN 搜索

PICK [20] 是面向点云 kNN 搜索的 SRAM-PIM 代表工作之一，重点展示了如何把距离计算和 Top-k 维护压进 SRAM 阵列。

PICK 基于 BS-PIM（Bit-Serial Processing-in-Memory）原理：在 6T SRAM 阵列的每列末端添加 1 位加法器（adder cell），使 SRAM 可以在不读出数据的情况下，在片内逐位执行加法运算。KNN 搜索中的距离计算被分解为逐位的”bit-serial 乘加” 操作：

$$d^2 = \sum_{k=1}^{3} (p_k - q_k)^2 = \sum_{b=0}^{B-1} 2^b \cdot extPopCount(P_{k,b} \oplus Q_{k,b}) \tag{80}$$

其中 B 为量化位宽，$P_{k,b}$ 和 $Q_{k,b}$ 为坐标第 b 位的二进制值，PopCount 由片内硬件完成。通过这种方式，候选点的距离计算不再要求把全部数据先搬到外部计算单元再回写结果。为保证实用性，PICK 还配套三类电路机制：位宽裁剪在保证精度影响可控的前提下减少 bit-serial 计算长度；筛选与选择策略使任意 k 值下的 Top-k 搜索保持接近常数的时间复杂度；两级流水线将距离计算与 Top-k 搜索并行化，以隐藏一部分阵列内部时延。在 KITTI、SONN、S3DIS 和 DALES 等真实点云数据上，PICK 相对前一代代表设计达到 4.17 倍加速和 4.42 倍节能 [20]，说明若 kNN 可以被表述成适合 bit-serial 阵列执行的形式，PIM 的收益会直接体现在能效和数据搬运缩减上。

5.4.3 C^2IM-NN：CAM-PIM 与 CNN 预测协同

C^2IM-NN [89] 采取了与 PICK 不同的路线：不是在 SRAM 阵列里做 bit-serial 计算，而是利用 CAM 的相似匹配特性，把查询过程表达成“在存储阵列中直接找最相近的内容”。

C^2IM-NN 利用模拟 CAM 实现近似相似度搜索：让每个存储单元直接参与” 查询点与存储点有多接近” 的比较, 由片内电路输出候选结果, 从而持续减少数字域中的显式数据搬运和逐点比较。在此基础上, C^2IM-NN 还引入轻量级 1D-CNN 在搜索前预测查询更可能落在哪个区域，再让 CAM 只在该区域内执行相似匹配——这一剪枝的前提是区域预测本身足够轻量，且误判率不能把后续搜索精度拖垮。在 28 nm CMOS 验证下，C^2IM-NN 相对之前的 ASIC 加速器达到 23.08 倍能效提升，并减少 48.4% 存储占用 [89]。这一结果说明，若允许引入模拟匹配和预测剪枝，PIM 的收益可以继续扩大；但代价是设计会明显更依赖工艺、校准和可靠性控制。

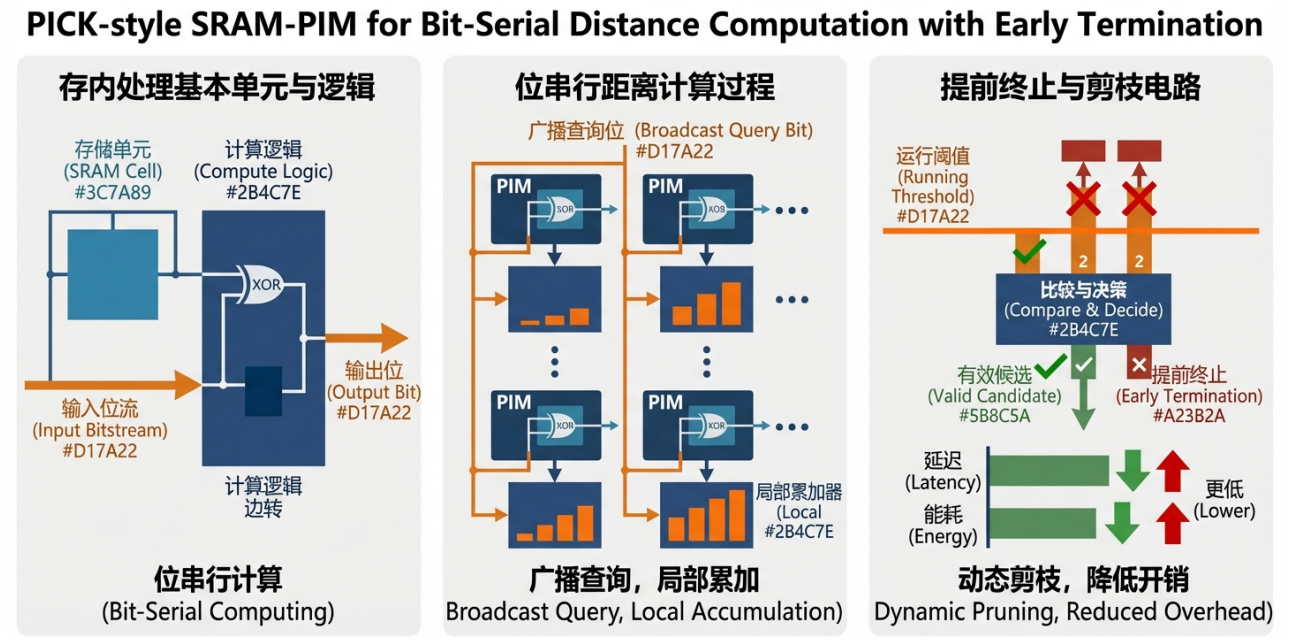


图 62: PICK 的 bit-serial 阵列执行与候选筛选机制示意图。该图用于说明阵列内距离计算、位宽裁剪和候选维护之间的关系，不对应单一实验中的精确时延分解或每列候选规模。

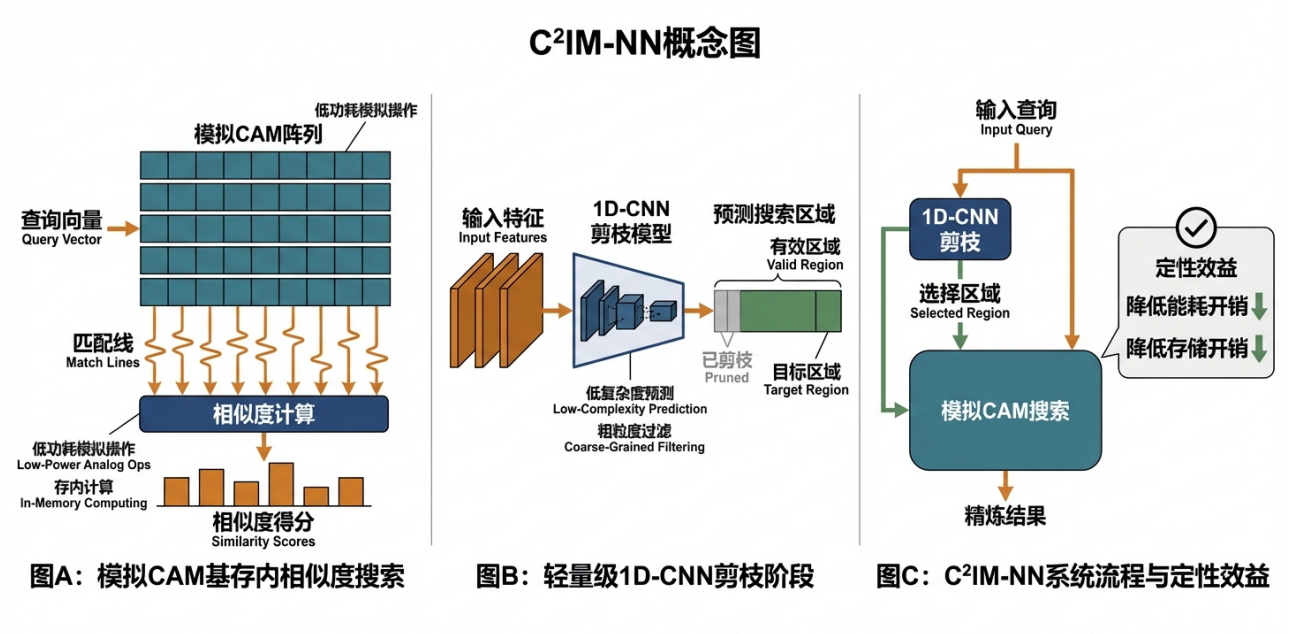


图 63: C^2IM-NN 的 CAM 搜索与预测剪枝协同示意图。该图用于说明模拟匹配、区域预测和系统流水线之间的关系，不对应单一实验中的精确预测准确率、模拟延迟或总帧时延。

5.4.4 PIM 在 ICP 系统中的集成挑战

尽管 PIM 在 KNN 搜索上性能出色，将其集成到完整 ICP 系统仍面临工程挑战：

尽管 PIM 在 KNN 搜索上性能出色，将其集成到完整 ICP 系统仍面临三类工程挑战。首先是点云更新问题：ICP 每帧都会引入新的查询点云，而建图系统中的目标点云也可能增量更新；PIM 若要保持高收益，就需要控制这些写入和重组织成本，否则阵列内部节省的数据搬运会被更新代价抵消，对增量维护问题可参考第 4.1 节 的动态数据结构思路，但如何把它们映射到 PIM 友好的更新路径仍未定型。其次是稀疏访问模式问题：PIM 阵列更喜欢规则、成批的并行访问，而标准树遍历带有大量随机跳转，因此算法经常需要先把搜索流程改写成阵列友好的形式才能真正发挥 PIM 的优势。第三是与主控处理器的接口问题：即便核心搜索已经下沉到阵列内部，系统仍要处理查询下发、结果回读和控制同步，若接口协议和 DMA 调度没有处理好，PIM 的阵列收益仍可能在系统边界被吞掉。

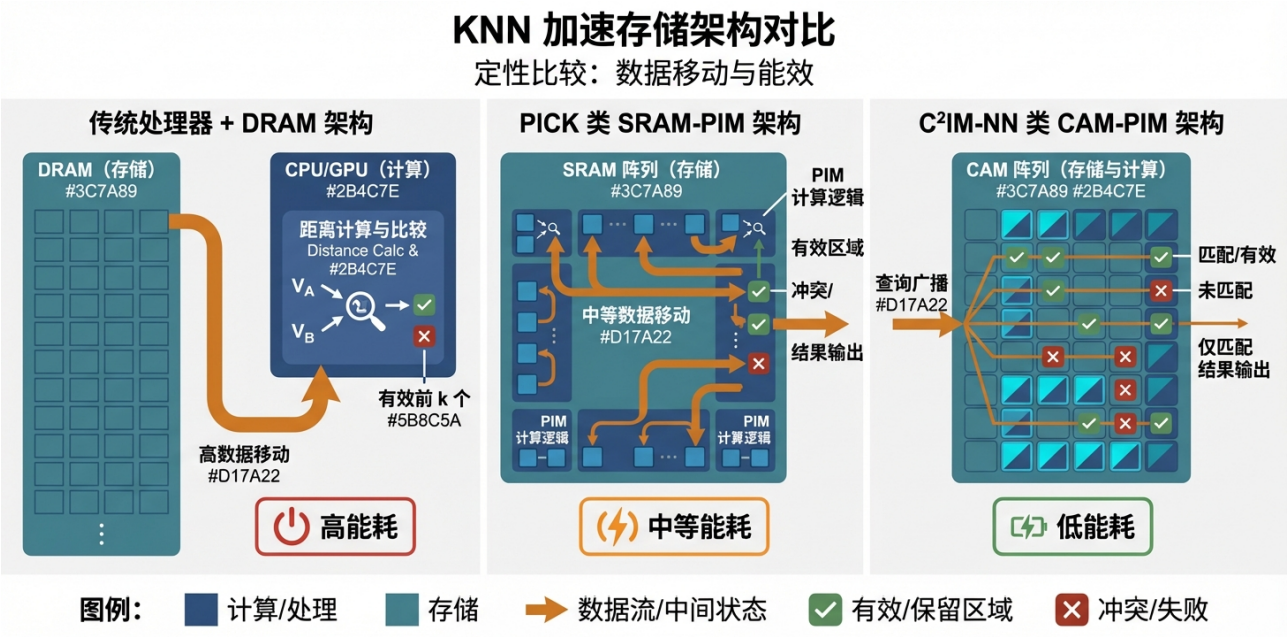


图 64: 传统存储架构、SRAM-PIM 与 CAM-PIM 的机制对比示意。该图用于说明“外部搬运主导”与“阵列内部执行主导”两类路线的差异，不对应统一实验条件下的精确带宽或能效坐标。

5.4.5 PIM 方案综合对比

表 28: 第 5.4 节 PIM 代表方案汇总。表中仅保留论文明确报告的机制与结果，不对不同实现的总时延和工艺可迁移性做未经验证的统一排序。

方案	PIM 类型	主要机制	已报告结果	主要代价
PICK [20]	SRAM bit-serial PIM	阵列内距离计算 + 位宽裁剪 + 两级流水线	4.17× 加速，4.42× 节能	需要让 kNN 流程适配 bit-serial 执行与片上存储组织
C^2IM-NN [89]	CAM-based PIM	模拟相似匹配 + 1D-CNN 预测剪枝	23.08× 能效提升，48.4% 存储占用下降	更依赖模拟电路设计、工艺校准和系统可靠性控制

PIM 的意义在于提供了一种不同于 FPGA 和 ASIC 的问题重述方式：如果带宽墙才是主瓶颈，那么就不必先从执行核心下手，而可以先改变数据与计算的相对位置。它的工程门槛同样明显，包括更新路径、接口协议和阵列友好型算法表达。第 5.5 节 将把第 5.2 节 到本节的几条路线放到同一共设计框架下，讨论哪些算法改写值得换取硬件收益。

5.5 算法-硬件协同设计方法论 (Algorithm-Hardware Co-design Methodology)

第 5.2 节 到第 5.4 节 展示的几条路线虽然器件形态不同，但真正决定收益的并不只是硬件本身，而是算法是否愿意为硬件让渡一部分自由度。本节不再追求给出统一分数或统一坐标，而是把前文已经核实过的工作归纳为几类常见的协同设计动作，说明这些动作在什么条件下成立，又会把风险转移到哪里。

5.5.1 协同设计的三层框架

从前文案例出发，协同设计大致可以分成三个层次：

层次一：数值与数据组织适配在这一层，算法目标函数并不改变，修改集中在表示形式和存储布局上。例如，固定点量化让 [90] 和 [8] 能把更多算子塞进目标芯片；而将目标点云或索引结构长期保留在片上缓存，则使第 5.2 节 中的多条 FPGA 流水线可以减少反复的片外访问。这一层的优点是改动相对保守，风险集中在量化误差和边界数据范围是否可控。

层次二：搜索流程改写当仅靠数值和布局仍不足以支撑硬件吞吐时，研究者会开始改写最近邻搜索本身的执行流程。Tigris 用两阶段 KD-Tree 与近似搜索挖掘查询级和节点级并行 [78]；PointISA 把点云原语下沉到 ISA，并将 FPS、kNN 改写成 MP2MP 模式 [92]；PICK 则把距离计算和 Top-k 搜索表达成阵列友好的 bit-serial 流程 [20]。这一层的核心收益来自“减少真正昂贵的查询和搬运”，代价是软件实现与经典算法流程开始出现可见偏离。

层次三：搜索结构替换收益最大的一类协同设计，经常来自直接更换搜索结构或候选生成方式。RPS 和 SA-RPS 利用有组织 LiDAR 的扫描线拓扑，把随机树遍历替换成局部窗口搜索 [88][91]；HA-BFNN-ICP 进一步接受规则扫描和阈值筛选，以换取稳定的流式吞吐 [8]；C^2IM-NN 则把搜索改写成 CAM 相似匹配与区域预测的组合 [89]。这一层的风险也最直接：一旦输入点云不再满足结构化拓扑、候选分布不再受控，或区域预测失准，收益会首先回落。

