doi：10. 12068/j. issn. 1005－3026. 2026. 20249039

采用多尺度特征融合的点云配准算法

汪 洋[1,2]，张 蔚[1,2,3]，谢 磊[1,2]，陈德山[1,2]

（1. 武汉理工大学 智能交通系统研究中心，湖北 武汉 430063；
2. 水路交通控制全国重点实验室，湖北 武汉 430063；3. 武汉理工大学 交通与物流工程学院，湖北 武汉 430063）

摘 要：为了解决深度学习方法在处理部分重叠点云配准时出现的泛化性和精度下降问题，提出一种多尺度特征融合的点云配准算法．首先，对不同尺度的点云进行特征描述以获得局部结构信息，通过加权和对不同尺度下的特征进行融合，以增强特征的表示能力；然后，利用多注意力机制增强点云之间的特征关联性，发现更准确的对应关系；最后，使用位置估计模块计算源点云的虚拟对应点，并通过奇异值分解(SVD)求解源点云与虚拟对应点的最佳变换．在ModelNet40数据集的完整点云场景下，旋转与平移的均方误差分别为6.3×10^{-2}与1.0×10^{-5}；在部分重叠点云场景下旋转平移的均方误差分别为2.29×10^{-4}与3×10^{-4}，均低于深度最近点(DCP)算法．对于未知种类点云和噪声点云，本算法的误差仍低于迭代最近点（ICP）、快速全局配准（FGR）以及深度最近点（DCP）等代表性对比方法，进一步验证了算法的有效性．

关 键 词：点云；点云配准；多尺度特征融合；多注意力机制；虚拟对应点

中图分类号：TP 391 **文献标志码：**A **文章编号：**1005-3026(2026)02-0040-10

Point Cloud Registration Algorithm with Multi-scale Feature Fusion

WANG Yang[1,2], *ZHANG Wei*[1,2,3], *XIE Lei*[1,2], *CHEN De-shan*[1,2]

（1. Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, China; 2. State Key Laboratory of Maritime Technology and Safety, Wuhan University of Technology, Wuhan 430063, China; 3. School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China. Corresponding author: CHEN De-shan, E-mail: dschen@whut.edu.cn）

Abstract: To address the problem of poor generalization and decreased accuracy when deep learning methods are utilized to handle partially overlapping point cloud registration, a point cloud registration algorithm with multi-scale feature fusion was proposed. First, feature descriptions of point clouds at different scales were performed to obtain local structural information, and features at different scales were fused through weighted summation to enhance the feature representation capability. Then, a multi-attention mechanism was used to strengthen the feature correlation between point clouds and discover more accurate correspondences. Finally, a position estimation module was employed to calculate the virtual corresponding points of the source point cloud, and singular value decomposition (SVD) was used to solve for the optimal transformation between the source point cloud and the virtual corresponding points. In the complete point cloud scenario of the ModelNet40 dataset, the mean square errors for rotation and translation are 6.3×10^{-2} and 1.0×10^{-5}, respectively. In the partially overlapping point cloud scenario, the mean square errors for rotation and translation are 2.29×10^{-4} and 3×10^{-4}, respectively, both of which are lower than those of the deep closest point (DCP) algorithm. Moreover, for unknown-category and noisy point clouds, the proposed method also yields lower errors than representative baseline methods such as iterative closest point (ICP), fast global registration (FGR), and DCP, further validating its effectiveness.

收稿日期：2024-08-25
基金项目：国家重点研发计划项目(2023YFB4301800，2023YFC3010803)；国家自然科学基金资助项目(52272424，52372320)．
作者简介：汪 洋(1976—)，男，湖北武汉人，武汉理工大学研究员．
通信作者：陈德山，E-mail：dschen@whut.edu.cn.

Key words: point cloud; point cloud registration; multi-scale feature fusion; multi-attention mechanism; virtual correspondence point

随着三维扫描设备的广泛应用,点云已经成为计算机视觉中的一类重要研究对象.点云是由大量离散的点在三维空间中的位置坐标组成的数据集,通过三维坐标描述空间中每个点的特定位置,这些点的集合能够表示物体或场景的几何形状和表面特征.点云配准是指将多个点云数据通过空间变换,使其对齐到一个共同坐标系中,并使位于不同点云、但表征相同空间特征的点元素尽可能重叠.点云配准在自动驾驶、医学成像、水下探测等众多领域发挥着重要作用,其主要用途在于将多个来源的点云数据精确衔接,从而拼接生成完整的三维模型.这一技术在自动驾驶中用于场景感知和对象探测,在医学成像中用于多模态图像融合或术前术中图像对比,在水下探测中用于水下环境建模和目标的识别与分类.点云配准的算法主要分为传统算法和基于机器学习的算法.

