

# 北京理工大学

## 本科生毕业设计（论文）外文翻译

外文原文题目： Multi-instance Point Cloud Registration

by Efficient Correspondence Clustering

中文翻译题目： 多实例点云配准的高效对应聚类方法

## 基于深度学习的端到端多实例点云配准

**Deep Learning Based End-To-End  
Multi-instance Point Cloud Registration**

学 院： 自动化学院

专 业： 自动化

班 级： 06011902

学生姓名： 杨润一

学 号： 1120191211

指导教师： 由育阳

## 多实例点云配准的高效对应聚类方法

### 摘要

我们解决了在目标点云中估计多个源点云实例姿态的问题。现有的解决方案需要采样大量假设以检测可能的实例并拒绝异常值，当实例数量和异常值增加时，其鲁棒性和效率显著降低。我们建议直接根据距离不变性矩阵将嘈杂的对应关系集合分组到不同的簇中。通过聚类自动识别实例和异常值。我们的方法既鲁棒又快速。我们在合成数据集和真实世界数据集上评估了我们的方法。结果表明，我们的方法在存在 70% 异常值的情况下，可以正确配准多达 20 个实例，F1 得分为 90.46%，性能明显优于现有方法且速度至少快 10 倍。

**关键词：**点云配准；多实例；聚类；对应聚类；深度学习；毕业设计（外文翻译）

## 目 录

摘要 .....	I
第1章 引言 .....	1
第2章 相关工作 .....	3
第3章 问题陈述 .....	5
第4章 方法 .....	6
4.1 不变性矩阵和兼容性向量 .....	6
4.2 快速对应关系聚类 .....	7
4.3 递归簇细化 .....	8
4.4 合并重复变换 .....	8
4.5 从簇中提取变换 .....	9
4.6 处理大量对应关系 .....	9
第5章 实验验证 .....	10
5.1 合成数据集 .....	10
5.2 基准数据集 .....	13
5.3 真实场景测试 .....	14
第6章 局限性 .....	16
第7章 结论 .....	17
参考文献 .....	18

## 第 1 章 引言

三维点云配准<sup>[1-3]</sup>主要关注在源点云和目标点云之间估计一个单一的变换。然而，我们有时可能希望在点云之间估计多个变换。例如，我们有一个物体的 3D 扫描，可能希望在目标点云中找到如图 1-1 所示的同一物体在桌子上的姿态。这个问题，我们称之为多实例点云配准，文献中较少研究。这个问题，我们称之为多实例点云配准，文献中较少研究。将现有的点云配准方法扩展到解决这个问题并非易事。

主要的挑战是在嘈杂的对应关系集合中识别属于不同实例的对应点的不同簇。一种解决方案是采用 3D 物体检测器或将实例分割应用于目标点云。然后，通过常规的点云配准方法估计每个实例的姿态。然而，这种方法需要为特定对象或类别训练检测器或分割网络<sup>[4-5]</sup>，不适用于未知对象或任意 3D 扫描。

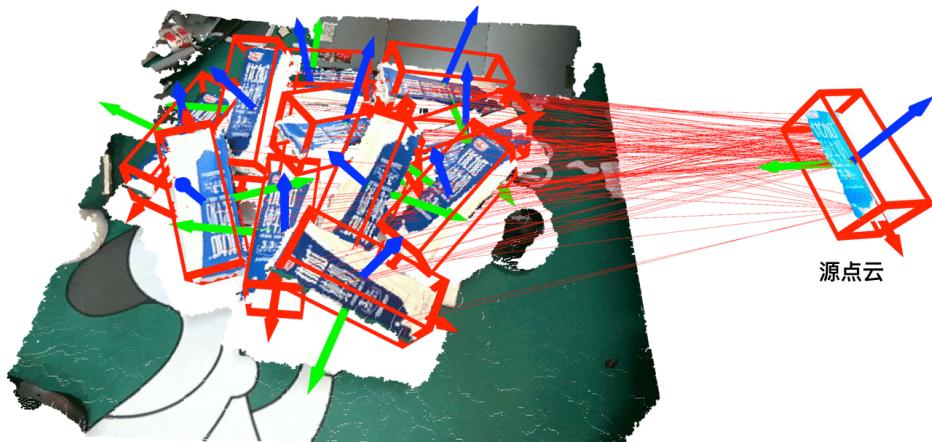


图 1-1 多实例点云配准：给定一个物体的源点云，多实例配准需要估计目标点云中每个物体的姿态。

另一种解决方案是通过多模型拟合<sup>[6-8]</sup>或<sup>[9-11]</sup>。现有的多模型拟合方法依赖于采样有效的假设，当模型数量或异常值比例变高时，涉及大量的采样步骤，使得这些算法的效率和鲁棒性急剧下降。

在本文中，我们提出了一种鲁棒且高效的多实例三维配准问题解决方案。关键思想是根据距离不变性矩阵直接将对应点分组到不同的簇中。具体而言，在使用诸如 D3Feat<sup>[12]</sup>、PREDATOR<sup>[13]</sup>或 SIFT<sup>[14]</sup>等描述符进行特征匹配得到点对应关系之后，通过检查每对对应关系之间的距离一致性来构造矩阵。我们发现这个矩阵的行或列

向量具有强大的表示能力，可以用于识别特定实例的对应关系集合。因此，我们应用了一种简单且高效的聚类算法将这些对应关系划分为团。通过将相似簇合并并重新分配簇 id 给每个对应关系，递归地进行几步细化聚类。最后，通过一个简单的排序策略自动识别每个实例的异常值和内点。

由于不需要耗时的假设采样，我们的方法具有很高的效率。我们在合成数据集和真实世界数据集上进行了广泛的实验。结果表明，我们的方法在准确性和鲁棒性方面表现明显优于现有方法，速度至少快 10 倍。总之，我们的贡献包括：

- 我们提出了一种针对多实例点云配准问题的高效且鲁棒的解决方案，其在准确性、鲁棒性和速度方面都具有优越的性能。
- 我们提出使用三个指标（平均命中召回率、平均命中精确度和平均命中 F1）来全面评估多实例点云配准的性能。
- 我们的解决方案可以用于零样本检测 3D 物体，正如我们的真实世界测试所示。