5.5.2 协同设计的评估原则

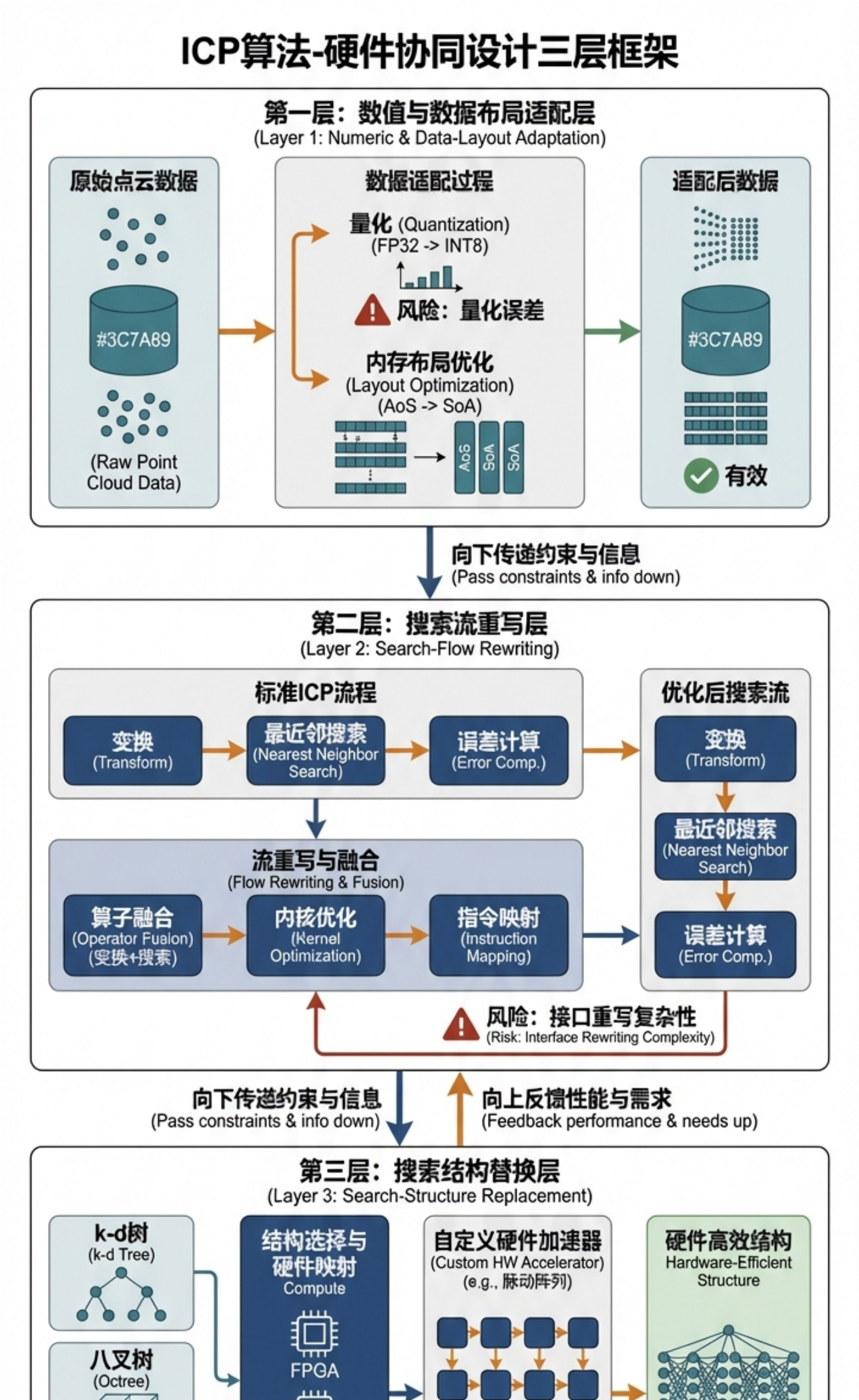
协同设计不适合用一个统一公式打分，因为不同论文的场景、数据规模、基线和功耗测量方式都不同。更稳妥的做法，是沿着四个问题逐项核验：

1. **算法是否真的改写了瓶颈**：如果改动没有触及对应搜索、候选筛选或数据搬运，硬件收益通常只会停留在局部算子。
2. **精度损失是否在原始论文中被显式报告**：例如 Tigris、RPS、HA-BFNN-ICP、C^2IM-NN 都给出了各自场景下的精度或召回边界；没有核实的数据不应被横向拼接。
3. **硬件收益是否来自统一口径的指标**：有的论文报告子过程加速，有的报告端到端帧率，有的报告能效或面积开销。只有指标口径一致时，比较才有意义。
4. **系统代价是否被单独说明**：包括重配置开销、更新路径、接口同步、量化误差和校准复杂度。若这些代价被省略，局部加速往往无法转化为系统收益。

5.5.3 典型收益与典型代价

把前文工作放在一起看，可以得到几条较稳定的结论：

- 若输入点云存在扫描线或结构化拓扑，直接利用这种结构比继续优化通用树遍历更划算，[88] 和 [91] 都属于这一类。
- 若目标平台更在意稳定吞吐而不是算法形式保真，规则扫描和阈值筛选会比复杂索引更容易映射到硬件，[8] 和 [20] 体现了这一点。
- 若系统仍需要较强的软件通用性，就需要把协同设计限制在处理器接口层或 ISA 层，[81] 与 [92] 都更接近这种路线。


ICP算法-硬件协同设计三层框架
第一层：数值与数据布局适配层
(Layer 1: Numeric & Data-Layout Adaptation)
原始点云数据
#3C7A89
(Raw Point Cloud Data)
数据适配过程
量化 (Quantization)
(FP32 -> INT8)
风险：量化误差
内存布局优化
(Layout Optimization)
(AoS -> SoA)
AoS
SoA
SoA
适配后数据
#3C7A89
有效
向下传递约束与信息
(Pass constraints & info down)
第二层：搜索流重写层
(Layer 2: Search-Flow Rewriting)
标准ICP流程
变换
(Transform)
最近邻搜索
(Nearest Neighbor Search)
误差计算
(Error Comp.)
优化后搜索流
变换
(Transform)
最近邻搜索
(Nearest Neighbor Search)
误差计算
(Error Comp.)
流重写与融合
(Flow Rewriting & Fusion)
算子融合
(Operator Fusion)
(变换+搜索)
内核优化
(Kernel Optimization)
指令映射
(Instruction Mapping)
风险：接口重写复杂性
(Risk: Interface Rewriting Complexity)
向下传递约束与信息
(Pass constraints & info down)
向上反馈性能与需求
(Feedback performance & needs up)
第三层：搜索结构替换层
(Layer 3: Search-Structure Replacement)
k-d树
(k-d Tree)
八叉树
(Octree)
结构选择与硬件映射
Compute
FPGA
自定义硬件加速器
(Custom HW Accelerator)
(e.g., 脉动阵列)
硬件高效结构
Hardware-Efficient Structure

对应的代价同样明确：

- 结构化搜索依赖输入点云的组织形式。
- 定点化和近似搜索依赖误差边界可控。
- PIM 和模拟匹配依赖阵列友好的数据表达以及更复杂的更新、校准与接口设计。

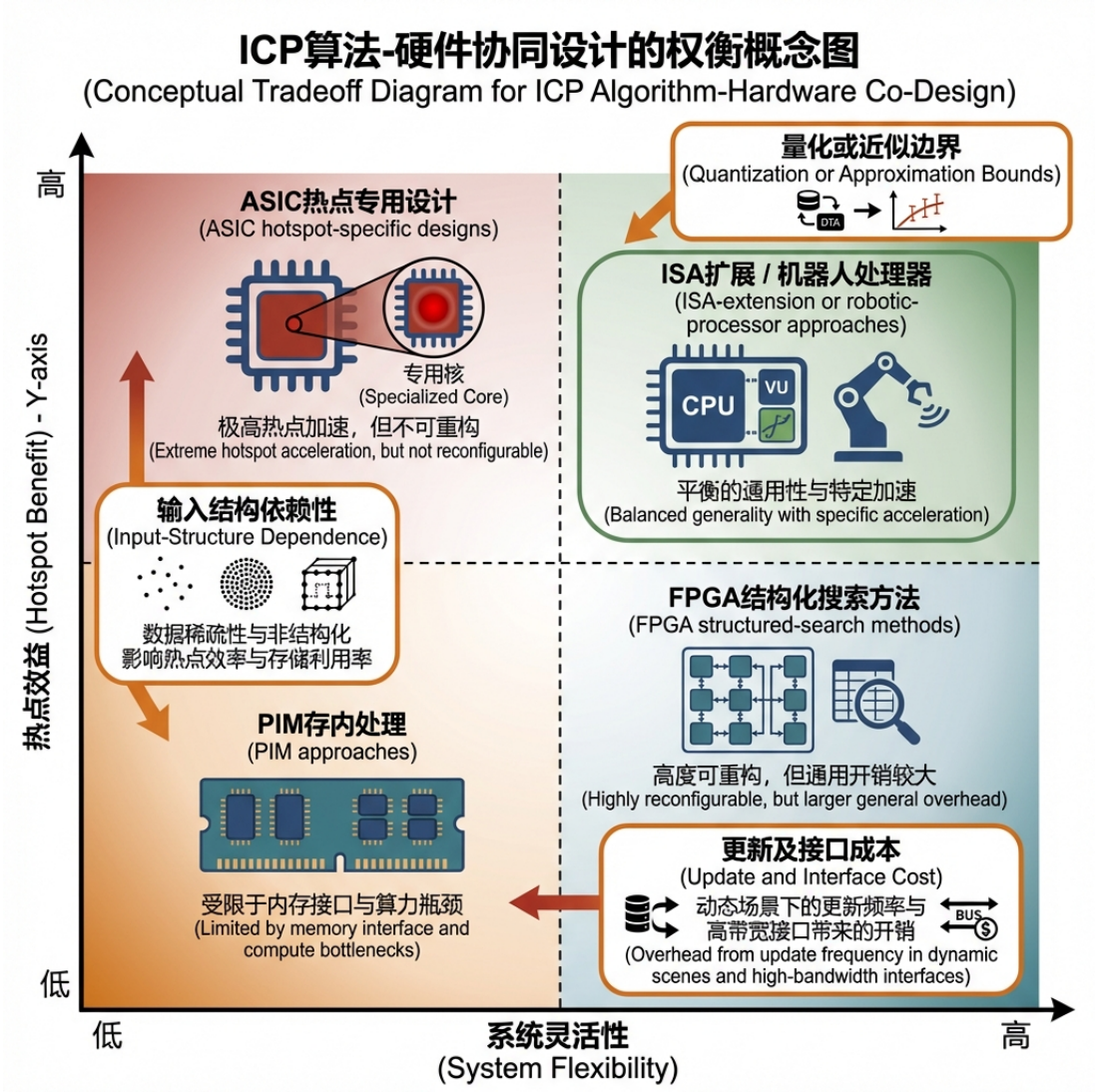


图 66: 协同设计中“收益与代价并存”的概念示意图。该图用于说明不同路线在延迟、误差、灵活性之间的相对关系，不对应统一实验条件下的精确 Pareto 前沿。

5.5.4 仍待解决的问题

第 3.7 节 中的深度学习配准引入了与经典 ICP 不同的计算图，如何在不牺牲经典 ICP 时延优势的前提下同时支持这些新算子，仍缺少成熟答案。动态数据结构和增量地图维护对 FPGA、ASIC 和 PIM 同样是难点：当前多数工作更擅长处理静态或半静态搜索结构，而不擅长高频更新。即使未来平台能把 CPU、可重配置逻辑和近存储阵列放进更紧的封装，更新路径、接口协议、热设计和软件调度仍然需要单独解决——器件更近，并不自动等于系统更简单。

综上，算法-硬件协同设计并不是一句口号，而是一组必须被明确写出来的交换条件：为了换取更低延迟或更高能效，算法究竟让出了哪些自由度，硬件又因此获得了哪些可利用的规则性。只要这些交换条件没有被讲清楚，所谓“硬件加速”就很容易退化成不可复现的局部结果。

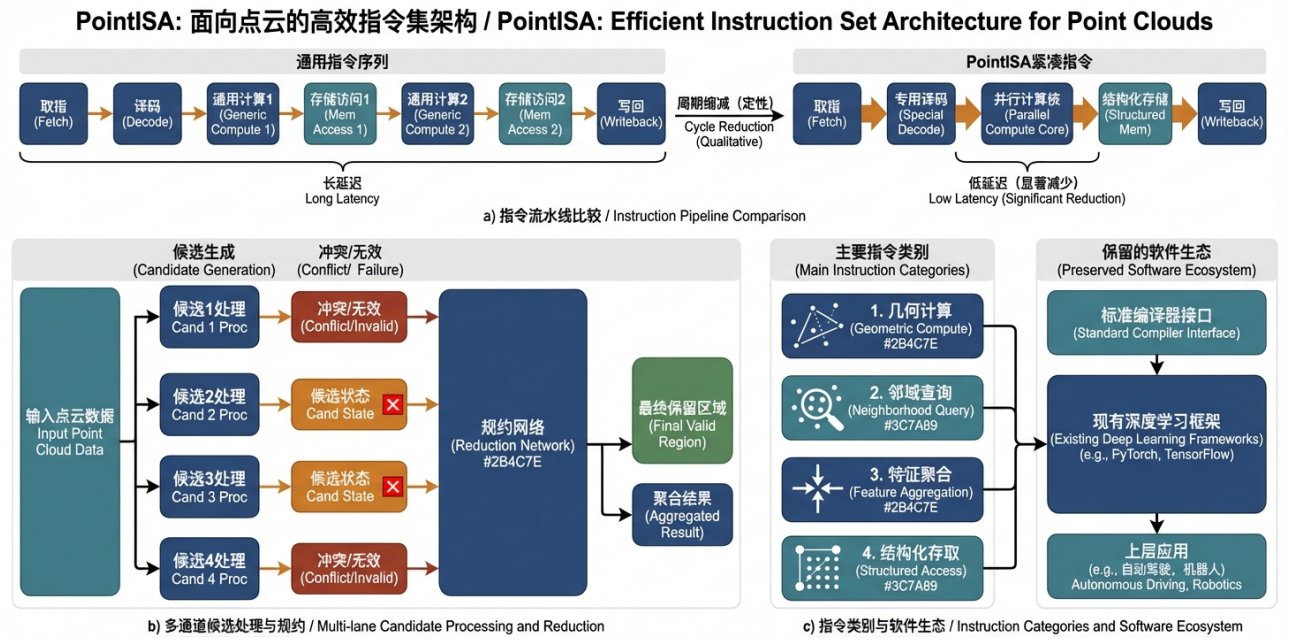


图 67: PointISA 点云专用指令与通用指令序列之间关系的机制示意图。该图用于说明“把点云原语提升到 ISA 层”这一思路，不对应统一实验条件下的精确周期数或整体 ICP 加速比。

表 29: 第 5.5 节协同设计动作与代表工作汇总。表中强调“改了什么”和“换来了什么”，不把不同论文的异构指标强行压缩成单一分数。

协同动作	代表工作	直接收益	主要约束
数值与布局适配	Runtime Reconfigurable Localization、HA-BFNN-ICP	降低资源占用，提升片上驻留比例	量化范围、误差边界、资源预算
搜索流程改写	Tigris、PointISA、PICK	减少昂贵查询与数据搬运	需要重写算法执行流程和软件接口
搜索结构替换	RPS、SA-RPS、HA-BFNN、C^2IM-NN	最大化规则访问和阵列友好执行	强依赖输入结构、候选分布或预测准确性

5.6 本章小结

硬件加速的根本动机，在于点云对应搜索和 kNN 查询会把系统瓶颈推向数据访问侧，而不是继续停留在通用算术吞吐上。本章沿 FPGA、ASIC 与 PIM 三条路线展开，实际讨论的都是同一个问题：为了减少随机访存和数据搬运，算法愿意让出哪些自由度，硬件又因此获得了哪些可利用的规则性。