文献[1]的迭代最近点(ICP)是一种被广泛使用的传统算法[1].它迭代地寻找最近的对应点,并且计算对应状态下的刚性变换,通过最小化对应点的坐标距离误差来确定最优变换.但是ICP非常依赖点云的初始位置,如果2个点云的初始位置相差较大,算法易于陷入局部最优.此外,当噪声点和离群点较多时,往往导致ICP的配准精度不理想.为此,之后的研究工作尝试对ICP进行多种改进.全局优化迭代最近点(GO-ICP)将分支定界过程加入了ICP,它在整个空间中搜索最优解,有效地避免算法陷入局部最优[2].三角迭代最近点(TRICP)将最小二乘加入了ICP,以处理部分点云场景,但这个方法对初始位置很敏感[3].快速全局配准(FGR)使用快速点特征直方图(FPFH)对点云进行特征描述[4],在特征空间寻找最近点,并放弃了迭代采样,从而加速配准过程.然而,FGR的不足在于对噪声点和离群点的处理能力有限[5].正态分布变换(NDT)是另一种重要的传统点云配准算法[6],它将点云进行体素化,计算每个体素内点云的概率密度函数,进而寻求最优变换矩阵使得点云概率密度函数的整体似然最大.但NDT对体素的划分要求很高,因此它的收敛性不甚理想.四点集一致性(4PCS)是一种传统点云配准算法[7],其核心是利用几何不变量,在2个点云中找到一个四点集,使这个四点集在2块点云中保持相同的几何关系.4PCS计算每一对四点集的变换矩阵,选择使源点云到目标点云重叠点数最多的变换作为最佳变换.这些传统方法受初始位置和噪声点的影响,缺乏较强的鲁棒性.

随着机器学习算法的兴起,研究人员使用深度学习方法来解决配准问题.PointNet是用深度学习来解决点云分类和分割问题的标志性算法[8].PointNet网络通过多层感知机(MLP)从点云坐标信息中提取出高维特征,然后通过最大池化得到整个点云的全局特征.由于最大池化并不会改变点云特征的置换不变性和无序性,因此可有效提取点云特征.受PointNet的启发,PointNetLK旨在用深度学习算法解决点云配准问题[9].它使用PointNet的编码器对每个点云进行全局特征描述,但舍弃了变换网络(T-Net)部分,而使用改进的Lucas-Kanade算法来获得刚性变换矩阵.点云配准网络(PCRNet)利用2个PointNet编码器提取点云特征,通过2个点云的全局特征预测出旋转四元数[10],对噪声点有很好的鲁棒性.这2种算法都使用了PointNet的编码器,仅提取全局特征而忽略了点云的局部特征,因此精度上会有损失.深度最近点(DCP)算法通过边缘卷积来提取特征,而后使用变换器(Transformer)建立点云之间的对应关系,最后使用奇异值分解(SVD)求解出刚性变换[11-13].但是DCP在处理部分点云配准时,精度较低.为了改进这一问题,部分到部分配准网络(PRNet)引入了关键点检测层[14],通过计算特征向量的L_2距离来判断每个点云的重要性,并选出若干L_2距离最大的点作为关键点,最后通过Gumbel-Softmax算法来确定重叠区域内这些关键点之间的对应关系.在部分点云的匹配问题中,这有利于减轻重叠点之外点的干扰,提高了部分点云配准的精度.相比传统方法,深度学习方法配准精度更高、速度更快.但是在部分重叠点云的场景下,现有的深度学习方法中网络泛化性及其配准精度尚未表现出满意的稳定性.

针对深度学习算法在点云配准中的这些不足,提出一种基于多尺度特征融合的点云配准算法.整体思路为利用多尺度的特征融合方式来提取更加精细的点云特征,通过引入多注意力模块,借助自注意力机制加强点云内部的特征关联

性,并借助交叉注意力机制加强不同点云之间的特征关联性;最后通过位置估计模块,计算出更加准确的虚拟对应点,从而提升配准精度.