## 第 2 章 相关工作

**点云配准**可以分为三个阶段：点匹配、异常值剔除和姿态估计。大多数工作集中在前两个阶段，因为获得正确的点对应关系是成功配准的关键。点匹配通常依赖于特征，可以是手工设计的特征<sup>[15-16]</sup>，也可以是基于学习的特征<sup>[12,17-20]</sup>。尽管近期的结果显示，在某些基准测试中，后者优于手工设计的特征，但这些特征仍然无法实现完美匹配，因此仍然需要强大的异常值剔除机制。**RANSAC**<sup>[21]</sup>及其变体（<sup>[22-24]</sup>）遵循假设-验证过程来剔除异常值。当存在许多异常值时，这类方法需要大量的采样步骤，耗时严重，而且仍可能无法获得正确的模型。**GORE**<sup>[25]</sup>和 **PMC**<sup>[26]</sup>试图通过几何一致性检查来减少异常值。其他方法如 **FGR**<sup>[27]</sup>和 **TEASER**<sup>[1]</sup>采用鲁棒估计器直接从噪声对应关系中解决变换问题。通过仔细处理每个子问题，**TEASER**<sup>[1]</sup>在鲁棒性和效率方面取得了令人印象深刻的性能。还有一些基于学习的异常值剔除方法。**DGR**<sup>[28]</sup>和 **3DRegNet**<sup>[29]</sup>将异常值剔除视为二分类问题，并预测每个对应关系的内点概率。**PointDSC**<sup>[30]</sup>进一步将空间一致性嵌入到特征学习中，以便更好地训练内点分类器。最近，一系列工作（如 **PointNetLK**<sup>[31]</sup>，**FMR**<sup>[32]</sup>，**DCP**<sup>[2]</sup>，**PRNet**<sup>[33]</sup>，**RPMNet**<sup>[34]</sup>）尝试将端到端学习应用于解决配准问题。它们在性能方面也表现出色，特别是在低重叠情况下<sup>[13]</sup>。

现有的点云配准方法主要关注一对一配准问题，即估计两个点云之间的单一变换。然而，将源点云与目标点云中的多个实例对齐的多实例配准问题则研究较少。这个任务与多路配准<sup>[35]</sup>不同，后者的目标是通过成对配准<sup>[27][28]</sup>从多个片段生成全局一致的重建。多实例配准需要不仅从噪声对应关系中剔除异常值，还要识别出各个实例的内点集，使其比经典配准问题更具挑战性。

**3D 目标检测和实例分割**与多实例 3D 配准密切相关。给定一个单独的点云，3D 目标检测<sup>[4]</sup>的目的是获取每个感兴趣对象的边界框，而 3D 实例分割<sup>[5,36]</sup>为每个点生成实例标签。尽管它们生成的结果<sup>[37-38]</sup>与多实例配准类似，但它们需要将特定对象或类别的先验训练到网络中。相比之下，多实例配准通过直接将源点云与目标点云中的多个实例对齐来处理两个点云，而不使用关于输入 3D 扫描内容的任何先验知识。

**多模型拟合**多实例配准可以通过多模型拟合方法来解决，其目的是从多模型生

成的数据点中估计模型参数。现有的多模型拟合方法可以分为基于聚类的方法和基于 RANSAC 的方法。基于聚类的方法（例如<sup>[6-8]</sup>）通过采样点初始化一个庞大的假设集，然后为每个点计算关于这些假设的偏好向量。根据它们的偏好向量对这些数据点进行聚类。最后，从不同的簇计算模型参数。基于 RANSAC 的方法（如<sup>[9-11,39-41]</sup>）顺序运行修订后的 RANSAC 以获得多个模型参数。它们在每次迭代中改变每个点的采样权重以获得不同的模型参数。CONSAC<sup>[10]</sup>是一种基于学习的方法，通过学习为每个点分配采样权重。无论是基于聚类的方法还是基于 RANSAC 的方法，都依赖于采样有效假设。当模型数量或异常值比例增加时，需要采样很多假设，这使得这些算法效率非常低。

**3D 空间一致性**是 3D 配准中异常值剔除的重要属性，它在每对点之间通过刚性变换来定义。谱匹配<sup>[42]</sup>通过在每对对应关系之间使用长度一致性来构建一个图，并从图中提取最大团以剔除异常值。现有的方法如 TEASER<sup>[1]</sup>， GORE<sup>[25]</sup>和 PMC<sup>[26]</sup>也将空间一致性纳入到它们的算法中。最近，ROBIN<sup>[43]</sup>将空间一致性的概念推广到更高阶。PointDSC<sup>[30]</sup>将空间一致性集成到端到端学习管道中，以便更好地回归内点概率。

受这些工作的启发，我们也采用了空间一致性作为解决方案。与现有方法（如谱聚类<sup>[42]</sup>或近似解<sup>[43]</sup>）在空间一致性图中应用的方法不同，这些方法速度较慢且难以处理多个实例，我们采用了一种高效的算法来在对应关系中找到多个实例。具体来说，我们将距离不变矩阵的行向量或列向量作为对应关系的“特征向量”，并运行自底向上的聚类来从不同实例中获得内点对应关系。我们的方法避免了假设采样，这是现有多模型拟合方法的关键弱点。它还不依赖于任何特定特征来获得点对应关系，因此，如果采用更好的特征（无论是 3D 特征还是图像特征），性能可以进一步提高。

### 第3章 问题陈述

在多实例点云配准问题中，源点云  $\mathbf{X}$  提供了一个 3D 模型的实例，目标点云  $\mathbf{Y}$  包含了这个模型的  $K$  个实例，其中这些实例是一组点的集合，这些点可能只采样了 3D 模型的一部分。如果我们将第  $k^{th}$  个实例写为  $\mathbf{Y}_k$ ，那么目标点云  $\mathbf{Y}$  可以分解为  $\mathbf{Y} = \mathbf{Y}_0 \cup \mathbf{Y}_1 \cup \dots \cup \mathbf{Y}_k \dots \cup \mathbf{Y}_K$ 。这里我们使用  $\mathbf{Y}_0$  表示点云中不属于任何实例的部分。多实例 3D 配准的目标是找到刚性变换  $(\mathbf{R}_k, \mathbf{t}_k)$ ，将源实例  $\mathbf{X}$  对准到每个目标实例  $\mathbf{Y}_k$ 。如果我们设法获得源实例与每个目标实例  $\mathbf{X} \leftrightarrow \mathbf{Y}_k$  之间的对应关系，那么通过最小化对齐误差之和 (3-1)<sup>[44]</sup>，可以从对应关系集合  $\mathbf{X} \leftrightarrow \mathbf{Y}_k$  中求解目标点云中第  $k^{th}$  个实例的位姿  $(\mathbf{R}_k, \mathbf{t}_k)$ ：

$$\min_{\mathbf{R}_k, \mathbf{t}_k} \sum_i \|\mathbf{y}_{ki} - (\mathbf{R}_k \mathbf{x}_i + \mathbf{t}_k)\|^2. \quad (3-1)$$

考虑到我们已经获得了源点云和目标点云之间的一组对应关系  $\mathcal{C}$ 。多实例配准任务的关键是将这些对应关系分类为与不同实例相关的独立集合，即：

$$\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1 \dots \cup \mathcal{C}_K. \quad (3-2)$$

这里， $\mathcal{C}_0$  用来表示异常值集合。如我们所见，多实例配准不仅需要剔除异常值对应关系，还需要解决来自不同实例的对应关系的歧义。这个任务并不容易，因为所有实例看起来都一样，而且通常存在大量的异常值对应关系。