第 5.2 节 表明，可重配置逻辑的优势不只是“把搜索做快”，而是允许研究者围绕具体输入结构重写搜索流程：RPS 和 SA-RPS 利用扫描线拓扑把随机搜索改成局部窗口搜索 [88][91]，HA-BFNN-ICP 则接受规则扫描与阈值筛选以换取稳定流式吞吐 [8]。第 5.3 节 说明，若热点已经足够稳定，专用处理器可以把这种协同进一步固化：Tigris 围绕 KD-Tree 搜索建立两阶段搜索和向量化数据通路 [78]，Tartan 与 PointISA 则分别在处理器微架构层和 ISA 层保留了更强的软件通用性 [81][92]。第 5.4 节 进一步把问题改写为“怎样减少搬运本身”：PICK 用 SRAM bit-serial 阵列压低 kNN 的搬运成本 [20]，C^2IM-NN 则把 CAM 相似匹配和区域预测结合起来继续扩大量效收益 [89]。第 5.5 节 的总结因此可以归结为一句话：收益最高的硬件加速，几乎都伴随着搜索结构、数值表示或数据组织的同步改写。

从第 4 章 的软件优化到本章的专用硬件设计，ICP 的工程问题已经不再只是“能不能算完”，而是“为了在目标平台上算完，需要先改哪些算法前提”。这一点也决定了后续评测不能只看局部算子。第 6 章 将转向应用与基准，讨论这些硬件和算法选择在具体场景里该如何比较、如何取舍。

6. 应用、基准与横向比较

第 3 章 到第 5 章 已经把 ICP 的目标函数、鲁棒机制、软件优化和硬件实现拆开讨论；但工程部署并不按“章节”发生，而是按场景约束发生。自动驾驶前端首先受制于扫描频率和里程计漂移，工业检测首先受制于重复精度和夹具先验，医疗配准首先受制于失败告警与不确定性表达。只有把这些约束与公开基准、代表性数值和失败模式放在同一语境下，前面几章的方法比较才有可执行的选型意义。

本章沿三条主线展开。首先，第 6.1 节 讨论自动驾驶、工业检测、机器人抓取和医疗配准四类场景，逐项说明“为什么选这种 ICP 变体，而不是前一类方法”，并补入代表论文在原文里给出的数据集、指标和数值。随后，第 6.2 节 把 TUM RGB-D、ScanNet、3DMatch/3DLoMatch、KITTI、ETH、nuScenes 等基准的规模、阈值和适用边界放到同一张表里，避免把来自不同协议的结果直接并排解读。最后，第 6.3 节 回到“初始化质量、外点率、重叠率、算力预算”四个条件变量，结合第 3.1 节 到第 5.5 节 的数据表，总结局部 ICP、鲁棒 ICP、全局初始化、学习型对应以及硬件加速各自在哪个前提下成立、又在哪个环节先失效。

6.1 典型应用场景 (Typical Application Scenarios)

ICP 在不同系统里承担的角色并不相同。自动驾驶把它放在连续帧里程计闭环中，首要矛盾是“每帧能否在采样周期内完成”；工业检测把它放在 CAD 与扫描件之间，首要矛盾是“局部精修能否稳定收敛到同一个装夹基准”；机器人抓取和医疗导航则进一步要求对遮挡、对称性和失败风险作出显式处理。因此，本节不再笼统描述“ICP 被广泛应用”，而是按场景补齐部署条件、公开实验和失效链条，并把相关方法与第 3 章 、第 4 章 和第 5 章 的技术选择对应起来。

6.1.1 自动驾驶：LiDAR 里程计与建图

自动驾驶前端优先选择点到面残差、局部地图配准和增量数据结构，不是因为这些方法在所有基准上都更优，而是因为车载 LiDAR 以 10 Hz 到 100 Hz 连续输出点云，配准必须在固定采样周期内完成；一旦单帧时延超过预算，后续去畸变、建图和控制都会级联失效。因此，这一场景最看重的是“同等精度下每帧处理时间”和“在退化场景中能否维持可观测性”，而不是单次离线对齐的最低 RMSE。

LiDAR 里程计（LiDAR Odometry）：以相邻两帧或当前帧与局部地图之间的配准估计瞬时位姿，再经积分得到轨迹。LOAM [93] 的关键设计不是“用了 ICP”，而是先把全量点云压缩为角点和平面点，再把里程计和建图拆成两个频率不同的线程。原文在 KITTI 里程计基准上使用 10 Hz Velodyne 数据，给出 39.2 km 总行驶距离上的平均位置误差 0.88%；其里程计线程约 10 Hz 输出，建图线程约 1 Hz 输出。这组数字说明 LOAM 成

立的前提是结构化道路、较高重叠率和可稳定提取的边/面特征；如果特征提取先退化，后面的局部 ICP 就会失去足够约束。

FAST-LIO2 [42] 进一步把“先提特征再配准”的依赖拿掉，改为直接把原始点注册到 ikd-Tree 维护的局部地图中。它之所以能这么做，是因为第 4.1 节 中的 ikd-Tree 把增量插入、删除和近邻查询压到了可实时的量级。原文在 19 个公开序列上比较多种 LiDAR-IMU 里程计，报告 FAST-LIO2 在其中 17 个序列上精度最佳；在大场景下，整套前端和建图可达到 100 Hz，上 Intel 处理器时每帧总处理时间约 1.82 ms，在 ARM 处理器上约 5.23 ms；针对实飞数据，平均每帧 2.01 ms，仍能承受 912–1198 deg/s 的快速翻转。这类直接法的优势在于弱特征场景下仍可利用稠密局部几何；但它依赖 IMU 去畸变和局部地图质量，一旦 IMU 外参、时间同步或法向近似先出错，点到面残差会先被系统偏差污染。

隧道、地下停车场等弱 GPS 场景：这类环境以平面和长走廊为主，配准先坏的多半不是最近邻搜索，而是可观测性。当前帧大部分法向垂直于行驶方向时，点到面约束对“沿隧道轴平移”几乎不给信息，优化会先在该方向产生漂移。第 3.5 节 已经说明这种退化与 Hessian 特征值塌缩直接相关，因此工程上多把 IMU、里程计或地图先验并入状态估计；若场景更接近大尺度平面分块，NDT [25] 也会被用来替代纯最近邻 ICP，因为它用体素分布而不是单点对应约束局部表面。但 NDT 自己依赖体素分辨率，一旦分辨率和场景尺度不匹配，局部极值同样会增多。

三维地图构建（3D Mapping）：局部里程计只能保证短时间一致性，长时序地图需要回环检测把远距离误差重新拉回。此时“描述子 + ICP 精修”的两阶段流程比直接对整帧做局部 ICP 更合理，因为第一阶段负责把误差压入收敛域，第二阶段才负责毫米到厘米级精修。FPFH [9] 的价值在于特征计算量相对 PFH 下降约 75%，适合 CPU 上快速生成粗对应；FCGF [56] 则在 3DMatch 上把平均 Registration Recall 做到 0.82，32 维描述子在 FMR 指标下达到 0.952 ± 0.029，5 cm 体素时特征提取约 0.17 s/fragment，并在 KITTI 上以 RTE 4.881 cm、RRE 0.170°、成功率 97.83% 支撑后续 RANSAC+ICP 精修。这里的局限同样明确：若回环候选来自重复立面或低重叠片段，描述子先给出偏置候选，后续 ICP 只会把错误解精修得更稳定。

6.1.2 工业检测：CAD 对齐与表面质量检测

工业制造中，质量检测的核心问题是将扫描获得的实测点云与 CAD 设计模型对齐（CAD-to-Scan Registration），再计算逐点偏差来发现制造缺陷。

工业检测与自动驾驶的差别在于：前者很少需要从完全未知位姿开始搜索，更多时候已有治具、标靶或坐标测量机给出的粗位姿，因此局部精修是否稳定比全局入盆能力更重要。换言之，这里关心的不是“能否扛住 70% 外点”，而是“在高重叠、低噪声、初值较准的条件下，每次装夹都能否回到同一个配准结果”。

ICP 在工业检测中的特殊要求（典型表述为量级需求，具体数值随行业标准与传感器配置而变）：

- **精度**：工业检测普遍要求亚毫米级配准误差，并强调可重复性；在低噪声与高重叠条件下，点到面约束更容易达到此类精度目标。
- **初始化**：CAD 模型和实测点云之间的初始位姿在很多产线中已由夹具或测量坐标系给出，ICP 只需局部精修，第 3.6 节 中的全局初始化一般不是主耗时。
- **外点处理**：实测点云可能包含工装夹具、支撑结构等“非被测体”点，这时先坏的是对应排序而不是闭式位姿更新，因此常把 TrICP 或 M-估计量放在对应残差筛选环节中。[14] 在 Frog 数据上给出一个典型例子：在约 3000 点、70% 重叠的条件下，TrICP 88 次迭代取得 MSE 0.10、耗时 2 s，而标准 ICP 45 次迭代仍停在 MSE 5.83、耗时 7 s。换言之，只要重叠率估计合理，截断式 ICP 足以覆盖“夹具点少量污染主件”的场景；但若真实重叠率估计过低，TrICP 会先把有效约束一起裁掉。
- **非刚体变形**：薄壁件扫描时可能发生局部弹性变形（装夹应力），严格刚体 ICP 会在变形区域产生系统性偏差。对于此类情况，需引入非刚体配准（如 BCPD）或分区域刚体配准策略，已超出本综述范围。

代表性工业软件：GOM Inspect、PolyWorks、ZEISS CALYPSO 等商业软件多以点到面 ICP、局部特征过滤和多阶段粗精配准作为内核，但公开文档很少披露统一协议下的配准数值，因此本节不把这些产品写成“基准结果”。可核查的公开证据主要来自方法论文和公开案例，而不是厂商手册。

6.1.3 机器人操作与抓取：位姿估计

六自由度抓取把 ICP 放在感知链的末端，因此它的任务不是从零完成目标识别，而是在检测、分割或粗定位之后把位姿误差压缩到控制器可接受的范围。之所以还需要 ICP，是因为抓取点常对姿态误差极敏感；只靠检测网络输出的 6D pose，末端执行器经常仍会在尖边、孔洞或对称面附近累积几度误差。

流程：以 RGB-D 传感器获取工件的深度图，生成工件可见表面的点云，与工件 CAD 模型的 ICP 配准直接输出变换矩阵（即工件在相机坐标系的位姿），机器人控制器以此规划抓取路径。

挑战：

- **部分可见**：机器人多只观测到工件的局部表面，低重叠率会显著缩小 ICP 的收敛盆。此时不能直接套用工业检测那套“已知初值 + 高重叠”假设，而要先用第 3.6 节 的粗配准把误差压回局部可收敛区域。以合成对象配准为例，RPM-Net [19] 在 ModelNet40 部分可见且含噪的设置下，把 isotropic rotation / translation error 降到 1.712° / 0.018；同一设置下 DCP 在 PREDATOR 论文复现实验中为 11.975° / 0.171，说明软对应和显式外点槽对“只看到局部表面”的任务更合适。若进一步转向真实低重叠室内片段，第 3.7 节 中的 GeoTransformer [57] 在 3DLoMatch 上可把 RR 提到 75.0%，比依赖随机采样的旧式前端更稳定。
- **堆叠工件（Bin Picking）**：多个工件堆叠时，ICP 可能将相邻工件的点云误认为同一工件。深度学习实例分割先将工件分离，再对每个实例做 ICP，是当前最有效的工程方案。
- **位姿不确定性**：安全关键应用（如协作机器人，共融机器人）需要位姿估计的置信区间，而不仅是一组点估计。第 3.4 节 中的 Stein ICP [46] 在 RGB-D 对称物体和稀疏 LiDAR 场景里使用 100 个粒子近似后验，KL 中位数约 0.6–5.7、OVL 约 0.7–0.9，运行时间约为 Bayesian ICP 的 1/8 到 1/5。它解决的是“多解并存时该不该相信当前估计”的问题；但代价也清楚，粒子后验的计算量比常规闭式 ICP 高得多，不适合无条件放进所有抓取循环。

6.1.4 医疗：手术导航与术中配准

手术机器人（如达芬奇系统的改进型）和计算机辅助手术（CAS）需要将术前 CT/MRI 影像与术中实时扫描对齐——即” 术中配准”（Intraoperative Registration），这是 ICP 在医疗领域的核心应用。

脊柱外科导航：术前 CT 提供椎骨三维模型，术中以光学追踪仪扫描椎骨表面获得稀疏点云，二者的 ICP 配准用于估计当前解剖标志的位置。这里的前提是椎骨局部可近似为刚体，因此点到面约束仍有效；但出血、残留组织和器械遮挡会先破坏术中表面提取，再把错误点送入对应搜索，所以常与第 3.2 节 中的鲁棒估计联合使用。若遮挡使可见表面过少，局部精修就会先因法向约束不足而失稳。

软组织手术（如肝脏）：软组织在呼吸、触压下发生非刚体变形，刚体 ICP 多半难以满足精度与一致性要求。此类场景需要非刚体配准或显式的形变建模；即便如此，刚体配准仍常作为非刚体优化的初始化环节。

医疗配准的特殊性能要求：可重复性、不确定性量化与安全失败模式。这里最重要的不是“平均误差更小”，而是当解出现多模态时系统能否拒绝输出错误位姿。Stein ICP 正适合承担这个角色，因为它明确输出变换后验而不是单点估计；相反，标准 ICP 即使给出很小残差，也可能只是沿某个对称方向滑入了错误局部极小。

6.1.5 场景-算法匹配总结

应用场景的差异决定了“先解决什么问题”。自动驾驶先解决时延和退化方向，工业检测先解决重复精度与夹具外点，机器人抓取先解决部分可见和实例混叠，医疗配准先解决错误解告警。由此可见，第 3 章 的方法差异并不是抽象的目标函数差异，而是不同失败模式的应对路径。第 6.2 节 将进一步说明，为什么这些场景不能用同一组数据集和阈值做简单并排比较。

6.2 标准数据集与基准测试 (Standard Datasets and Benchmarks)

跨方法比较的前提不是“都在做点云配准”，而是“使用了同一数据集、同一阈值、同一成功判据”。一旦训练集构造、点云下采样、RANSAC 采样数或成功阈值不一致，表面上接近的 RR、RTE 或 RMSE 就无法互相

表 30: 第 6.1 节应用场景中的代表性公开结果。表中数值来自各原论文或第 3 章已整理的数据表，用于说明“何种条件下该方法被采用”，不构成跨协议统一排行。

场景	代表方法	原文场景与条件	指标	代表性数值	方法成立前提
车载 LiDAR 里程计	LOAM [93]	KITTI 10 Hz Velodyne，39.2 km，道路场景	平均位置误差	0.88%	可稳定提取边/面特征，帧间
LiDAR-IMU 紧耦合建图	FAST-LIO2 [42]	19 个公开序列；Intel/ARM；局部地图 + ikd-Tree	最佳序列数、每帧处理时间、频率	17/19 序列精度最佳；1.82 ms/scan（Intel），5.23 ms/scan（ARM）；最高 100 Hz	IMU 去畸变可靠，局部地图连续
回环粗配准	FCGF [56]	3DMatch 5 cm 体素；KITTI RANSAC 后端	FMR、RR、RTE/RRE	3DMatch RR 0.82；FMR 0.952 ± 0.029；KITTI RTE 4.881 cm、RRE 0.170°、成功率 97.83%	片段间仍有可学习几何一致
工业局部精修	TrICP [14]	Frog 数据，约 3000 点，70% 重叠	MSE、时间	MSE 0.10，2 s；标准 ICP 为 MSE 5.83，7 s	重叠率可估，初值已在局部
抓取局部可见配准	RPM-Net [19]	ModelNet40 部分可见 + 噪声，约保留 70%，717 点	isotropic rotation / translation error	1.712° / 0.018	训练分布与部署几何相近
安全关键位姿估计	Stein ICP [46]	RGB-D 对称物体 + 稀疏 LiDAR，100 粒子	KL、OVL、运行时	KL 中位数约 0.6–5.7；OVL 约 0.7–0.9；比 Bayesian ICP 快 5× 以上	需要显式后验而不是单点估