## 第 4 章 方法

我们提出的方法的概述如图4-1所示。我们的方法以点对应关系作为输入。接着通过检查对应关系之间的距离一致性来构建一个不变性一致性矩阵。然后，通过将列或行向量视为这些对应关系的“特征”，将这些对应关系快速聚类成不同的组。通过凝聚聚类方法高效地进行聚类，然后通过交替合并相似的变换和重新分配簇标签进行多次迭代来进一步优化。在对应关系数量较大的情况下，我们可以选择性地应用降采样和上采样过程。详细内容将在接下来的章节中介绍。

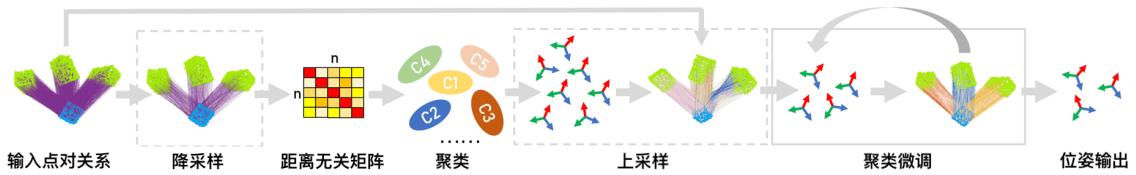


图 4-1 我们提出的多实例点云配准方法的流程。从输入对应关系构建距离不变性矩阵，用于将对应关系聚类为不同的簇（聚类），并进行优化（簇优化）。最后，从每个对应关系簇中估计与每个实例相关的刚体变换（变换）。为了处理大量的对应关系，采用两个附加过程（下采样和上采样）。

### 4.1 不变性矩阵和兼容性向量

距离不变性特性在 3D 配准领域已经被研究多年<sup>[1][43][42]</sup>，该特性描述了在刚性变换之后两点之间的距离保持不变。具体来说，如果  $c_i : \mathbf{x}_i \leftrightarrow \mathbf{y}_i$  和  $c_j : \mathbf{x}_j \leftrightarrow \mathbf{y}_j$  是两个真实的对应关系，那么它们应该满足

$$G_{ij} = |d_{ij} - d'_{ij}| < \delta \quad (4-1)$$

其中  $d_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$ ,  $d'_{ij} = |\mathbf{y}_i - \mathbf{y}_j|$ ,  $\delta$  是一个用于考虑噪声的阈值。因此， $d_{ij}$  和  $d'_{ij}$  之间的差异可以用作度量是否存在异常值，或者两个对应关系是否来自不同的刚性变换的指标。我们参考<sup>[45]</sup>，使用相对差异作为度量，而不是在 (4-1) 中定义的绝对差异，

$$G_{ij} = s_{ij}^2, s_{ij} = \min\left(\frac{d_{ij}}{d'_{ij}}, \frac{d'_{ij}}{d_{ij}}\right) \in (0, 1). \quad (4-2)$$

通过计算所有对应关系对之间的分数，可以获得一个距离不变性矩阵  $G$ （我们令  $G_{ii} = 1$ ）。距离不变性矩阵是对称的，其中每一列或行是一个向量，描述了给定对应关系与其他对应关系之间的兼容性<sup>[46]</sup>。

我们将列向量  $G_i = (G_{i1}, \dots, G_{ij}, \dots)^T$  称为对应关系  $c_i$  的兼容性向量。我们观察到，如果两个对应关系属于同一个实例，它们的兼容性向量具有相似的模式。考虑两个对应关系  $c_i, c_j \in \mathcal{Cs}$ 。对于任何对应关系  $c_k \in \mathcal{Cs}$ ，由于距离不变性，我们有  $G_{ik} \rightarrow 1, G_{jk} \rightarrow 1$ 。对于其他对应关系  $c_k \in \mathcal{C}/\mathcal{Cs}$ ，我们可能有  $G_{ik} \rightarrow 0, G_{jk} \rightarrow 0$ 。换句话说， $G_i, G_j$  具有相似的 0 – 1 模式。相比之下，如果两个对应关系属于不同的实例，它们的兼容性向量则非常不同。为了更好地理解这个观察，我们在图4-2中展示了一个简单的示例。

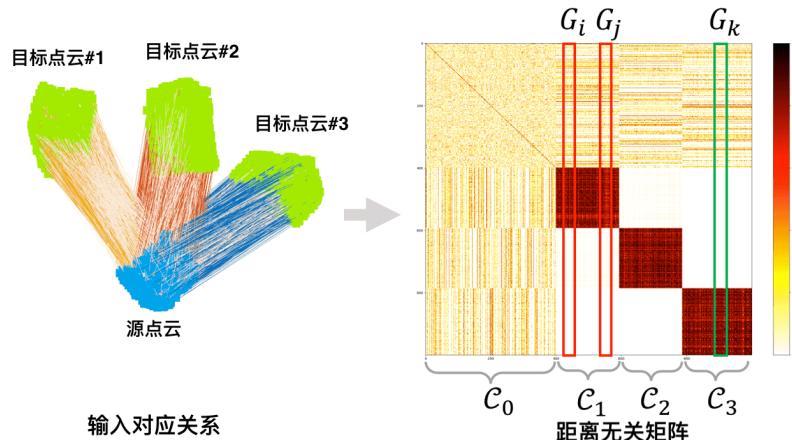


图 4-2 距离不变性矩阵中的列向量（兼容性向量）包含了与实例相关的丰富信息。这里  $G_i, G_j$  分别表示第  $i^{th}$  和第  $j^{th}$  对应关系的兼容性向量，它们都属于实例  $\mathcal{C}_1$ 。我们观察到  $G_i$  与  $G_j$  相似。相比之下， $G_i$  与  $G_k$  差异显著，因为第  $k^{th}$  对应关系位于不同的实例  $\mathcal{C}_3$  中。这里  $\mathcal{C}_0$  代表异常值集合。请参阅第4.1节以获取详细信息。

对应关系的兼容性向量可以被视为该对应关系的特征表示或“特征”。属于同一刚性变换的对应关系具有相似的特征。因此，基于这些兼容性向量，我们可以将对应关系聚类为与来自不同实例的内点相关的不同组。

## 4.2 快速对应关系聚类

我们以自底向上的方式聚类对应关系，这比现有方法采用的谱聚类<sup>[26][43]</sup>要快得多。一开始，每个对应关系被视为一个独立的组。然后，我们反复合并距离最小的两个组，直到两个组之间的最小距离大于给定值 ( $min_{dist} thresh$ )。定义组之间距离的