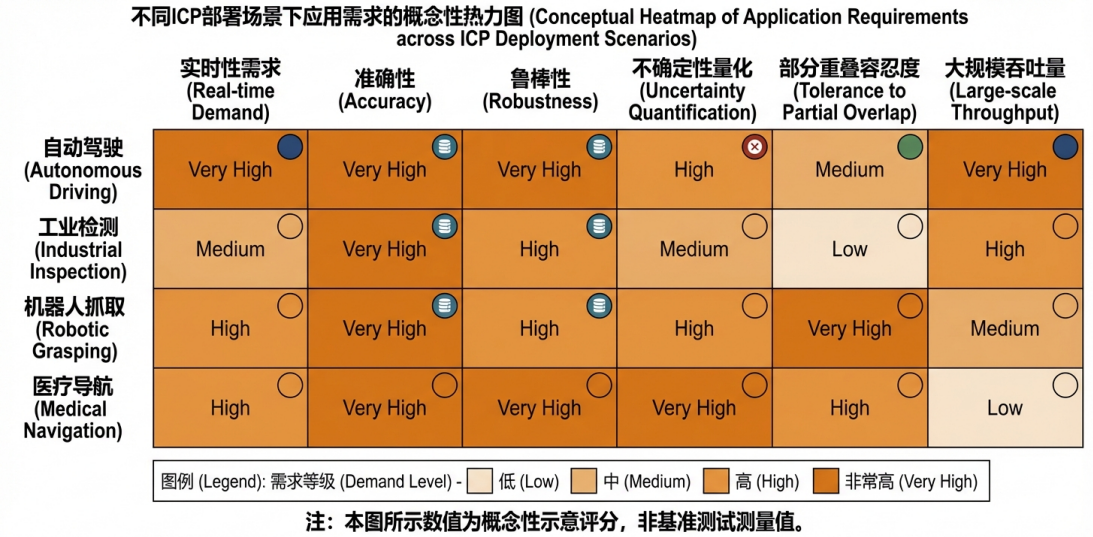


图 68: ICP 四大应用场景（自动驾驶、工业检测、机器人抓取、医疗手术）在六个关键特性（实时性、精度、鲁棒性、不确定性量化、部分重叠耐受、大规模点云）上的需求热图。该图按学术机制示意绘制，用于表达相对需求强弱，不对应统一基准下的具体测量数值。

表 31: 四大应用场景的 ICP 变体和加速方案推荐总结，包括关键约束条件。

应用场景	推荐 ICP 变体	推荐加速方案	关键约束
自动驾驶 LiDAR 里程计	P2Pl ICP + 紧耦合里程计框架（如 FAST-LIO2）	ikd-Tree/并行化，必要时硬件加速	严格延迟预算、大规模点云吞吐
工业 CAD 对齐	GICP 或 P2Pl + 鲁棒估计	高精度数据结构与稳定法向估计	亚毫米级误差控制、可重复性
机器人 6-DOF 抓取	全局粗配准 + ICP 精修	多线程 + 合理降采样	低重叠与遮挡、实例混叠风险
医疗术中配准	概率 ICP / 不确定性量化框架	可靠性优先的实现与验证	安全失败模式、不确定性告警

解释。因此，本节先补齐主流基准的数据规模和协议，再讨论它们各自缺什么，而不是直接把不同论文的最优数字拼在一起。

6.2.1 室内 RGB-D 数据集

TUM RGB-D 数据集提供室内 RGB-D 序列与地面真值轨迹，是 RGB-D SLAM 与基于深度图的局部配准常用测试集之一 [40]。原始 benchmark 含 39 个序列，Kinect 以 640×480、30 Hz 采样，动捕系统提供 100 Hz 真值轨迹。它适合检验近距离、小视场条件下的局部精配准稳定性；但由于视距短、点云范围有限，不能直接外推到车载 LiDAR 的大尺度稀疏场景。

ScanNet 数据集提供大规模室内 RGB-D 扫描序列与重建场景，是室内三维理解与片段级配准任务常见的数据来源之一 [94]。它的价值不在于“配准协议已经标准化”，而在于提供跨房间、跨遮挡的大量真实重建片段，使学习型方法可以从中再构造局部片段对。也正因为如此，ScanNet 更接近上游数据源而不是直接可比较的配准榜单；若论文只写“在 ScanNet 上测试”，还必须继续交代如何抽取片段、如何生成真值和采用何种重叠阈值。

Stanford 3D 扫描库（Stanford 3D Scanning Repository）包含 Bunny、Dragon、Happy Buddha 等经典扫描模型，常用于展示高重叠、低噪声条件下的局部配准精度上限。[1] 的早期曲面实验以及 [13] 的采样/收敛研究都依赖这类对象级数据。它们适合解释目标函数和采样策略的差异，但不适合代表真实传感器噪声、动态外点和长时序漂移。

3DMatch 数据集是学习型局部特征与配准方法的核心评测基准之一：从室内 RGB-D 重建中构造点云片段对，并提供训练与评估用的配准真值 [95]。PREDATOR [64] 进一步把官方 3DMatch 中重叠率大于 30% 的样本之外的片段对单独整理为 3DLoMatch，仅保留 10%–30% 重叠的困难样本。这个划分很重要，因为很多方法在 3DMatch 上已经接近饱和，但在 3DLoMatch 上仍会明显掉点。

评测指标（不同工作阈值设置差异较大，横向比较时需同时报告阈值）[95]：

- **FMR（Feature Matching Recall）**：特征匹配召回率，反映描述子质量与匹配稳定性。
- **IR（Inlier Ratio）**：对应集中满足几何一致性阈值的比例。
- **RR（Registration Recall）**：配准变换误差落在给定旋转/平移阈值内的比例。

6.2.2 室外 LiDAR 数据集

KITTI 里程计数据集是自动驾驶与移动机器人领域的标准基准之一，提供车载多传感器数据与里程计评测协议 [5]。在点云配准语境中，KITTI 常被用于评估室外驾驶序列上的帧间扫描匹配与里程计漂移，常见报告口径是相对平移/旋转误差。更具体地说，LOAM 使用的 KITTI odometry 数据以 10 Hz Velodyne 记录，配准结果由 benchmark 服务器按 100 m 到 800 m 的轨迹段统一打分 [93]。因此，KITTI 更适合衡量“前端累计误差会不会在道路尺度上扩散”，而不适合替代片段级 RR/FMR 这类局部配准指标。

KITTI 的主要局限：

- 场景较” 规整”（城市道路），缺少隧道、森林等非结构化场景。
- 以车载多线 LiDAR 为主，对低线数、稀疏点云与极端遮挡的覆盖不足。
- 无动态物体标注，动态物体（行人、车辆）引入的外点对 ICP 的干扰未被单独量化。

ETH 数据集是室外静态场景的三维点云配准基准之一，常用于比较不同 ICP 变体在噪声、外点与部分重叠条件下的精度与鲁棒性 [23]。它的价值在于场景固定、协议清楚，适合比较 P2P、P2Pl、GICP、TrimmedDist 和采样策略的组合；局限也同样明显，即动态外点极少，无法反映自动驾驶中的车辆和行人干扰。

M2DGR 数据集提供多传感器、多场景的 SLAM 数据序列与地面真值，是近年来用于检验算法跨场景泛化能力的代表性数据集之一 [96]。[76] 在自适应降采样评测中同时使用了 M2DGR 和 KITTI，反映了评测从单传感器、单场景向多模态、多场景综合比较的趋势。

更大规模与长时序数据集常用于检验跨天气、跨光照与跨传感器配置的稳定性：nuScenes [97] 与 Waymo Open Dataset [98] 提供多模态车载传感器套件；Oxford RobotCar [99] 与 MulRan [100] 覆盖更长周期与更复

杂城市变化；手持/近距离建图的 Newer College 数据集 [101] 则更接近轻量移动平台的扫描匹配负载。这类数据集不一定直接提供“点云对点云”的配准真值，但对工程系统的端到端鲁棒性评估更贴近现实。

6.2.3 合成数据集与点云配准专用基准

ModelNet40 最早随 3D ShapeNets 工作推广为对象级 3D 形状基准之一 [102]。DCP [18] 和 RPM-Net [19] 等方法都把它作为对象级刚体对齐的标准起点：前者在 12,311 个 CAD 模型上随机采样 1024 点，后者在部分类别不重叠、加噪和部分可见设置下继续扩展协议。它的优势是可控真值和可控噪声；缺点是局部几何过于干净，外点和遮挡是人工注入的，和真实 RGB-D 或 LiDAR 片段仍有明显分布差异。

Redwood/片段级配准基准：室内重建数据常被切分为局部片段并构造片段对评测（与 3DMatch 的设置相近），Redwood 系列数据与其衍生评测在开源实现中被广泛采用 [103]。其优势是可直接检验” 低重叠 + 遮挡 + 噪声” 下的局部配准与鲁棒对应模块。

PCL 基准（Pomerleau et al., Autonomous Robots 2013 [23]）：首个系统化的 ICP 变体对比基准，在 ETH 场景上评测约 30 种组合（对应方法 × 外点处理 × 误差最小化）。其价值不只在于给出优劣排序，而在于证明三点：对应建立方式常比局部最小化器更影响结果；降采样和搜索结构常比“再换一个损失函数”更影响总时延；场景一旦变化，最优配置也随之改变。这也是第六章坚持把“条件”写在“结果”前面的原因。

表 32: 第 6.2 节主流基准的任务边界汇总。表中信息用于说明“不同数字来自什么协议”，避免把片段级 RR、里程计漂移和对象级 RMSE 混成同一类结果。

数据集/协议	典型场景	原文规模或划分	常用指标/阈值	适合回答的问题	不足
TUM RGB-D [40]	室内 RGB-D	39 序列；640×480@30 Hz；真值 100 Hz	ATE/RPE、局部对齐误差	小场景精配准是否稳定	点云范围短，不代表车载稀疏场景
ScanNet [94]	大规模室内扫描	大规模室内重建序列，常被二次切片	论文自定义片段级指标	学习型方法是否能从真实室内重建中泛化	不是单一固定配准协议
Stanford 3D	对象级高重叠扫描	Bunny/Dragon/Buddha 等	RMSE、成功率	局部目标函数和采样策略的上限表现	噪声与外点过少
3DMatch [95]	室内片段级配准	62 场景；PREDATOR 使用 46/8/8 划分	FMR、IR、RR	学习特征和片段级全局配准是否有效	只覆盖 >30% 重叠的标准样本
3DLoMatch [64]	低重叠室内片段	从 3DMatch 中抽取 10%–30% 重叠对子	FMR、IR、RR	低重叠前端是否还可靠	场景仍局限于室内 RGB-D
KITTI odometry [5]	车载道路 LiDAR	10 Hz Velodyne；按 100–800 m 轨迹段评分	相对平移/旋转误差，端到端漂移	前端里程计是否能在长距离驾驶中稳住漂移	动态外点未单独量化
ETH/PCL benchmark [23]	室外静态点云对	固定六场景，约 30 种 ICP 组合	时间、收敛质量	传统 ICP 模块化取舍	动态外点和硬件约束缺失
nuScenes [97]	大规模车载多模态	多传感器、长时序场景	多任务指标，配准常需自定义协议	跨天气、跨传感器鲁棒性	配准协议分散，难直接横比

6.2.4 评估指标的标准化

点云配准的评估指标尚未完全统一，不同论文采用不同指标造成比较困难：

变换精度指标：

$$\mathrm{RMSE}_T = \sqrt{\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \|e_i\|^2} \tag{81}$$

其中 e_i 为第 i 个测试对的变换误差（可以是欧氏位移误差或旋转角度误差）。

配准召回率（Registration Recall，RR）：

$$\mathrm{RR} = \frac{|\{i : \|e_i\| < \tau\}|}{n_{\text{test}}} \tag{82}$$

τ 的选取决定了 RR 的严格程度；不同基准与不同论文常采用不同的阈值组合，必须与 RR 同时报告，否则 RR 无法互相比较 [95]。

RTE/RRE（3DMatch 系列常用）：令估计变换为 (R, t)、真值为 $(R^{\cdot}, t^{)}$，则平移误差 $\mathrm{RTE} = \|t - t^{\|_2}$；旋转误差常定义为 $\mathrm{RRE} = \arccos((\mathrm{tr}(R^{\top} R^{)-1)/2)}$，多数论文再把它转成角度表示 [95]。

时间指标：

- 单帧延迟（Latency）：测量从输入点云到输出变换的总时间，包含预处理、对应搜索、优化全流程。
- 吞吐量（Throughput）：单位时间处理的点对数量（对并行系统更重要）。
- 首帧响应时间 vs 稳态帧率：流式处理系统（LiDAR 里程计）中，稳态帧率（包含 Pipeline 重叠）更具参考价值。

内存指标：片上 BRAM 用量（FPGA）、参数量（深度学习方法）、运行时峰值内存（CPU/GPU）。

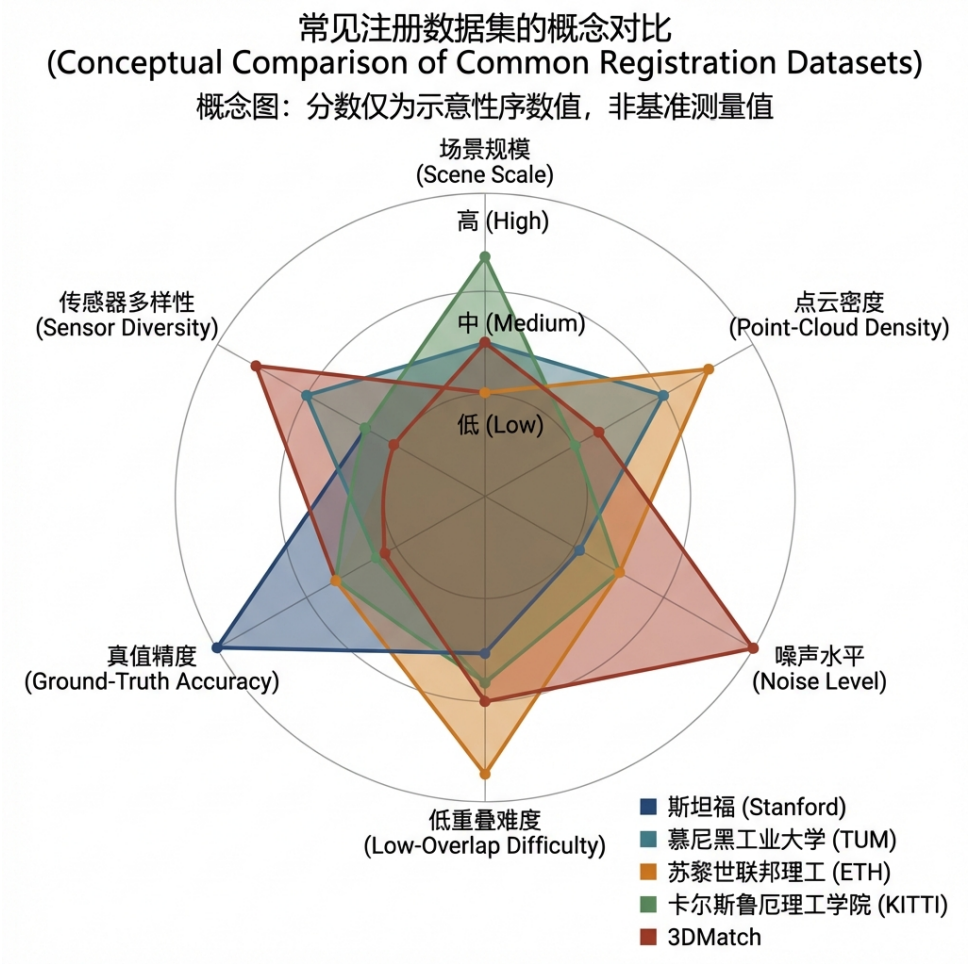


图 69: 五个常用点云配准数据集（Stanford、TUM RGB-D、ETH、KITTI、3DMatch）在六个特性维度（场景规模、点云密度、噪声水平、低重叠挑战、地面真值精度、传感器多样性）上的雷达图对比。该图按学术概念图绘制，用于帮助读者建立“数据集偏向什么困难”的直觉，不对应统一协议下的定量测量。

6.2.5 评估体系的局限性与展望

当前基准测试体系存在若干值得关注的局限性：

静态场景假设：绝大多数基准数据集以静态场景为主（如 Stanford、ETH），或只包含有限的动态干扰。真实交通与人机协作环境中，动态物体经常是外点的主要来源，但现有基准对动态外点的系统刻画仍不足。

传感器偏差：室外车载多线激光雷达与室内 RGB-D 的噪声模型、采样密度与遮挡模式差异显著。算法在一个数据集上的表现不一定能迁移到不同传感器类型。

硬件感知评估缺失：现有基准几乎全部聚焦于算法精度，忽略了硬件资源占用、功耗和延迟的综合评估。随着第 5 章 所述的专用硬件加速兴起，急需一套统一的“精度-效率-资源”综合基准框架，将算法精度与硬件实现效率联合评估。

[3] 的综述已指出上述问题，但目前仍缺乏一个统一的、跨传感器、跨场景、硬件感知的综合基准平台。这一缺口会直接影响第六章的解读方式：任何只在单一数据集上成立的优势，都不该被外推成“通用结论”。相关开放问题将在第 7 章 继续讨论。

6.3 方法横向比较 (Cross-Method Comparison)

横向比较的关键不是在一张表里罗列尽可能多的方法，而是在统一的约束与评测语境下明确取舍：同一配准任务在初始化质量、外点率、重叠率与算力预算不同的条件下会呈现完全不同的有效域。本节以“任务条件 → 方法族 → 代表性证据 → 失效前提”为主线，总结第 3 章 到第 5 章 中各类技术的互补关系。

6.3.1 算法变体的综合对比

不同论文报告的 RMSE、RR 多半来自不同数据集、不同阈值与不同实现细节，直接拼接到同一张数值表会产生误导。因此, 这里只保留“同类任务下能互相解释”的数字, 并显式写出它们来自哪一类协议, 例如 ETH/PCL 基准、3DMatch/3DLoMatch 或 ModelNet40 [23][95]。需要做定量对比时，最小可行做法是固定公开基准、复用同一实现框架并明确超参数搜索范围；否则，实现差异经常会压过算法差异 [23]。

表 33: 主要方法族的横向对比。表格刻画的是典型取舍与适用语境，而非跨论文可直接对比的统一数值。

方法族	代表方法	代表性数据与数值	主要收益	典型代价	常用评测语境
基线局部 ICP	P2P ICP [1]	经典曲面例子中 RMS 0.59；ETH/PCL 中位时间约 1.45 s 第 3.1 节 6	实现简单，作为局部精修基线	收敛盆小，对外点敏感	高重叠、初值较好
几何约束增强	P2Pl ICP [2]、GICP [26]	GICP 以 20 近邻估计协方差，50 次迭代上限；30 m 间隔实测扫描仍可稳定配准 第 3.4 节 12	旋转与切平面约束更强	依赖法向/协方差估计质量	结构化几何、法向可信
鲁棒化	TrICP [14]、FRICP [33]	TrICP：Frog 上 MSE 0.10 vs ICP 的 5.83；FRICP：Bunny 上 0.34 s、RMSE 0.85/0.69×10^-3 第 3.2 节 8	提升外点与部分重叠耐受	额外超参数/计算开销	外点率较高、部分重叠
收敛加速	AA-ICP [17]	TUM RGB-D + Bunny 上中位加速约 35%，误差中位数改善约 0.3% 第 3.3 节 10	降低迭代次数	对噪声/非线性更敏感，仍需入盆	初值较好、迭代成本高
全局初始化 + 精修	FPFH+FGR+ICP [28]	UWA benchmark 0.05-recall 84%；FGR 在合成 range 数据噪声 σ=0.005 时平均 RMSE 0.008 第 3.6 节 16	扩大可用初始化范围	额外特征与全局优化开销	大初始误差、易局部极小
学习化对应	DCP [18]、RPM-Net [19]、GeoTransformer [57]	DCP：ModelNet40 RMSE(R) 3.150°、RMSE(t) 0.0050；RPM-Net：部分可见 + 噪声 1.712°/0.018；GeoTransformer：3DLoMatch RR 75.0% 第 3.7 节 18	低重叠或复杂扰动下更鲁棒	训练、域偏移与部署成本	对象级合成 / 室内片段对 / 低重叠片段

关键洞察:

1. **初始化决定可用范围**：在大初始误差或低重叠场景下，全局初始化比局部目标函数的微调更关键；它的作用是把问题拉回 ICP 的收敛盆内，而不是单纯提高局部最优的数值精度。FGR、TEASER++ 和 GeoTransformer 的收益都属于这一类，相关数据见第 3.6 节 16和第 3.7 节 18。

2. **鲁棒化经常是最先该加的一层保护**：截断、核函数或 GNC 等机制能直接处理“对应里混入了坏点”这一问题。它们解决的不是初值，而是偏置残差，因此在已入盆但外点较多时收益最直接；若初值还没进盆地，再强的鲁棒核也只能稳定地收敛到错误局部解。第 3.2 节 8里的 TrICP 和 FRICP 数据正反映了这一点。

3. **学习化方法的核心风险是域偏移与评测协议差异**：对象级合成任务与真实扫描在噪声、采样密度与遮挡模式上差异显著；横向对比时必须同时说明训练数据、测试数据与阈值设置。第 6.2 节 已经说明 ModelNet40、3DMatch 和 KITTI 回答的并不是同一个问题。

4. **GICP 的优势在于建模统一性**：它以局部协方差统一 P2P 与 P2Pl 的误差结构，在结构化场景中常具备较稳健的收敛行为；但它依赖协方差估计和近邻选择，若局部平面假设不成立或法向估计受噪声污染，收益会先在对应建模环节消失，相关前提见第 3.1 节 6和第 3.4 节 12。

6.3.2 软件优化的叠加效果

第 4 章 的软件优化策略并非互斥，工程中常按“先降规模、再降常数”的顺序叠加：先用降采样或多分辨率降低问题规模，再用数据结构、并行化与近似最近邻降低常数项开销 [23][80]。

表 34: 软件优化叠加的典型路径与取舍（定性总结）。

优化策略	主要收益	典型代价/风险
降采样（体素/法向空间/自适应）	显著降低对应搜索次数	可能削弱细节约束，改变收敛盆
更快的数据结构（KD 树变体/体素哈希）	降低最近邻常数项	动态更新与内存局部性权衡
并行化（SIMD/OpenMP/GPU）	提升吞吐并降低延迟	并行效率受内存访问模式限制
近似最近邻（ANN）	在可控误差下加速	近似误差可能引入系统性偏置
多分辨率/粗到细	扩大收敛盆、降低总体迭代	参数较多，调参成本上升

软件优化常能在不改变硬件的前提下实现数量级加速，尤其是在最近邻搜索与数据布局上；但其收益高度依赖点云规模、内存层级与场景结构，评测应明确软硬件配置与实现细节，参见第 6.2 节 。

6.3.3 硬件加速路线的终端效果

结合第 5 章 的数据, 专用加速器的收益主要来自对最近邻搜索与相关矩阵构建的结构化重写, 尤其是第 5.1 节 和第 5.5 节 所示的数据通路重排。

表 35: 典型硬件路线的代表工作与取舍（定性总结）。

路线	代表工作	主要优势	主要代价
FPGA	多模式对应搜索加速器 [91]	高能效、可重配置	开发成本与验证周期较长
ASIC	Tigris 专用处理器 [78]	极致延迟与功耗	灵活性低、设计前期需充分定型
PIM	PICK SRAM-PIM [20]	减少数据搬运，能效高	受存储阵列结构约束，算法耦合更强

Pareto 视角的组合建议（以定性原则为主）：

- **移动机器人与通用平台**：优先软件优化与稳健的局部目标（如 P2Pl/GICP），在满足延迟约束后再考虑硬件投入。
- **嵌入式与严格能耗预算**：当软件优化仍无法满足约束时，再引入 FPGA/PIM/ASIC 等路线，并在原型阶段尽量冻结数据结构与计算图，详见第 5 章 。

6.3.4 算法-硬件协同选择原则

综合第 3 章 到第 5 章 的讨论，可把选型原则压缩为下面四条：

原则一：从应用约束出发，反向确定技术路线。先明确最严格的约束（是延迟、功耗、精度还是成本），再选择能满足该约束的最简单方案，而非追求理论最优。

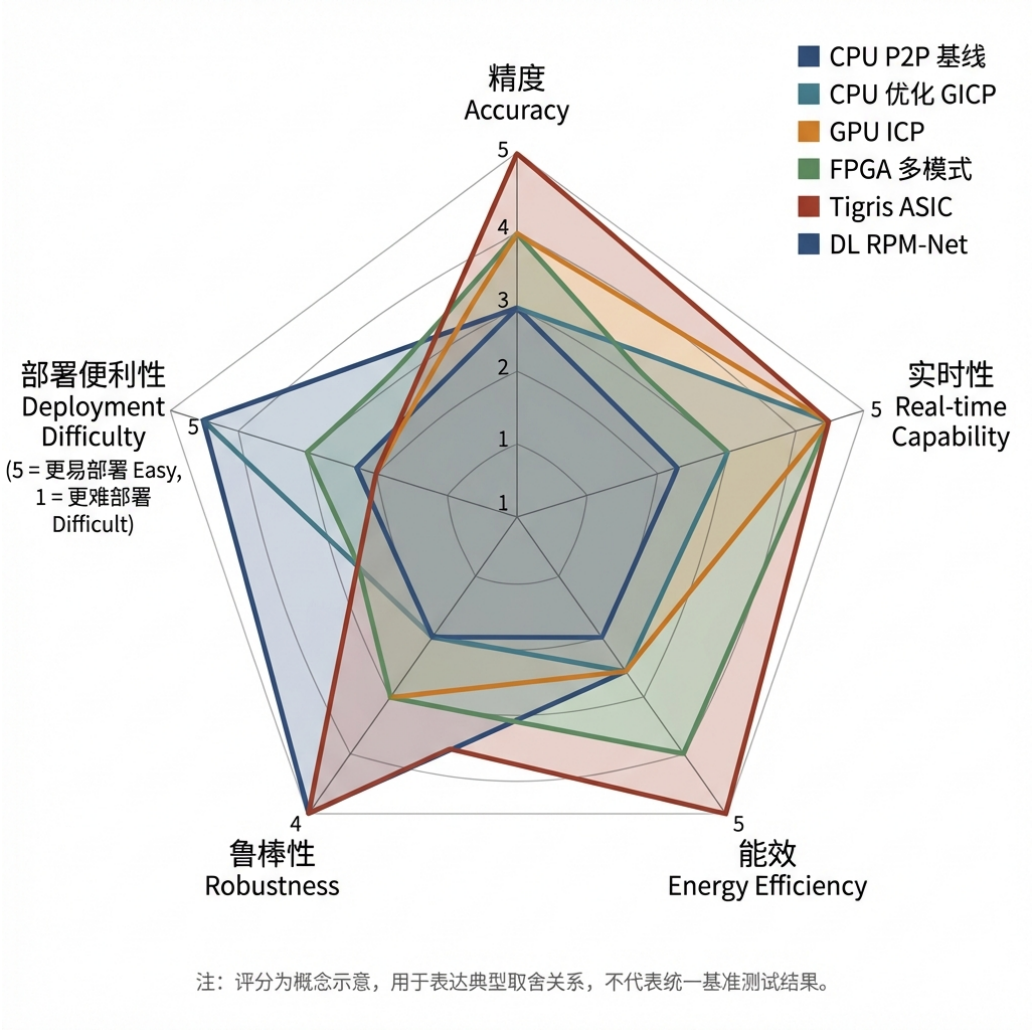


图 70: 六种 ICP 配置在五个维度（精度、实时性、功耗效率、鲁棒性、部署难度）上的雷达图全景对比。配置：(1) CPU 基线 P2P、(2) CPU 优化 GICP、(3) GPU ICP、(4) FPGA 多模式、(5) Tigris ASIC、(6) 深度学习 RPM-Net。每个维度 1–5 为概念性评分，用于呈现典型取舍，而非严格基准测试结果。

原则二：优先软件优化，再考虑硬件。降采样、近似最近邻与并行化常能提供数量级加速；只有当软件优化后仍不满足约束时，才值得投入 FPGA/ASIC 等硬件开发。

原则三：全局初始化是鲁棒性的保险。当初值质量不可控或低重叠频繁出现时，在 ICP 前加入全局粗配准 (第 3.6 节) 能显著扩大可用范围；是否采用取决于系统对延迟与失败率的容忍度。

原则四：硬件设计与算法深度绑定。FPGA/ASIC 方案一旦确定数据结构（KD-Tree、RPS 或球形桶等），后续算法变更成本极高。应在 FPGA 原型阶段充分验证算法选型，再考虑 ASIC 流片。

这些原则把“方法优劣”重新还原成“条件匹配”。真正需要避免的不是选错某一个缩写，而是在没有写清初值、重叠率和时延预算的情况下讨论优劣。第 7 章 将继续讨论这些条件在动态场景、跨传感器泛化和硬件感知基准中的未解部分。

6.4 本章小结

应用、基准与横向比较三条主线共同回答了一个核心问题：前几章提出的技术，在真实约束下是否能达到可测量的系统目标。

第 6.1 节 表明，ICP 的部署场景决定了变体选型的第一优先级。自动驾驶场景首先受端到端延迟和退化方向约束，因此 LOAM、FAST-LIO2 这类方法把特征筛选、局部地图和增量数据结构放在前面；工业检测场景首先受重复精度和夹具外点约束，因此更依赖高重叠下的局部精修与截断式鲁棒机制；机器人抓取首先受部分可见和实例混叠约束，因此要先做粗配准或学习型对应，再做局部 ICP；医疗导航则把失败告警和不确定性表达放在首位，这正是 Stein ICP 这类概率方法存在的理由。第 6.2 节 进一步说明，TUM RGB-D、3DMatch、KITTI、ETH 与 nuScenes 分别回答的是不同问题，任何单一数据集上的性能数字都不能直接外推为通用结论。第 6.3 节 则把这些事实收束为一个选型框架：初始化质量、外点率、重叠率与算力预算四个条件变量共同决定了不同方法族的有效域。

本章也暴露出几个仍未被系统解决的缺口：动态场景中的结构化外点如何统一建模，不可观方向何时应主动拒绝输出结果，跨传感器与跨场景的泛化应如何统一评测，以及硬件实现应如何与算法精度一起报告。第 7 章 将沿“失败模式 $\rightarrow$ 代表性对策 $\rightarrow$ 仍未解决的缺口”的顺序继续讨论这些问题。

7. 开放挑战与未来方向 (Open Challenges and Future Directions)

第 6 章 从应用需求出发，系统梳理了 ICP 在自动驾驶、工业检测、机器人操作与医疗图像四大场景中的部署逻辑，并通过公开基准与横向比较确立了方法选型的四维约束框架。评测分析的核心价值不仅在于展示” 什么方法在什么条件下更好”，更在于揭示现有技术体系在哪些条件下会**系统性失效**。初始位姿落在收敛盆外、动态物体形成结构化外点、几何退化导致不可观方向、实时约束与功耗预算在工程系统中同时收紧——这些失败模式在单一指标的基准评测中往往被平均效应掩盖，却是实际部署中最常见的障碍。本章将这些系统性挑战整理为” 失败模式 $\rightarrow$ 代表性对策 $\rightarrow$ 仍未解决的缺口” 的分析框架，目的不是给出最终答案，而是为未来研究划定最值得投入的方向边界。