方式产生了不同风格的算法。我们遵循<sup>[6]</sup>来定义距离。设  $\mathbf{p}_i, \mathbf{p}_j$  为两个组  $i$  和  $j$  的表示向量，组距离定义为

$$d(\mathbf{p}_i, \mathbf{p}_j) = 1 - \frac{\langle \mathbf{p}_i, \mathbf{p}_j \rangle}{\| \mathbf{p}_i \|^2 + \| \mathbf{p}_j \|^2 - \langle \mathbf{p}_i, \mathbf{p}_j \rangle}. \quad (4-3)$$

如果两个组合并，新组的表示向量更新为  $\mathbf{p}_i \leftarrow \min(\mathbf{p}_i, \mathbf{p}_j)$ , 其中  $\min(\cdot)$  表示取两个向量每个维度的最小值。在聚类开始时，一个组（只包含一个对应关系）的表示向量设置为该对应关系的兼容性向量。

### 4.3 递归簇细化

在凝聚聚类之后，我们通过重复以下步骤来进一步优化结果，直到没有变化发生。

步骤 1. 从对应关系数量大于阈值  $\alpha$  的簇中估计刚性变换。

步骤 2. 合并相似的变换。这一步将在下一节中解释。

步骤 3. 为每个对应关系重新分配簇标签。将每个对应关系分配给对齐误差最小的变换。如果在所有变换中最小的对齐误差大于  $inlier_{thresh}$ ，则将对应关系标记为异常值。

在迭代过程中，对应关系变得越来越集中，因此我们可以在步骤 1 中调整  $\alpha$  以增加异常值拒绝的强度。我们在每次迭代中更新  $\alpha$  的策略如下：

$$\alpha \leftarrow \min(\alpha_0 \times \theta^{n-1}, [N/100]), \quad (4-4)$$

其中  $n$  表示第  $n^{th}$  次迭代， $N$  是对应关系的数量， $[.]$  是四舍五入运算。在我们的实验中，我们设置  $\alpha_0 = 3$  和  $\theta = 3$ 。细化过程通常在我们的实验中在三次迭代内收敛，因此效率也非常高。

### 4.4 合并重复变换

有时来自不同簇的相似变换会生成，这意味着它们可能属于同一个实例。在这种情况下，我们需要合并它们。给定两个估计的变换  $(\mathbf{R}1, \mathbf{t}1)$  和  $(\mathbf{R}2, \mathbf{t}2)$ ，我们计算每个对应关系的对齐误差，即  $eki = |\mathbf{y}_i - (\mathbf{R}k\mathbf{x}_i + \mathbf{t}_k)|^2, (k = 1, 2)$ 。接下来，我们设置  $pki = 1$  如果  $eki < inlier_{thresh}$ ，否则  $pki = 0$ 。因此，我们为两个变换获得两个

二进制集  $P_1, P_2$ 。合并两个变换的条件是

$$IOU = |P_1 \cap P_2| / |P_1 \cup P_2| \geq 80 \quad (4-5)$$

如果满足这个条件，我们将放弃一个异常值较多的变换 ( $p_{ki} = 0$ )。然后我们根据在所有变换中对齐误差最小的一个重新为每个对应关系分配簇标签。

## 4.5 从簇中提取变换

在聚类之后，我们需要从这些对应关系簇中提取刚性变换。由于我们不知道目标点云中真实实例的数量，因此我们需要自动选择那些内点簇。我们首先选择内点簇的元素数量大于阈值（在我们的实验中为 10）并从这些簇中估计变换。接下来，我们根据其内点数量按降序对变换进行排序。一个变换拥有的内点越多，它与真实实例相关联的可能性就越高。最后，我们检查内点数量在变换之间的降低比例，以及第一个变换（具有最多内点）之间的比例，通过

$$\gamma_k = \#I_k / \#I_0, \quad k = 1, 2, \dots \quad (4-6)$$

其中  $\#I_k$  表示第  $k^{th}$  变换的内点数量。如果  $\gamma_k <= \gamma_{thresh}$ ，我们忽略所有在  $k$  之后的变换。 $\gamma_{thresh}$  可以更改以在召回和精确度之间进行权衡。

## 4.6 处理大量对应关系

当输入对应关系的数量很大时，计算距离不变矩阵和聚类对应关系可能会变得昂贵。我们通过添加下采样和上采样过程来解决这个问题。在构建距离不变矩阵之前进行下采样过程，通过随机抽样固定数量的对应关系（在我们的实现中为 1024）进行进一步处理。在选定对应关系聚类之后进行上采样过程，将所有对应关系分配给现有簇。分配是通过选择对齐误差最小的变换来完成的，如第 4.3 节中的步骤 3 所述。

## 第 5 章 实验验证

我们在合成数据集和真实世界数据集上进行实验，将我们的方法与三种最先进的多模型拟合方法进行比较：T-linkage(2014)<sup>[6]</sup>，Progressive-X(2019)<sup>[41]</sup> 和 CONSAC(2020)<sup>[10]</sup>。其他多模型拟合方法：RPA<sup>[7]</sup> 和 RansaCov<sup>[8]</sup> 非常慢（需要几个月的时间）来运行我们的实验，因此我们没有包括它们。我们还展示了最先进的对一配准方法 TEASER(2020)<sup>[1]</sup> 的结果作为比较。我们仔细调整所有方法，在合理的时间和内存消耗范围内，在评估数据集上实现最佳性能。为了公平比较，所有方法都使用相同的点对应关系集作为输入。

我们使用 Pytorch<sup>Pytorch</sup> 实现我们的算法。T-linkage 和 Progressive-X 是纯 CPU 算法，而 CONSAC 是基于 GPU 的学习方法。我们在与 T-linkage 和 Progressive-X 相同的 CPU (Intel Core i7-8700K) 上运行我们的算法，并在与 CONSAC 相同的 GPU (GTX 1080Ti) 上运行。我们的方法有三个参数，其中在我们的实验中设置为  $\text{min\_dist\_thresh} = 0.2$ ,  $\text{inlier\_thresh} = 0.3$  and  $\gamma\_thresh = 0.5$ 。所有点云都在  $0.05m$  体素大小中进行下采样。正如补充材料中的消融研究所示，我们的方法对参数变化不敏感。

由于一对配准中使用的指标不能用于多实例设置，我们从检索任务中采用三个评估指标：MHR (Mean Hit Recall), MHP (Mean Hit Precision), MHF1 (Mean Hit F1)。它们的定义详见补充材料。

### 5.1 合成数据集

我们从 PointNet++<sup>[47]</sup> 预先采样的 Modelnet40 数据集<sup>[48]</sup> 生成一个合成数据集。我们将每个点云下采样到 256 个点，并随机生成  $K$  个变换（在我们的测试中最多为 20），形成一个目标点云。目标点云还与其他对象和随机点混合，以更好地模拟真实世界情况。