7.1 初始化与全局粗配准：把问题拉回收敛盆

局部 ICP 的收敛性建立在” 初始位姿足够接近全局最优” 这一前提上，其可用范围由初始变换误差和点云重叠率共同决定（第 3.6 节 ）。当重叠率低于约 30%、或初始旋转误差超过几度时，ICP 极易陷入错误盆地，后续的目标函数迭代难以纠正这一偏差 [13]。工程中至少有三类典型场景会先触碰这一限制：机器人重启或长时间中断后需要重新定位，此时局部里程计的累积漂移已将初始估计推出收敛域；跨平台或跨时段扫描导致点云视角差异过大，重叠区域稀少而纯距离的对应建立质量极差；以及特征稀疏或重复结构场景中，全局描述子产生大量误对应，下游局部精修的起点本身就是错误的 [9][28]。这一系列场景推动了” 全局初始化 + 局部精修” 的两阶段范式，目标不是取代 ICP，而是先把问题拉回 ICP 的收敛盆。

基于几何描述子的两阶段流程是当前最普及的工程方案。FPFH 描述子通过编码局部点分布的法向量直方图为三十三维向量，计算量相比早期 PFH 下降约 75%，可在 CPU 上快速生成候选对应，再由 RANSAC 随机抽样剔除误对应以估计粗位姿 [9]。FGR 以 Geman-McClure 鲁棒核直接优化所有候选对应，规避了 RANSAC

的随机采样开销；在合成数据集最高噪声（$\sigma = 0.005$）的设置下，平均 RMSE 约 0.008，且在低重叠（约 21%）的 UWA benchmark 上保持 0.05-recall 为 84%[28]。对于无法可靠提取描述子的场景，4PCS 与 Super4PCS 通过搜索四点全等集实现无特征全局对齐，在点数较少时可给出可用粗位姿 [58][59]。

可认证鲁棒估计是另一条重要路线，目标是从数学上保证解的全局性。Go-ICP 用分支定界在完整 $SE(3)$ 空间搜索全局最优解，理论上消除了局部极值问题，代价是计算时间较长，实用性主要在点数较少的离线场景 [16]。TEASER/TEASER++ 通过截断最小二乘将外点剔除转化为鲁棒估计，并以半正定松弛提供可验证的最优性证书；在外点率高达 99% 的极端情况下仍能给出正确位姿，同时比 RANSAC 快一个数量级 [36]。SE-Sync 在同步旋转估计框架下提供凸松弛后验，在 SLAM 后端位姿图优化中实现可认证全局最优 [72]。

学习型对应方法的兴起进一步扩展了两阶段框架的低重叠适用范围。FCGF 用全卷积几何特征网络取代手工描述子，在 3DMatch 上以 32 维描述子取得 FMR 为 0.952 ± 0.029 的内点匹配率 [56]。PREDATOR 专门针对低重叠场景引入重叠得分预测，在 3DLoMatch（重叠率 10–30%）上将 Registration Recall 提升至 74.0%[64]。GeoTransformer 通过超点级几何编码与旋转不变位置嵌入，在 3DMatch 上内点率（IR@0.1m）达到 91.5%，并配合 LGR 实现无 RANSAC 的鲁棒估计 [57]。CoFiNet 采用从粗到细的分层对应策略，先在下采样超点建立语义对应再细化至点级，进一步压缩低重叠下的失败率 [63]。

然而，上述进展并未完全覆盖工程部署所需的能力。可认证方法（TEASER、SE-Sync）的计算代价和接口复杂度，使之难以与实时里程计和位姿图后端无缝集成；失败告警、置信度输出与下游系统策略之间仍缺乏标准化接口，可证鲁棒性与实时性之间的张力尚未被系统解决 [36]。学习型描述子的泛化能力严重依赖训练数据的多样性，在跨传感器（室内 RGB-D $\rightarrow$ 室外 LiDAR）或跨场景分布偏移下退化明显，评测协议需显式覆盖这些域外设置 [95][97]。当重叠率低于 10% 时，现有所有方法的失败率都急剧上升，但标准基准中尚无统一的极低重叠协议，方法改进难以被独立量化。此外，全局粗配准阶段产生的位姿不确定性（尤其是误对应导致的多模态假设）在下传给局部 ICP 时往往被丢失，精修阶段误认初值可信而提前收敛，如何将粗配准的假设集与不确定性一并传递给局部优化，是两阶段范式尚待打通的关键接口。

7.2 动态场景与语义信息：把” 外点” 当成可建模结构

经典 ICP 假设点云中的点来自同一静态刚体，但现实场景中行人、车辆、机械臂等动态物体会在两次扫描之间改变位置，在点云中留下” 结构化外点”。这类干扰与随机测量噪声有本质区别：它们空间上连贯（动态车辆形成大块连续区域）、时间上相关（同一目标在相邻帧中出现在不同位置），且往往具有方向性——快速横移的行人主要污染平移估计，旋转物体会引入虚假旋转分量。更隐蔽的危害在于，这种偏置会在里程计积分链路中持续累积，引发长期漂移，这与偶发误测量的零均值特性完全不同 [35]。

M-estimator 和截断策略（第 3.2 节 ）能缓解随机外点的影响，但面对结构化外点时效果有限。这类方法依赖” 内点多于外点” 的统计假设，而在城市道路等动态目标密集场景中，动态点可能占据相当大比例，该假设可能根本不成立。TrICP 的截断比例依赖对重叠率的准确估计，若重叠率估计本身受动态点干扰，截断策略会先把有效约束一起裁掉 [14][33]。

将语义信息引入对应建立是更直接的解法。Semantic ICP 通过类别一致性约束，只允许同类语义标签的点之间建立对应，从而将行人、车辆等动态类别天然排除在配准优化之外，在室外 LiDAR SLAM 任务中显著降低动态物体引起的轨迹偏差 [104]。将深度语义特征融入 NDT 框架的方法（Zaganidis 等）进一步表明，语义特征不仅对抗动态点，还能改善结构相似区域的重复性问题，提升对应的几何歧义性分辨能力 [105]。

物理量辅助配准在 FMCW LiDAR 普及后提供了另一条无需语义先验的路线。Doppler ICP 将径向速度残差并入 ICP 目标函数：静态结构的径向速度应与平台自运动一致，动态点的速度则表现出与自运动不相符的残差，从而可通过速度一致性检验自动剔除动态点，无需任何分割网络或类别标签 [35]。这一物理约束在实测数据上效果显著：纯距离 ICP 在 Baker-Barry 隧道序列的路径误差达 525 m，而 Doppler ICP 将其压缩至 1.23 m；在 Brisbane Lagoon Freeway 序列上，路径误差从 4337 m 降至 4.16 m，平均迭代次数也从 30.8 次降至 7.6 次。

上述方法的共同局限在于串联管线结构：分割或检测首先产生分类结果，后续步骤依赖这一分类。当分割本身在雨雪、夜晚、新目标类别下失效时，错误传播无法被后续配准级别检测到。现实系统” 分割/检测 $\rightarrow$ 过滤 $\rightarrow$ 配准” 的串联结构中，每一级的不确定性都会向下累积放大，而更理想的方向是将” 动态点分类” 作为潜变量，与对应关系和位姿在同一贝叶斯框架中联合推断，而不是通过前序步骤” 硬过滤” 后再配准 [104]。目前

联合框架的计算代价仍过高，难以满足实时要求。与此同时，主流配准基准对动态外点的覆盖极为有限，缺乏对动态外点比例、类别分布、运动速度等维度的系统刻画，使得方法改进难以被独立量化和可靠复现 [97]。此外，Doppler ICP 依赖 FMCW LiDAR，而现有大量部署系统使用的是 ToF LiDAR，如何在传感器条件受限或语义失效的情况下保持鲁棒性，目前仍无通用解决方案。从更长远的视角看，单帧配准完全忽视了相邻帧间目标运动的时序连续性，将动态目标的运动预测与逐帧配准结合以利用时序冗余消除瞬时遮挡的影响，是目前动态场景配准中较少被系统研究的方向。

7.3 几何退化与不确定性：何时” 拒绝配准” 更可靠

ICP 目标函数的约束来源于点对之间的距离或法向量残差。当点云中的约束方向分布不均匀时，Hessian 矩阵（或信息矩阵）的某些特征值趋近于零，使得对应方向的位姿分量变得不可观，优化问题在该方向上退化（degeneracy）[47][27]。退化的具体表现形式取决于场景几何：在长走廊或隧道中，法向量主要沿横截面分布，沿通道方向的平移约束极弱，ICP 可以在沿通道方向” 自由滑动” 而目标函数几乎不变；在大面积平面主导的场景中，面内平移方向退化；在对称结构或重复纹理场景中，多个不同位姿可能产生几乎相同的残差，最优解呈多模态分布。退化在工程中最危险的特征在于” 错误自信”：ICP 仍会收敛——残差下降到阈值以下，算法认为匹配成功——但实际位姿可能在不可观方向上偏移了数厘米甚至数米，这种失败模式在仅靠点云约束的系统中很难被内部检测到 [47]。

退化检测的早期工作以特征值分解为核心。Zhang 等人通过对信息矩阵做特征值分解，识别出约束不足的方向，并在该方向上用先验”best guess” 替代 ICP 求解结果，避免错误分量进入积分链路；在长轨迹退化场景下，末端位置误差相比无退化处理的基线方案从 6.37 m 降至约 0.71%（相对轨迹长度）[47]。Ji 等人提出点到分布退化检测因子，利用局部点分布的高斯几何模型定量刻画不同方向的约束强度，比特征值方法对噪声和稀疏点云更鲁棒 [51]。Hinduja 等人将退化感知因子嵌入因子图后端，实现与 SLAM 状态估计的紧耦合 [27]。

将退化检测升级为主动约束优化是更近的研究方向。X-ICP 不仅识别是否退化，还量化各方向的约束强度，构造方向选择矩阵，将位姿更新分解为” 良约束方向用 ICP 求解、退化方向用先验或外部传感器填充” 的混合策略；在 Seemühle 地下矿坑（VLP-16，高度退化场景）上，终点位置误差为 0.27 m，而对比方法分别为 6.37 m 和 24.17 m[48]。LP-ICP 将可定位性概念推广至更一般的退化模式，支持多类不可观方向的检测与自适应权重分配 [49]。DAMM-LOAM 将退化感知检测与 IMU 预积分融合，在退化触发时自动切换到 IMU 主导模式，保证姿态估计在纯 LiDAR 不可靠时的连续性 [54]。

确定性退化指标依赖人工设定阈值，在退化程度连续变化的场景中容易出现误判。Hatleskog 等人提出概率退化检测方法，在贝叶斯框架下建模约束强度的后验分布，避免硬阈值带来的不稳定性，在走廊和隧道等连续退化梯度场景中表现出更好的鲁棒性 [50]。GenZ-ICP 采用隐式路线，通过联合点到点与点到面度量的自适应加权，让优化过程在不显式检测退化的前提下自动降低退化方向的权重，在纯走廊场景（无 IMU）上验证了退化鲁棒性 [53]。更进一步，Stein ICP 将配准问题重新表述为后验分布估计，用粒子集（SVGD 更新）近似位姿后验，直接量化因几何退化或噪声引起的位姿不确定性；在对称几何物体和具有挑战性的 LiDAR 序列上，其运行时间约为 Bayesian ICP 的 1/5 至 1/8，同时提供多模态后验表达，能识别因对称结构导致的多解情形，为下游规划提供置信度输入 [46]。

尽管退化检测方法已经相当丰富，几个核心缺口仍然突出。退化检测输出的不确定性要真正发挥作用，必须映射为下游可执行策略（触发重定位、限制速度、切换传感器模态），而不仅是作为附加指标展示；现有方法与完整系统安全机制之间的接口设计——包括触发条件、置信度阈值校准和失败恢复路径——仍缺乏系统性研究 [55]。不同退化检测方法各自在私有场景上评测，缺乏覆盖单轴退化、双轴退化、对称歧义等不同退化类型和退化程度的统一基准，方法间横向比较极难进行。特征值分解的退化阈值通常需要针对具体场景手动调节，而点云的几何特性会随楼层、天气和传感器参数实时变化，基于历史统计或在线学习的自适应阈值校准在实际长时间自主导航中尤为重要，但目前实用化程度仍低。此外，当 LiDAR 在某方向退化时，理想策略是从 IMU、视觉或雷达等互补传感器获取该方向的约束，但多传感器融合的权重分配和故障检测本身引入了新的复杂性，现有退化鲁棒框架在极端退化与传感器同时失效的复合场景下鲁棒性保证仍然不足 [48][54]。

7.4 硬件感知算法：从” 加速某一步” 走向” 端到端共设计”

第 5 章 已经表明，ICP 的实时化瓶颈并非算力不足，而是点云数据的固有特性与通用硬件设计假设之间存在系统性不匹配。从三个维度可以刻画这种不匹配：在数据特性层面，点云的稀疏性（有效点通常不足三维空间栅格的 2%）、无序性与密度不均匀性导致 kNN 搜索存在大量冗余距离计算，且访存路径高度不规则，缓存命中率大幅下降 [6][20]；在数据流层面，kNN 树遍历涉及” 取节点—比较—更新候选集” 的短计算链路，外部存储访问延迟往往先于浮点算力成为性能天花板，即便增加并行计算单元，若片外带宽不足，系统性能上限也将被带宽墙而非算力墙所决定 [87]；在控制流层面，条件分支与动态任务分配导致 GPU 等通用硬件的线程束并行利用率常不足 40%，树遍历中有效点与空区域的频繁判断造成线程束发散，效率骤降 30%–50%[23]。三类瓶颈相互耦合，如果算法结构与硬件存储层级不匹配，即便处理器算力充足，系统也将长期被带宽与数据搬运所拖垮。

ASIC 专用数据通路的核心思路是将 kNN 搜索和几何算子固化到定制硬件，通过” 少搬数据、少走控制流” 换取确定性的能效收益。Tigris 将 kNN 搜索核心固化到专用数据通路，在 KITTI 点云配准任务中 kNN 子过程相对 RTX 2080 Ti 实现 77.2 倍加速，同时功耗降至约 1/7.4[6]。Tartan 将 kNN 搜索、矩阵运算与 IMU 积分统一到机器人专用处理器，通过任务级并行压缩端到端延迟，在保证功耗约束的同时支持多传感器融合 [81]。PointISA 则采用指令集扩展路线，将” 单点对多点” 的串行搜索转化为多点并行批处理，在不修改算法逻辑的前提下降低硬件实现复杂度，兼顾了软件可编程性与硬件加速效率 [92]。

FPGA 路线的核心是把随机树遍历改写成对硬件缓存友好的规则化流式访问。SA-RPS-CS 构建扫描线辅助范围投影结构，将点云按扫描线与距离重排到连续内存并保留几何拓扑信息，同时设计七级深度流水线并集成滑动窗口缓存；在 Xilinx ZCU104 FPGA 上实现 21.5 FPS，相比同类 FPGA 方案搜索速度提升 2.3–32.4 倍、能效提升 1.8–26.2 倍，精度损失小于 1%[91]。RPS-ICP 利用车载 LiDAR 扫描的角度规律优化投影后数据的结构化程度，在对应搜索阶段达到 18.6 FPS，搜索速度比既有 FPGA 实现快 13.7 倍、能效比 GPU 高 50.7 倍 [88]。HA-BFNN-ICP 在 3.4 W 功耗约束下实现相对 CPU 17.36 倍加速 [8]；ParallelNN 采用 HBM 与多通道片上缓存将 kNN 的外部带宽瓶颈替换为片上高带宽访问，在 KITTI 上相对 CPU 和 GPU 分别达到 107.7 倍和 12.1 倍加速、73.6 倍和 31.1 倍能效增益 [87]。

当数据搬运功耗在系统总功耗中的占比持续上升时，将距离计算与 Top-k 维护下推到存储阵列内部是消除带宽墙的根本路径。PICK 利用 SRAM 内部并行直接完成 kNN 搜索，在 KITTI、S3DIS、DALES 等数据集上相对既有设计实现 4.17 倍加速和 4.42 倍节能 [20]。C^2IM-NN 基于 9T1C SRAM 单元设计行方向内容寻址存储与列方向存内计算，并以 1D-CNN 预测体素点数将外部内存访问复杂度从 $O(N^2)$ 降至 $O(N)$；在 28 nm CMOS 工艺下功耗仅 137.41 mW，相比 FPGA 基线方案功耗降低 99.51%，能效提升 23.08 倍 [89]。