**仿真对应关系** 在这个测试中，我们直接通过混合地面真实值和离群值来生成输入对应关系。我们测试了不同的离群比例， $10\% \sim 50\%$ ,  $50\% \sim 70\%$  和  $70\% \sim 90\%$ 。请注意，对于每个测试样本，离群点是在给定范围内随机抽样的。结果显示在表 5-1 中。随着离群比例的增加，几乎所有方法的性能都有所下降，但我们的方法下降缓

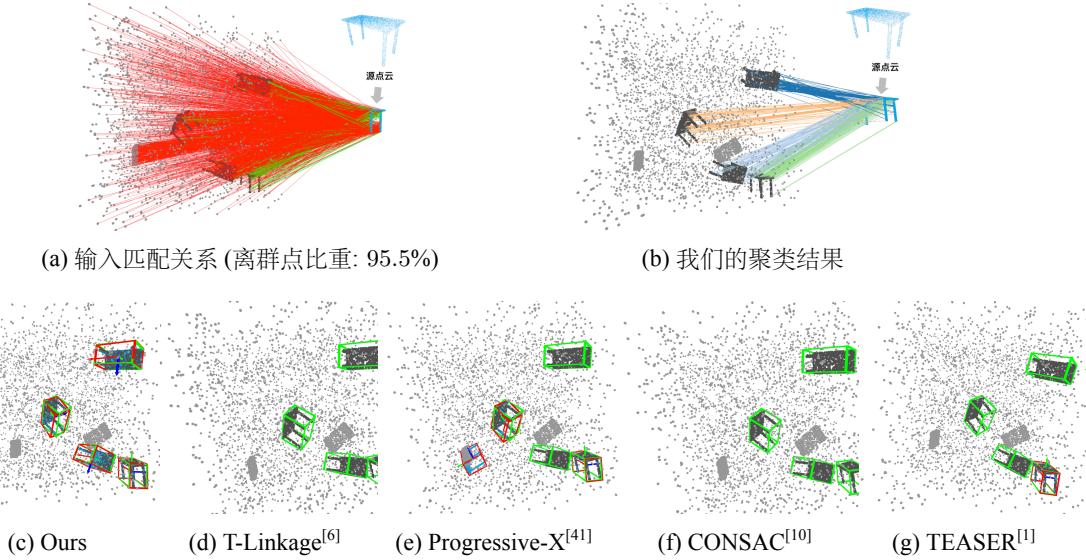


图 5-1 在数据集上的结果。 (a) 通过匹配 PREDATOR<sup>[13]</sup> 特征得到的输入对应关系。内点和离群点分别用绿色和红色显示。 (b) 我们的方法聚类结果用不同的颜色显示 (只显示内点)。在 (c-g) 中, 我们用红色框显示估计的位姿, 用绿色框显示真实位姿。我们的方法 (c) 配准了所有实例。T-linkage (d) 和 CONSAC (f) 无法配准任何实例。Progressive-X (e) 配准了 2 个实例, 但产生了错误的配准。TEASER (g) 配准了一个实例。

慢, 且仍然明显优于其他方法。我们的算法在 CPU 或 GPU 上的速度比现有方法快 10 倍。我们还在图 5-2(a) 中绘制了我们的方法在不同离群比例下的 MHF1(Mean Hit F1) 曲线, 其中包括 20 个实例。尽管当离群比例非常大时性能迅速下降, 但我们的方法在 70% 的离群比例下仍然能达到 90.46% 的 MHF1。图 5-2(b) 显示了在固定离群比例 50% 的情况下, 不同实例数量的 MHF1 曲线。即使存在 30 个实例, 我们的方法的 MHF1 也约为 92.73%。

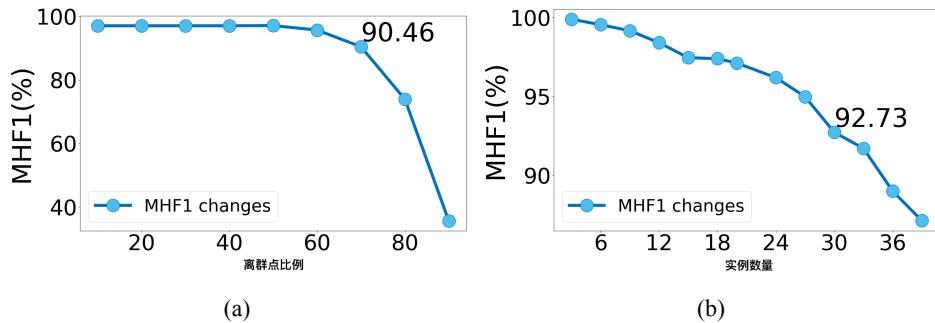


图 5-2 (a) 平均 Hit F1 与离群比例关系。 (b) 平均 Hit F1 与实例数量关系 (固定离群比例为 50%)。

方法	MHR(%) ↑	MHP(%) ↑	MHF1(%) ↑	时间 (s) ↓
离群点比重: 10% ~ 50%				
T-Linkage	3.05	14.80	4.65	57.27
Progressive-X	27.91	80.28	41.04	87.25
CONSAC	0.47	0.47	0.47	9.23
<b>Ours</b>	<b>96.08</b>	<b>99.73</b>	<b>97.03</b>	<b>0.62/0.30</b>
离群点比重: 50% ~ 70%				
T-Linkage	1.33	7.00	2.05	56.90
Progressive-X	20.60	75.10	31.70	85.54
CONSAC	0.49	0.49	0.49	9.55
<b>Ours</b>	<b>93.99</b>	<b>99.49</b>	<b>95.51</b>	<b>0.55/0.28</b>
Outlier ratio : 70% ~ 90%				
T-Linkage	0.81	4.42	1.25	56.89
Progressive-X	12.88	62.60	20.73	84.5
CONSAC	0.51	0.51	0.51	7.70
<b>Ours</b>	<b>60.39</b>	<b>94.42</b>	<b>69.36</b>	<b>0.50/0.24</b>
离群点比重: 90% ~ 99%				
T-Linkage	0.28	1.30	0.42	56.69
Progressive-X	7.13	39.19	11.67	84.43
CONSAC	0.51	0.51	0.51	9.57
<b>Ours</b>	<b>14.70</b>	<b>65.20</b>	<b>22.75</b>	<b>0.47/0.21</b>

表 5-1 在不同离群比例的合成对应关系上的结果。↑表示越大越好，而↓表示相反。我们方法在 CPU/GPU 上的运行时间也展示出来。

**基于特征匹配的对应关系** 在此测试中，我们使用 PREDATOR<sup>[13]</sup> 和 D3Feat<sup>[12]</sup> 特征匹配来获得点对应关系。这两个特征模型都是在合成数据上训练的。结果显示在表 5-2 中。请注意，这两个特征都产生了超过 90% 的高离群点比例。在这样一个具有挑战性的情况下，我们的方法在鲁棒性和效率方面仍然表现良好，且明显优于现有方法。使用 D3Feat 的结果要比使用 PREDATOR 的结果差得多。原因不仅是因为更多的离群点，而且我们在检查结果时发现缺少内点。我们在图 5-1 中展示了一些结果。

Metric	MHR(%) ↑	MHP(%) ↑	MHF1(%) ↑	Time(s) ↓
PREDATOR ( estimated outlier ratio : 94.32%)				
T-Linkage	0.19	0.54	0.27	43.46
Progressive-X	15.90	31.01	18.98	86.39
CONSAC	0.1	0.07	0.08	7.65
<b>Ours</b>	<b>53.39</b>	<b>61.44</b>	<b>51.80</b>	<b>1.28/0.48</b>
D3Feat ( estimated outlier ratio : 99.30%)				
T-Linkage	0.07	0.29	0.1	56.37
Progressive-X	4.29	15.28	5.94	87.22
CONSAC	0.13	0.04	0.05	9.53
<b>Ours</b>	<b>16.98</b>	<b>27.05</b>	<b>17.91</b>	<b>0.68/0.30</b>

表 5-2 使用特征匹配在合成数据上生成对应点的结果。部分结果可在图 5-1 中进行可视化。

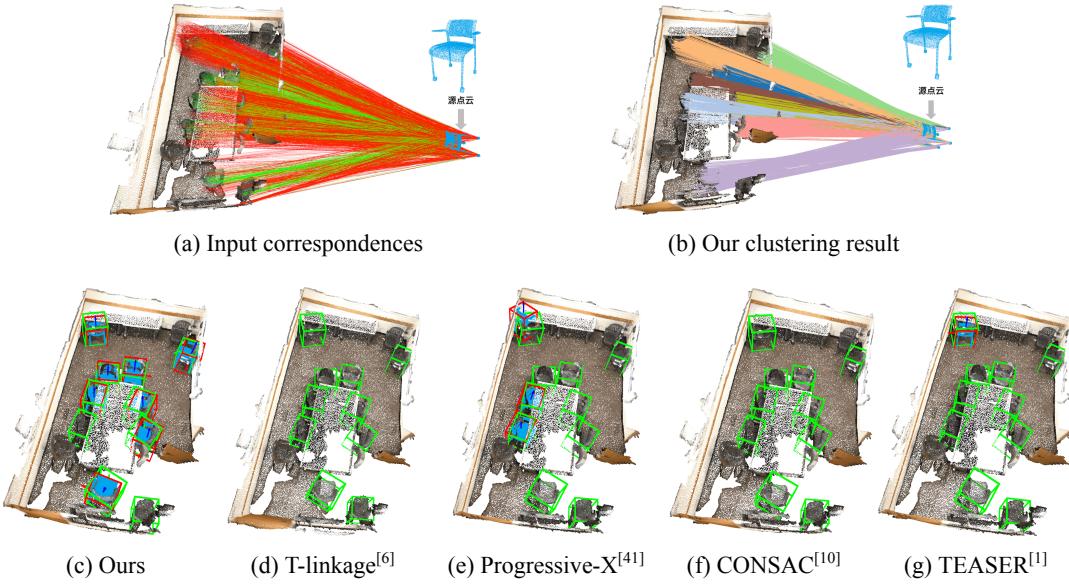


图 5-3 Scan2CAD 结果。 (a) 通过匹配 PREDATOR<sup>[13]</sup> 特征得到的输入对应关系。内点和离群点分别用绿色和红色显示。 (b) 我们的聚类结果用不同颜色显示 (仅显示内点)。在 (c-g) 中, 我们将估计的姿态用红色框显示, 将真实姿态用绿色框显示。我们的方法 (c) 正确地对齐了 8 个实例。 T-Linkage (d) 和 CONSAC (f) 未能配准任何实例。 Progressive-X (e) 配准了 3 个实例。 TEASER (g) 注册了一个实例。

## 5.2 基准数据集

Scan2CAD<sup>[49]</sup> 是一个基准数据集, 它将 ShapeNet<sup>[50]</sup> CAD 模型与 ScanNet<sup>[51]</sup> 点云中的对象实例对齐。部分扫描有多个已对齐的 CAD 模型和标注的姿态。我们选择那些包含多个 CAD 模型的扫描作为目标点云, 并从 CAD 模型中采样源点云进行测试。我们为配准测试生成了 173 个样本, 其中大多数样本包含 2 ~ 5 个实例。请注意, 在每个点云中, Scan2CAD 中仅标注了部分实例。这意味着我们无法使用部分标注的姿态正确评估性能, 如准确率和召回率。为解决这个问题, 我们仅在目标点云中标注的物体的真实边界框内匹配点以生成对应关系。同样, 我们使用 PREDATOR<sup>[13]</sup> 和 D3Feat<sup>[12]</sup> 进行点匹配, 两者都是使用 Scan2CAD 数据集中的 1028 个训练样本和 187 个验证样本进行微调的。结果如表 5-3 所示。我们的方法在使用 PREDATOR 时明显优于现有方法。请注意, 当使用 D3Feat 时, 所有方法的性能都很差。在仔细检查结果后, 我们发现原因不仅是高异常值比率 (约 97.25%), 而且在使用 D3Feat 时, 即使特征匹配仅限于目标点云中的真实边界框内, 内点也不足。一些结果如图 5-3 和图 5-4 所示。我们还通过将边界框扩大  $1.5\times$ ,  $2.0\times$  和  $4.0\times$  来评估我们方法的性能。

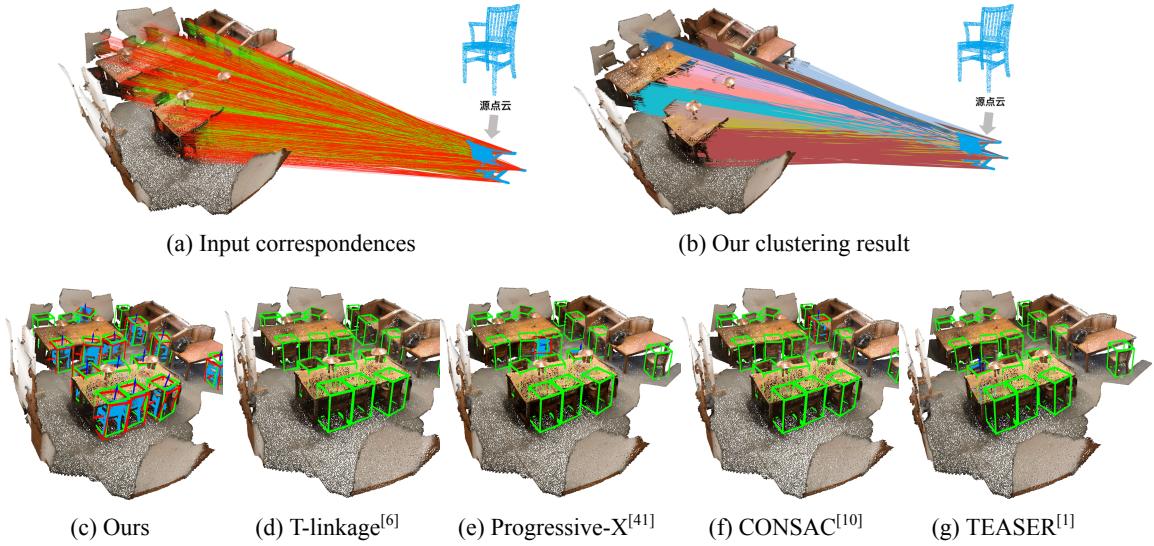


图 5-4 Scan2CAD 结果。我们的方法 (c) 在 16 个椅子中配准了 13 个实例。Progressive-X (e) 配准了 2 个实例，但其中一个的姿态误差较大。CONSAC (f) 和 TEASER (g) 都配准了一个实例。T-Linkage (d) 未能配准任何实例。