单模块加速比不等同于系统级收益，端到端 SoC 设计通过集成 ICP 全流程使评测结果能直接反映工程部署可行性。Kosuge 等人的 SoC-FPGA 设计在工业拣选机器人任务中实现 0.72 s、4.2 W 的完整位姿估计，比四核 CPU 快 11.7 倍 [7]。面向 Zynq-7000 的可重构定位加速器通过运行时配置适配不同场景，相对 Intel 和 Arm CPU 分别达到 59.1 倍和 9.2 倍加速，并通过动态功耗管理将平均能耗再降约 18%[90]。

尽管如此，几个系统性缺口仍然制约着硬件加速方案的实际可用性。现有加速器大多针对” 对应搜索 + 小规模求解” 的核心环节，而动态点剔除、法向估计、退化检测等预处理步骤往往对系统稳定性影响更大，一旦这些步骤仍运行在通用 CPU 上，专用加速器的端到端延迟优势就会被大幅压缩。动态目标引发的点云密度时变和运行时邻域图更新对控制流提出了更高要求，现有设计主要面向静态或结构化场景，GPU 因线程束发散在此类负载上效率骤降，专用加速器又难以在不牺牲灵活性的前提下支持动态拓扑变更，二者之间存在明显设计真空。点处理（kNN/FPS）、体素处理（稀疏卷积）、投影映射（BEV 注意力）与混合范式在计算结构、内存访问模式和控制流上差异显著，单一固定架构难以同时高效覆盖所有范式 [81]。最后，延迟、吞吐、内存峰值、功耗与精度损失需要同时报告，否则不同硬件方案之间难以公平比较，尤其在边缘部署语境下功耗预算往往比峰值算力更能区分方案优劣，但目前业界缺乏标准化的硬件感知评估协议 [23]。

7.5 评测与复现：从单指标到” 协议 + 资源 + 失败率”

评测结论的稳定性并不由指标本身决定，而由测量协议决定。同名指标在不同论文中常对应不同阈值、不同预处理与不同实现细节，导致横向对比结论不稳（第 6.2 节 ）。Pomerleau 等人对 ICP 变体的系统化对比揭示了这一问题的深度：即便使用相同算法族，模块组合（降采样策略、法向估计方法、对应过滤阈值）对最终结果

的影响常超过核心目标函数的差异；缺少协议一致性时，方法差异会被实现差异淹没 [23]。

在已有数据集和协议方面，TUM RGB-D 以 ATE/RTE 为主要指标，适合评测近距离刚体对齐，但动态目标和遮挡比例有限 [40]；3DMatch/3DLoMatch 以 5 cm/5° 阈值下的 Registration Recall 和对应内点率为核心，在低重叠配准研究中形成了相对稳定的比较语境，但阈值选择会直接左右方法排名，不同论文所用的 recall 阈值变体之间并不完全可比 [95]；KITTI 里程计基准以连续帧配准为测试对象，更贴近真实部署的分布，但场景以结构化道路为主，对走廊、室内等退化场景覆盖不足 [5]。nuScenes、Waymo Open Dataset、RobotCar 和 MulRan 等大规模多模态数据集更贴近工程部署的分布漂移与长时序变化，但其评测协议尚未在配准子任务上形成统一标准 [97][98][99][100]。

尚未解决的核心缺口在于缺乏硬件感知的统一评估维度。现有基准几乎全部以精度指标（RMSE、Recall）为核心，延迟、吞吐、内存峰值与功耗等资源维度既无标准化报告口径，也难以在方法间直接比较，尤其是在硬件加速与边缘部署日益普及的背景下（第 5 章 ）。另一个尚待解决的问题是失败率的系统刻画：当前评测多报告平均性能，而极端退化、动态干扰和极低重叠场景下的失败率——才是决定工程系统安全边界的关键维度——往往在汇总指标中被平均效应掩盖。

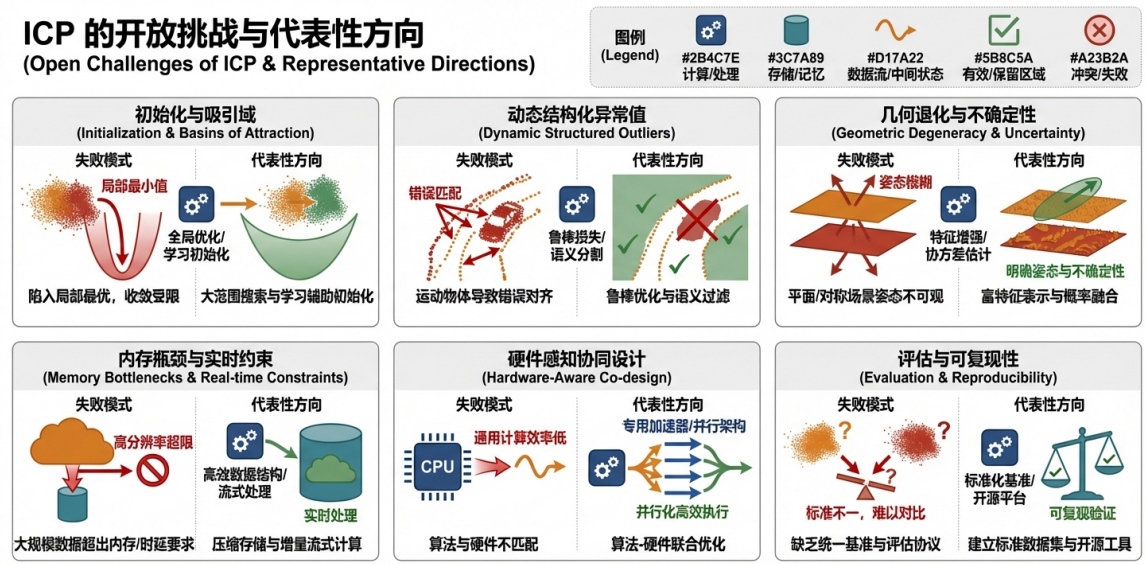


图 71: ICP 研究的主要开放挑战全景图。每个挑战都对应典型失败模式（收敛盆外、结构化外点、几何退化、内存瓶颈、协议不一致）及代表性技术方向（全局鲁棒估计、语义/物理量辅助、退化检测与不确定性、端到端共设计、标准化基准）。

表 36: 第 7 章开放挑战的“失败模式—对策—缺口”总结表。

挑战维度	典型失败模式	常见对策（代表工作）	仍未解决的问题
初始化与全局粗配准	初值落在收敛盆外、低重叠匹配不稳定	全局初始化 + ICP 精修 [28]、可认证鲁棒估计 [36]	可证鲁棒性与实时性/系统接口的统一
动态与语义	结构化外点导致系统性偏置	语义辅助配准 [104]、物理量辅助（Doppler）[35]	动态基准与联合推断框架不足
退化与不确定性	不可观方向导致漂移与错误自信	退化检测因子 [51]、不确定性估计 [46]	安全失败模式与系统策略耦合
硬件/资源约束	随机访存与带宽限制吞吐	专用处理器 [78]、PIM[20]	预处理与质量控制的端到端共设计
评测与复现	指标阈值/协议不一致	PCL 系统化基准 [23]、3DMatch 协议 [95]	硬件感知的统一报告口径

上表中的五类挑战彼此耦合，而不是彼此独立。几何退化会进一步压缩初始化的容错窗口，动态外点的联合推断依赖不确定性估计是否可信，硬件资源约束又反过来限制复杂联合优化能否落地。因此，更值得投入的方向不是在单一指标上继续压榨局部改进，而是构建能在系统约束下同时处理多个失败模式的框架。第 8 章 将据此回到全文主线，总结本文的主要判断与工程启示。

8. 结论 (Conclusion)

ICP 在刚体点云配准中长期扮演” 局部精修基线” 的角色: 它的优势在于问题分解清晰、实现成本低、便于与系统其他模块组合; 它的局限也同样明确, 主要来自局部收敛、外点与动态干扰, 以及计算资源约束 [1][2][13][3][4]。本文沿着第 3 章 、第 4 章 、第 5 章 和第 6 章 组织材料, 目标不是给出单一” 最优方法”, 而是建立一套面向工程选型的判断框架。以下按” 关键结论 → 硬件加速设计理念 → 面向不同读者的建议” 三层收束全文。

8.1 关键结论与实践要点

经典对比研究反复表明, 运行时的主导开销往往集中在对应建立与其数据结构实现上, 而不是小规模的闭式求解本身 [13][23]。这一结论在 FPGA、ASIC 和 PIM 等硬件加速实验中被反复验证: 从 Tigris 到 PICK, 针对 kNN 搜索阶段的专用优化带来的收益始终显著优于针对位姿求解阶段的优化 [6][20]。

在外点与部分重叠普遍存在的任务中, 截断、核函数、GNC 等鲁棒机制能明显改善偏置与失败率（第 3.2 节 ）, 但其收益取决于外点结构与评测协议 [14][23]。动态场景中, 语义辅助与物理量辅助（如 Doppler 速度残差）比纯粹调整核函数形态更能从根本上区分” 静态结构” 与” 结构化外点”, 这意味着鲁棒性改造在大多数情况下比更换局部目标函数更具工程回报 [35]。

两阶段配准范式是目前最成熟的通用工程模板: 全局初始化把问题拉回 ICP 的收敛盆, 局部 ICP 提供可控的精修与误差收敛（第 3.6 节 ）。这一范式既适用于手工特征（如 FPFH）也兼容学习型特征（如 FCGF、GeoTransformer）[9][28][56][57]。在硬件部署语境下, 全局粗配准通常被离线预处理或轻量级初始化替代, 以避免其搜索开销侵占实时配准的延迟预算。

降采样、并行化与近似最近邻等软件优化策略往往可以按” 先降规模、再降常数” 的顺序叠加（第 4 章 ）, 而在严格能耗与延迟预算下, 专用处理器、FPGA 流式化与 PIM 路线更可能带来确定性的端到端收益 [6][91][20]。单模块加速比不等于系统级收益: 预处理、质量控制与位姿融合等环节若仍运行在通用 CPU 上, 加速器的理论收益会在端到端测量中大打折扣 [7]。

点云算子的三维复杂性根源决定了加速策略的优先级。数据特性层面, 稀疏性、无序性与密度不均决定了哪些算子计算冗余最严重; 数据流层面, 短计算链路与多级聚合决定了哪些阶段最容易被带宽墙卡住; 控制流层面, 动态分支与不均衡任务分配决定了哪些算子在通用硬件上并行利用率最低 [6][87][89]。三类瓶颈相互耦合, 但一旦定位清楚, 加速方向就变得明确: 数据特性瓶颈靠数据规整化解决, 数据流瓶颈靠高带宽缓存或存内计算解决, 控制流瓶颈靠专用流水线和动态跳过机制解决。

评测结论的有效性同样受到评测协议本身的制约。3DMatch 等基准的经验表明, 同一指标在不同阈值下结论可能截然不同; 缺少阈值与实现细节时, 跨论文比较很难稳定复现 [95]。在硬件评测中, 延迟、吞吐、内存峰值与功耗四个资源维度需要与精度损失一并报告, 否则无法在算法精度与工程资源之间做出有意义的权衡判断。

8.2 硬件加速的通用性设计理念

第 5 章 与第 7.4 节 所梳理的加速工作, 尽管在器件形态与处理范式上各有侧重, 但汇聚出三条具有普遍性的设计理念:

所有加速范式的共同起点都是把不规则点云数据转化为对硬件友好的规则化结构, 同时深度利用稀疏性剔除无效计算。FPGA 路线通过扫描线辅助投影将随机访问转化为连续流访问 [91][88]; PIM 路线通过将候选集维护下推到存储侧从根本上消除数据搬运 [20][89]; ASIC 路线通过专用数据通路让” 有效计算” 比例最大化 [6]。数据规整的粒度和方式因范式而异, 但” 让硬件只处理有效数据” 的原则在所有路线中一以贯之。

高效加速的关键不在于把现有算法简单移植到新器件上, 而在于算法与硬件从设计之初就相互适配。算法层面, 用简化运算（如径向距离替代完整欧氏距离、预计算查找表替代三角函数）降低硬件实现复杂度; 硬件层面, 针对特定算子设计专用流水线和计算单元; 数据流层面, 通过细粒度分块、片上缓存优化与数据依赖解耦减少外部内存访问 [92][81]。SA-RPS-CS 的多模式对应搜索策略与七级深度流水线架构, 是算法-硬件协同设计的典型体现 [91]。

单一处理范式难以应对所有点云感知任务。实际部署场景下, kNN/FPS 等点处理算子与稀疏卷积、注意力机制等常需同时出现（例如语义辅助的 SLAM 系统既需要 kNN 做配准, 又需要分割网络做动态点过滤）。面向

多任务的可重构架构（如运行时可切换处理路径的 Zynq-7000 加速器）通过动态调度机制在延迟、能效与精度之间实现灵活权衡 [90]。随着多模态感知（LiDAR + 视觉 + 雷达）在自动驾驶中日益普及，支持跨模态数据对齐与融合注意力计算的专用单元将成为下一代硬件架构的关键组件 [81]。

8.3 面向不同读者的建议

对算法研究者而言，最重要的是先把失败模式说清楚再提方案——是收敛盆外、结构化外点，还是退化不可观（第 7 章 ）——并对每个失败模式给出可复现的评测协议（第 6 章 ）同时报告阈值，否则改进幅度在不同实验设置下可能完全逆转 [95][23]。在外点或低重叠任务中，截断、语义辅助与全局初始化带来的收益通常比换一个目标函数更持久、更稳定，因此鲁棒化和初始化往往比局部目标函数微调更值得优先投入 [14][36]。跨传感器与跨场景测试是验证泛化性的最低要求：学习型方法在训练分布外的退化往往决定了其实际使用价值，评测需显式覆盖域偏移设置 [97]。

对硬件研究者而言，优化目标应当被写成系统约束而非单一模块指标：延迟、吞吐、功耗、内存峰值与精度损失需要共同报告（第 7.5 节 ），否则难以与软件方案公平比较，也难以指导工程团队的方案选型 [23]。kNN 搜索最容易被带宽与随机访存支配，也是专用硬件最能发挥优势的环节，因此优先围绕” 数据访问 + 对应建立” 做协同设计是最成熟的路径：数据特性（稀疏性/密度不均）→ 数据规整策略（扫描线重排/存内计算）→ 流水线设计 [6][20][87]。此外，动态点剔除与退化检测等预处理和质量控制步骤对系统稳定性的影响常超过 kNN 搜索本身，若这些步骤仍留在 CPU 上，端到端延迟优势就会被吞噬 [91]。点云密度（低线束 vs 高线束 LiDAR）和场景复杂度（城市 vs 室内 vs 地下矿道）会显著改变算子的计算特性，验证固定架构在分布外场景的性能退化程度是衡量硬件鲁棒性不可省略的步骤。

对系统工程师而言，先写约束再选算法是基本原则：延迟预算、功耗预算与失败后果（能否安全拒绝）决定了鲁棒估计、不确定性量化和硬件投入是否必要（第 6.1 节 与第 7 章 ）[46]。自动驾驶场景的首要约束是端到端延迟与退化方向鲁棒性；工业检测场景的首要约束是重复精度与外点处理；医疗导航则把不确定性输出与失败告警放在第一位。软件堆栈的常规收益应当先被充分挖掘：降采样、ANN 数据结构与并行化通常能先带来稳定收益（第 4 章 ）[80]；若软件优化后延迟仍不满足预算，再按”FPGA 流式化 →ASIC 专用通路 →PIM” 的顺序逐级评估硬件投入（第 5 章 ）。退化检测与不确定性估计的意义不在于降低 RMSE，而在于为下游决策（重定位、降速、切换传感器）提供可信的置信度输出；孤立的配准精度数字不能替代” 失败率 + 失败模式” 的联合报告 [46][51]。

twidthhtwidth

图 72: 本综述的结构概览图：从算法变体到软件优化、硬件路线与评测语境，强调这些层次如何共同影响 ICP 的工程选型。横向箭头表示各层次之间的约束传递（算法设计约束实现代价，实现代价约束评测口径），纵向注解标出第 7 章识别的五类开放挑战与每层的关联位置。

参考文献

[1] P.J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[2] Yang Chen and Gerard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[3] François Pomerleau, Francis Colas, and Roland Siegwart. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, May 2015.