Metric	MHR(%) $\uparrow$	MHP(%) $\uparrow$	MHF1(%) $\uparrow$	Time(s) $\downarrow$
PREDATOR (estimated outlier ratio : 76.44%)				
T-Linkage	2.46	3.79	2.71	1655.0
Progressive-X	11.58	6.86	7.87	26.32
CONSAC	2.66	0.35	0.62	21.35
<b>Ours</b>	<b>31.63</b>	<b>29.23</b>	<b>27.04</b>	<b>1.46/0.51</b>
D3Feat (estimated outlier ratio : 97.25%)				
T-Linkage	0.04	0.22	0.06	2178.43
Progressive-X	<b>0.67</b>	<b>0.30</b>	<b>0.4</b>	28.48
CONSAC	0	0	0	21.88
<b>Ours</b>	0.29	0.04	0.07	<b>2.13/0.89</b>

表 5-3 在 Scan2CAD 数据集上点的结果。

当框大小调整为  $4 \times$  时，目标点云几乎是原始扫描。基于 PREDATOR 特征的结果如表 5-4 所示。当包含更多背景点时，特征匹配变得更具挑战性，产生高度噪声的对应关系，这使得我们方法的 MHF1 (平均命中 F1) 显著降低。

### 5.3 真实场景测试

我们使用一个 RGB-D 相机 (Intel D455) 捕捉一系列点云，这些点云表示了桌子上的一堆物体，并将我们的算法应用于将某个特定物体的 3D 扫描与目标 RGB-D 扫描中的多个实例进行对齐。由于颜色信息可用，我们使用 SIFT 特征生成 3D 点对应关系。然后我们应用我们的算法提取每个物体的姿态。图 5-5 显示了一些结果。尽管桌子上摆满了不同的物体，但我们的方法几乎能实时地（每帧约 0.2 秒）正确地将源

Box Size	MHR(%) ↑	MHP (%) ↑	MHF1(%) ↑	Time(s) ↓
1.5 (94.50%)	40.58	12.61	16.64	0.63
2 (95.97%)	37.43	8.85	12.35	1.37
4 (98.73%)	7.25	6.28	4.92	3.86

表 5-4 在 Scan2CAD 数据集上使用不同大小的边界框进行特征匹配的结果。估计的异常值比率在框大小后的括号内列出。最后一列列出了 GPU 时间。

3D 扫描与多达十几个实例进行对齐。更多结果可以在补充材料中找到。

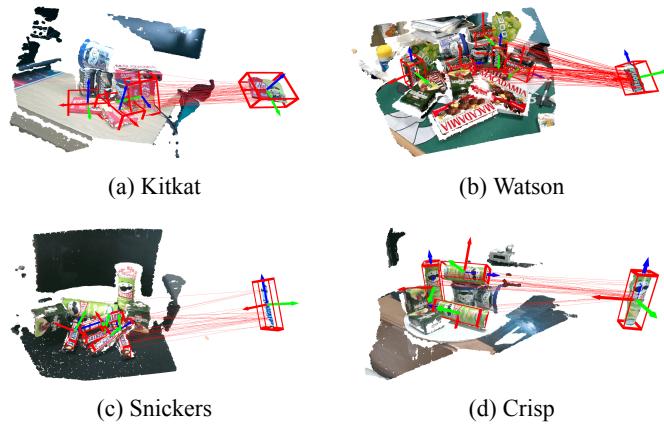


图 5-5 实际测试基于 RGB-D 扫描。源点云是从单个物体的深度扫描中提取的。目标点云是从相机视点捕获的深度扫描构建的。

## 第 6 章 局限性

我们方法的性能依赖于点对应关系的质量。不幸的是，我们发现使用诸如 D3Feat 和 PREDATOR 这样的最先进的 3D 特征产生的对应关系不令人满意，尽管它们在一些基准测试中表现良好。为了提高对应关系的质量，我们必须为实验中的每个数据集训练这些特征。即使这样做，异常值比率仍然很高，有时某些实例的内点会丢失（特别是在使用 D3Feat 时），这会显著降低召回率，正如实验所示。因此，3D 特征是一个需要显著改进的瓶颈。另一个局限性是距离不变性对于噪声点云来说是一个弱规则，有时不足以排除接近内点的异常值，这也可能降低我们方法的性能。一种可能的解决方案是在端到端学习管线中寻找更好的不变性“特征”来表示每个对应关系，如<sup>[30]</sup>。

## 第 7 章 结论

在本文中，我们讨论了多实例 3D 配准的新颖任务。我们发现距离不变性矩阵的列向量编码了与对应关系相关的实例的丰富信息。基于这个观察，我们通过聚类算法将对应关系高效地聚类到不同的组中，并通过多次迭代来优化结果。在合成、基准和实际数据集上的结果表明，我们的方法在鲁棒性、准确性和效率方面明显优于现有方法。尽管我们的解决方案仍然远非完美，正如所讨论的那样，但我们希望我们的工作能激发对这一主题的未来研究。