[4] G. K. L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian-Fang Sun, and Paul L. Rosin. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, July 2013.

[5] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, September 2013.

[6] Tiancheng Xu, Boyuan Tian, and Yuhao Zhu. Tigris: Architecture and algorithms for 3D perception in point clouds. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '52, pages 629–642, New York, NY, USA, October 2019. Association for Computing Machinery.

[7] Atsutake Kosuge, Keisuke Yamamoto, Yukinori Akamine, and Takashi Oshima. An SoC-FPGA-based iterative-closest-point accelerator enabling faster picking robots. *IEEE Transactions on Industrial Electronics*, pages 1–1, 2020.

[8] Yueze Liu, Yihong Tian, Xiaoxu Shen, Guanyu Qian, Hongwei Yang, and Xuemei Chen. HA-BFNN-ICP a streaming FPGA architecture for energy efficient real-time 3D LiDAR mapping. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1–11, 2025.

[9] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. FPFH: Fast Point Feature Histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, May 2009.

[10] Ningli Xu, Rongjun Qin, and Shuang Song. Point cloud registration for LiDAR and photogrammetric data: A critical synthesis and performance analysis on classic and deep learning algorithms. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 9:100067, 2023.

[11] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, September 1987.

[12] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, April 1987.

[13] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[14] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The Trimmed Iterative Closest Point algorithm. In *2002 International Conference on Pattern Recognition*, volume 3, pages 545–548 vol.3, August 2002.

[15] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse iterative closest point. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, SGP '13, pages 113–123, Goslar, DEU, July 2013. Eurographics Association.

[16] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, November 2016.

[17] A. L. Pavlov, G. V. Ovchinnikov, D. Yu Derbyshev, D. Tsetserukou, and I. V. Oseledets. AA-ICP: Iterative Closest Point with Anderson Acceleration. *arXiv:1709.05479 [cs]*, September 2017.

[18] Yue Wang and Justin Solomon. Deep Closest Point: Learning Representations for Point Cloud Registration. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3522–3531, Seoul, Korea (South), October 2019. IEEE.

[19] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust Point Matching Using Learned Features. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11821–11830, June 2020.

[20] Chen Nie, Chao Jiang, Liming Xiao, Weifeng Zhang, and Zhezhi He. PICK: An SRAM-based processing-in-memory accelerator for K-nearest-neighbor search in point clouds. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pages 1–7, June 2025.

[21] N. Brightman and L. Fan. A brief overview of the current state, challenging issues and future directions of point cloud registration. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-3-W1-2022:17–23, October 2022.

[22] Jiaqi Yang, Chu'ai Zhang, Zhengbao Wang, Xinyue Cao, Xuan Ouyang, Xiyu Zhang, Zhenxuan Zeng, Zhao Zeng, Borui Lu, Zhiyi Xia, Qian Zhang, Yulan Guo, and Yanning Zhang. 3d registration in 30 years: A survey, 2024.

[23] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, April 2013.

[24] P. Biber and W. Strasser. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 3, pages 2743–2748 vol.3, October 2003.

[25] Martin Magnusson. *The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, Örebro University, 2009.

[26] Aleksandr V. Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems (RSS)*, 2009.

[27] Akshay Hinduja, Bing-Jui Ho, and Michael Kaess. Degeneracy-Aware Factors with Applications to Underwater SLAM. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1293–1299, 2019.

[28] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast Global Registration. In *Computer Vision – ECCV 2016*, volume 9906, pages 766–782. Springer International Publishing, 2016.

[29] Yang Chen and Gerard Medioni. Object modeling by registration of multiple range images. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2724–2729, 1991. PDF mirror: https://www.cs.hunter.cuny.edu/ ioannis/chen$_m$*edioni*$_p$*oint*$_p$*lane*$_1$991.*pdf*.

[30] Szymon Rusinkiewicz. A symmetric objective function for ICP. *ACM Transactions on Graphics*, 38(4):85:1–85:7, July 2019.

[31] Leping He, Shuaiqing Wang, Qijun Hu, Qijie Cai, Muyao Li, Yu Bai, Kai Wu, and Bo Xiang. GFOICP: Geometric Feature Optimized Iterative Closest Point for 3-D Point Cloud Registration. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–17, 2023.

[32] Shaoyi Du, Tiancheng Shao, Canhui Tang, Wei Zeng, and Zhiqiang Tian. Robust point cloud registration based on semantic iterative closest point algorithm. *Fundamental Research*, February 2025.

[33] Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and Robust Iterative Closest Point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3450–3466, July 2022.

[34] Zongze Wu, Hongchen Chen, Shaoyi Du, Minyue Fu, Nan Zhou, and Nanning Zheng. Correntropy based scale ICP algorithm for robust point set registration. *Pattern Recognition*, 93:14–24, September 2019.

[35] Bruno Hexsel, Heethesh Vhavle, and Yi Chen. DICP: Doppler iterative closest point algorithm, May 2022.

[36] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Transactions on Robotics*, 37(2):314–333, April 2021.

[37] Jiaqi Yang, Zhiqiang Huang, Siwen Quan, Zhiguo Cao, and Yanning Zhang. RANSACs for 3D rigid registration: A comparative evaluation. *IEEE/CAA Journal of Automatica Sinica*, 9(10):1861–1878, October 2022.

[38] Lei Sun. SUCOFT: Robust Point Cloud Registration Based on Guaranteed Supercore Maximization and Flexible Thresholding. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–21, 2024.

[39] Donald G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM*, 12(4):547–560, 1965.

[40] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580. IEEE, October 2012.

[41] Seungpyo Hong, Heedong Ko, and Jinwook Kim. VICP: Velocity updating iterative closest point algorithm. In *2010 IEEE International Conference on Robotics and Automation*, pages 1893–1898, May 2010.

[42] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. FAST-LIO2: Fast direct LiDAR-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, August 2022.

[43] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and Fran
c cois Goulette. CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure, February 2022.

[44] Wenze Xia, Shaokun Han, Jingya Cao, Jie Cao, and Haoyong Yu. Scaling iterative closest point algorithm using dual number quaternions. 2017.

[45] Andrea Censi. An accurate closed-form estimate of ICP's covariance. In *Proceedings 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3167–3172. IEEE, 2007.

[46] Fahira Afzal Maken, Fabio Ramos, and Lionel Ott. Stein ICP for uncertainty estimation in point cloud matching. *IEEE Robotics and Automation Letters*, 7(2):1063–1070, April 2022.

[47] Ji Zhang, Michael Kaess, and Sanjiv Singh. On degeneracy of optimization-based state estimation problems. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 809–816, 2016.

[48] Turcan Tuna, Julian Nubert, Yoshua Nava, Shehryar Khattak, and Marco Hutter. X-ICP: Localizability-Aware LiDAR Registration for Robust Localization in Extreme Environments, 2022.

[49] Haosong Yue, Qingyuan Xu, Fei Chen, Jia Pan, and Weihai Chen. LP-ICP: General Localizability-Aware Point Cloud Registration for Robust Localization in Extreme Unstructured Environments. 2025.

[50] Johan Hatleskog and Kostas Alexis. Probabilistic Degeneracy Detection for Point-to-Plane Error Minimization. *IEEE Robotics and Automation Letters*, 9(12):11234–11241, 2024.

[51] Sehua Ji, Weinan Chen, Zerong Su, Yisheng Guan, Jiehao Li, Hong Zhang, and Haifei Zhu. A point-to-distribution degeneracy detection factor for LiDAR SLAM using local geometric models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12283–12289, 2024.

[52] Turcan Tuna, Julian Nubert, Patrick Pfreundschuh, Cesar Cadena, Shehryar Khattak, and Marco Hutter. Informed, Constrained, Aligned: A Field Analysis on Degeneracy-Aware Point Cloud Registration in the Wild. *IEEE Transactions on Field Robotics*, 2:485–515, 2025.

[53] Daehan Lee, Hyungtae Lim, and Soohee Han. GenZ-ICP: Generalizable and Degeneracy-Robust LiDAR Odometry Using an Adaptive Weighting. *IEEE Robotics and Automation Letters*, 10(1):152–159, 2025.

[54] Nishant Chandna and Akshat Kaushal. DAMM-LOAM: Degeneracy aware multi-metric LiDAR odometry and mapping. 2025.

[55] Wenda Wang, Qiuzhao Zhang, Yongfeng Hu, Michal Gallay, Wen Zheng, and Jianye Guo. Recent advances in SLAM for degraded environments: A review. *IEEE Sensors Journal*, 25(15):27898–27921, 2025.

[56] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8958–8967, 2019.

[57] Zheng Qin, Hao Yu, Changiian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric Transformer for Fast and Robust Point Cloud Registration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11133–11142, June 2022.

[58] Dror Aiger, Niloy J. Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics*, 27(3):1–10, 2008.

[59] Nicolas Mellado, Dror Aiger, and Niloy J. Mitra. Super 4PCS fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33(5):205–215, 2014.

[60] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & Efficient Point Cloud Registration using PointNet. *arXiv:1903.05711 [cs]*, April 2019.

[61] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation, 2016.

[62] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. DeepVCP: An End-to-End Deep Neural Network for Point Cloud Registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–21, 2019.

[63] Haoyu Yu, Fang Li, Mohamad Saleh, Benjamin Busam, and Slobodan Ilic. CoFiNet: Reliable coarse-to-fine correspondences for robust point cloud registration. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pages 23872–23884, 2021.

[64] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4265–4274. IEEE, June 2021.

[65] Rui She, Qiyu Kang, Sijie Wang, Wee Peng Tay, Kai Zhao, Yang Song, Tianyu Geng, Yi Xu, Diego Navarro Navarro, and Andreas Hartmannsgruber. PointDifformer: Robust Point Cloud Registration With Neural Diffusion and Transformer. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024.

[66] Efimia Panagiotaki, Daniele De Martini, Lars Kunze, Paul Newman, and Petar Veličković. NAR-*ICP: Neural Execution of Classical ICP-based Pointcloud Registration Algorithms, October 2025.

[67] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics, 2018.

[68] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. KISS-ICP: In defense of point-to-point ICP — simple, accurate, and robust registration if done the right way, 2022.

[69] Pavlos Mavridis, Anthousis Andreadis, and Georgios Papaioannou. Efficient Sparse ICP. *Computer Aided Geometric Design*, 35–36:16–26, 2015.

[70] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142, 2020.

[71] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated Non-Convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020.

[72] David M. Rosen, Luca Carlone, Afonso S. Bandeira, and John J. Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *The International Journal of Robotics Research*, 38(2–3):95–125, 2019.

[73] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, 2011.

[74] Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1–2):1–139, 2017.

[75] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J. Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[76] Yan Sun, Wanbiao Lin, Bohan Shi, Jiawei Shen, Liangbo Hu, and Lei Sun. EA2D-LSLAM: Environment analysis-based adaptive downsampling for point clouds in LiDAR SLAM. In *2025 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 1090–1095, June 2025.

[77] Yu. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020.

[78] Tiancheng Xu, Boyuan Tian, and Yuhao Zhu. Tigris: Architecture and algorithms for 3D perception in point clouds. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '52, pages 629–642. Association for Computing Machinery, 2019.

[79] Andreas Nüchter, Kai Lingemann, and Joachim Hertzberg. Cached k-d tree search for ICP algorithms. In *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, pages 419–426, 2007.

[80] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, November 2014.

[81] Mohammad Bakhshalipour and Phillip B. Gibbons. Tartan: Microarchitecting a robotic processor. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pages 548–565, Buenos Aires, Argentina, June 2024. IEEE.

[82] Yueze Liu, Yihong Tian, Xiaoxu Shen, Guanyu Qian, Hongwei Yang, and Xuemei Chen. HA-BFNN-ICP a streaming FPGA architecture for energy efficient real-time 3D LiDAR mapping. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1–11, 2025.

[83] Natasha Gelfand, Leslie Ikemoto, Szymon Rusinkiewicz, and Marc Levoy. Geometrically stable sampling for the ICP algorithm. In *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling*, pages 260–267, 2003.

[84] Wei Dong, Jaesik Park, Yi Yang, and Michael Kaess. GPU accelerated robust scene reconstruction. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7863–7870, November 2019.

[85] Qiong Chang, Weimin Wang, and Jun Miyazaki. Accelerating Nearest Neighbor Search in 3D Point Cloud Registration on GPUs. *ACM Trans. Archit. Code Optim.*, 22(1):43:1–43:24, March 2025.

[86] Aaron Barnes, Fangjia Shen, and Timothy G. Rogers. Extending GPU ray-tracing units for hierarchical search acceleration. In *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1027–1040, November 2024.

[87] Faquan Chen, Rendong Ying, Jianwei Xue, Fei Wen, and Peilin Liu. ParallelNN: A parallel octree-based nearest neighbor search accelerator for 3D point clouds. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 403–414, February 2023.

[88] Hao Sun, Jianzhong Xiao, Fupeng Chen, Weixiong Jiang, and Yajun Ha. A Real-time FPGA-Based Point Cloud Registration Framework with Efficient Correspondence Searching.

[89] Jeongmin Shin, Hoichang Jeong, Seungbin Kim, Keonhee Park, Sangho Lee, and Kyuho Jason Lee. C ^{2} IM-NN: A low-power 3D point clouds matching processor with 1D-CNN prediction and CAM-based in-memory k-NN searching. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1–12, 2025.

[90] Qiang Liu, Yuhui Hao, Weizhuang Liu, Bo Yu, Yiming Gan, Jie Tang, Shao-Shan Liu, and Yuhao Zhu. An energy efficient and runtime reconfigurable accelerator for robotic localization. *IEEE Transactions on Computers*, 72(7):1943–1957, July 2023.

[91] Qi Deng, Hao Sun, Yuhao Shu, Jianzhong Xiao, Weixiong Jiang, Hui Wang, and Yajun Ha. An energy-efficient and real-time FPGA-based point cloud registration framework with ultra-fast and configurable multi-mode correspondence search. *ACM Transactions on Reconfigurable Technology and Systems*, 18(4):1–30, 2025.

[92] Meng Han, Liang Wang, Limin Xiao, Hao Zhang, Bowen Jiang, Xilong Xie, Jianfeng Zhu, Shaojun Wei, and Leibo Liu. PointISA: ISA-extensions for efficient point cloud analytics via architecture and algorithm co-design. In *Proceedings of the 2025 58th IEEE/ACM International Symposium on Microarchitecture*, pages 1867–1881, Seoul Korea, October 2025. ACM.

[93] Ji Zhang and Sanjiv Singh. LOAM: LiDAR odometry and mapping in real-time. In *Robotics: Science and Systems (RSS)*, 2014.

[94] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443. IEEE, 2017.

[95] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 199–208. IEEE, July 2017.

[96] Jie Yin, Ang Li, Tao Li, Wenxian Yu, and Danping Zou. M2DGR: A Multi-Sensor and Multi-Scenario SLAM Dataset for Ground Robots. *IEEE Robotics and Automation Letters*, 7(2):2266–2273, April 2022.

[97] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628. IEEE, 2020.

[98] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng

Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2451. IEEE, 2020.

[99] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[100] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. Mulran: Multimodal range dataset for urban place recognition. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6246–6253. IEEE, 2020.

[101] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. The newer college dataset: Handheld lidar, inertial and vision with ground truth. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4353–4360. IEEE, 2020.

[102] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920. IEEE, June 2015.

[103] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565. IEEE, 2015.

[104] Steven A. Parkison, L. Gan, M. G. Jadidi, and R. Eustice. Semantic Iterative Closest Point through Expectation-Maximization. In *BMVC*, 2018.

[105] A. Zaganidis, Li Sun, T. Duckett, and Grzegorz Cielniak. Integrating Deep Semantic Segmentation Into 3-D Point Cloud Registration. *IEEE Robotics and Automation Letters*, 2018.