## 参考文献

- [1] Yang H, Shi J, Carlone L. Teaser: Fast and certifiable point cloud registration[J]. TRO, 2020, 37(2): 314-333.
- [2] Wang Y, Solomon J M. Deep closest point: Learning representations for point cloud registration[C]//ICCV. 2019: 3523-3532.
- [3] Besl P, McKay N D. A method for registration of 3-D shapes[J]. IEEE TPAMI, 1992, 14(2): 239-256.
- [4] Qi C R, Litany O, He K, et al. Deep hough voting for 3d object detection in point clouds[C]//ICCV. 2019: 9277-9286.
- [5] Han L, Zheng T, Xu L, et al. Occuseg: Occupancy-aware 3d instance segmentation[C]//CVPR. 2020: 2940-2949.
- [6] Magri L, Fusiello A. T-linkage: A continuous relaxation of J-linkage for multi-model fitting[J]. CVPR, 2014: 3954-3961.
- [7] Magri L, Fusiello A. Robust Multiple Model Fitting with Preference Analysis and Low-rank Approximation[J]. BMVC, 2015: 20.1-20.12.
- [8] Magri L, Fusiello A. Multiple model fitting as a set coverage problem[C]//CVPR. 2016: 3318-3326.
- [9] Magri L, Fusiello A. Fitting multiple heterogeneous models by multi-class cascaded t-linkage[C]//CVPR. 2019: 7460-7468.
- [10] Kluger F, Brachmann E, Ackermann H, et al. CONSAC: Robust Multi-Model Fitting by Conditional Sample Consensus[C]//CVPR. 2020.
- [11] Barath D, Rozumny D, Eichhardt I, et al. Progressive-X+: Clustering in the Consensus Space[J]. ArXiv preprint arXiv:2103.13875, 2021.
- [12] Bai X, Luo Z, Zhou L, et al. D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features[J]. CVPR, 2020.
- [13] Huang S, Gojcic Z, Usvyatsov M, et al. PREDATOR: Registration of 3D Point Clouds with Low Overlap[C]//CVPR. 2021: 4267-4276.
- [14] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. IJCV, 2004, 60(2): 91-110.
- [15] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]//ICRA. 2009: 3212-3217.
- [16] Drost B, Ulrich M, Navab N, et al. Model globally, match locally: Efficient and robust 3D object recognition[C]//CVPR. 2010: 998-1005.
- [17] Zeng A, Song S, Niessner M, et al. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions[J]. CVPR, 2017, 2017-Janua: 199-208.
- [18] Deng H, Birdal T, Ilic S. PPFNet: Global Context Aware Local Features for Robust 3D Point Matching[J]. CVPR, 2018: 195-205.
- [19] Gojcic Z, Zhou C, Wegner J D, et al. The perfect match: 3D point cloud matching with smoothed densities[J]. CVPR, 2019, 2019-June: 5540-5549.
- [20] Choy C, Park J, Koltun V. Fully convolutional geometric features[C]//ICCV. 2019: 8958-8966.
- [21] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381-395.
- [22] Barath D, Matas J. Graph-cut RANSAC[C]//CVPR. 2018.

- [23] Brachmann E, Rother C. Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses[C]//ICCV. 2019.
- [24] Chum O, Matas J, Kittler J. Locally Optimized RANSAC[C]//DAGM-Symposium. 2003.
- [25] Bustos A P, Chin T J. Guaranteed outlier removal for point cloud registration with correspondences[J]. IEEE TPAMI, 2017, 40(12): 2868-2882.
- [26] Parra Á, Chin T J, Neumann F, et al. A Practical Maximum Clique Algorithm for Matching with Pairwise Constraints[J]. ArXiv preprint arXiv:1902.01534, 2019.
- [27] Zhou Q Y, Park J, Koltun V. Fast Global Registration[J]. ECCV, 2016, 9906(August): 694-711.
- [28] Choy C, Dong W, Koltun V. Deep Global Registration[J]. CVPR, 2020: 2511-2520.
- [29] Pais G D, Ramalingam S, Govindu V M, et al. 3dregnet: A deep neural network for 3d point registration[C]//CVPR. 2020: 7193-7203.
- [30] Bai X, Luo Z, Zhou L, et al. PointDSC: Robust Point Cloud Registration using Deep Spatial Consistency[J]. CVPR, 2021.
- [31] Aoki Y, Goforth H, Srivatsan R A, et al. Pointnetlk: Robust & efficient point cloud registration using pointnet[C]//CVPR. 2019: 7163-7172.
- [32] Huang X, Mei G, Zhang J. Feature-Metric Registration: A Fast Semi-Supervised Approach for Robust Point Cloud Registration Without Correspondences[C]//CVPR. 2020.
- [33] Wang Y, Solomon J M. PRNet: Self-Supervised Learning for Partial-to-Partial Registration[C]//NeurIPS. 2019.
- [34] Yew Z J, Lee G H. RPM-Net: Robust Point Matching using Learned Features[J]. CVPR, 2020.
- [35] Choi S, Zhou Q Y, Koltun V. Robust reconstruction of indoor scenes[C]//CVPR. 2015: 5556-5565.
- [36] Wang W, Yu R, Huang Q, et al. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation[C]//CVPR. 2018: 2569-2578.
- [37] Avetisyan A, Dai A, Niessner M. End-to-end cad model retrieval and 9dof alignment in 3d scans[C]//ICCV. 2019: 2551-2560.
- [38] Avetisyan A, Khanova T, Choy C, et al. Scenecad: Predicting object alignments and layouts in rgb-d scans[C]//ECCV. 2020: 596-612.
- [39] Isack H, Boykov Y. Energy-based geometric multi-model fitting[J]. IJCV, 2012, 97(2): 123-147.
- [40] Barath D, Matas J. Multi-class Model Fitting by Energy Minimization and Mode-Seeking[C]//Ferrari V, Hebert M, Sminchisescu C, et al. ECCV. Cham: Springer International Publishing, 2018: 229-245.
- [41] Barath D, Matas J. Progressive-X: Efficient, anytime, multi-model fitting algorithm[C]//ICCV. 2019: 3780-3788.
- [42] Leordeanu M, Hebert M. A spectral technique for correspondence problems using pairwise constraints[J], 2005.
- [43] Shi J, Yang H, Carlone L. ROBIN: a graph-theoretic approach to reject outliers in robust estimation using invariants[C]//ICRA. 2021: 13820-13827.
- [44] Arun K, Huang T, Blostein S D. Least-Squares Fitting of Two 3-D Point Sets[J]. IEEE TPAMI, 1987, PAMI-9(5): 698-700.
- [45] Buch A G, Yang Y, Krüger N, et al. In Search of Inliers: 3D Correspondence by Local and Global Voting[C]//CVPR. 2014: 2075-2082.
- [46] Yang J, Xian K, Wang P, et al. A Performance Evaluation of Correspondence Grouping Methods for 3D Rigid Data Matching[J]. IEEE TPAMI, 2019, 14(8): 1-1.

- [47] Qi C R, Yi L, Su H, et al. PointNet++: Deep hierarchical feature learning on point sets in a metric space[J]. NeurIPS, 2017, 2017-Decem: 5100-5109.
- [48] Wu Z, Song S, Khosla A, et al. 3d shapenets: A deep representation for volumetric shapes[C]// CVPR. 2015: 1912-1920.
- [49] Avetisyan A, Dahnert M, Dai A, et al. Scan2cad: Learning cad model alignment in rgbd scans[C] //CVPR. 2019: 2614-2623.
- [50] Savva M, Yu F, Su H, et al. Shrec16 track: largescale 3d shape retrieval from shapenet core55[C] //3DOR: vol. 10. 2016.
- [51] Dai A, Chang A X, Savva M, et al. Scannet: Richly-annotated 3d reconstructions of indoor scenes[C] //CVPR. 2017: 5828-5839.