1 问题描述

给定源点云 $X=\{X_i\in\mathbf{R}^3|i=1,2,\cdots,N\}$ 和目标点云 $Y=\{y_j\in\mathbf{R}^3|j=1,2,\cdots,M\}$,其中 N,M 分别表示源点云和目标点云的点的数量.点云配准的目的是寻找一个变换矩阵,变换矩阵通常包含旋转矩阵 $\boldsymbol{R}\in\mathbf{R}^{3\times3}$ 和平移向量 $\boldsymbol{t}\in\mathbf{R}^3$,使源点云在变换后与目标点云对齐,可以定义为

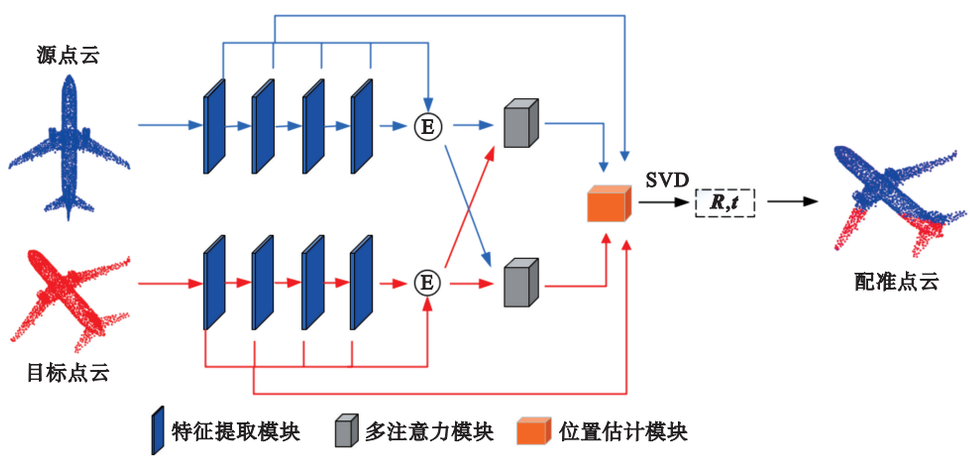
$$E(\boldsymbol{R},\boldsymbol{t})=\min_{\boldsymbol{R},\boldsymbol{t}}\sum_{i=1}^{N}\left\|\boldsymbol{R}X_i+\boldsymbol{t}-m(X_i)\right\|_2. \tag{1}$$

式中:$m(X_i)$ 为 X_i 点在目标点云 Y 中的对应点;$\|\cdot\|$ 为欧氏距离;$E(\boldsymbol{R},\boldsymbol{t})$ 为误差函数.点云配准的任务就是找到变换矩阵,使误差函数 $E(\boldsymbol{R},\boldsymbol{t})$ 最小化.

2 算法设计

2.1 整体思路

本文构造一种多尺度特征融合的网络,如图1所示.将源点云 X 和目标点云 Y 的三维坐标作为网络输入,通过特征描述将点云的三维坐标特征描述为包含局部信息的多维特征.特征提取模块将描述后的特征嵌入到高维的特征空间中,获取细化的特征信息.为了使配准更具有鲁棒性,采用不同尺度特征融合的策略,使提取的特征更具有代表性.通过多注意力模块来增强同一点云和不同点云的特征关联性.最后通过位置估计模块得到源点云的虚拟对应点,使用SVD算法计算变换矩阵.



注:Ⓔ为特征拼接.

图1 网络整体结构

Fig. 1 Overall network structure

2.2 特征提取

为了充分利用点云的局部结构,将每个三维点作为图的顶点,采用 K 近邻算法(KNN)为每个点构建局部图结构,与边缘卷积(EdgeConv)类似[15].不同的是,本文加入了两点之间的距离特征.以源点云为例,目标点云的构建过程与之相同.点云的局部图结构如式(2)所示:

$$f_{X_i}^{K}=\left(X_i,X_i-X_j,d_{ij}\right),\forall j\in N_i. \tag{2}$$

式中:X_i 为源点云 X 中第 i 个点的三维坐标;N_i 为 X_i 的最近点的索引集合;K 为近邻点个数;N_i 的基数由 K 决定;X_j 为 X_i 某个近邻点;d_{ij} 为点 X_i 与其近邻点 X_j 的欧氏距离特征.

经过KNN搜索近邻点后,通过特征拼接得到局部图结构.X_i 在第3个维度复制 K 次,维度变为 $(1,3,K)$,X_j 维度为 $(1,3,K)$,(X_i-X_j) 维度为 $(1,3,K)$,d_{ij} 维度为 $(1,1,K)$.因此,$f_{X_i}^{K}$ 维度为 $(1,7,K)$.

2.3 特征融合

先对包含点云局部信息的特征进行聚合操作,聚合函数选择为最大池化,特征聚合过程如下:

$$F_X^K=\Theta\left(h_\theta\left(\left(f_X^K\right)_{l-1}\right)\right). \tag{3}$$

式中:F_X^K 为 K 尺度下提取的特征;h_θ 为多层感知机;Θ 为最大池化;$(f_X^K)_{l-1}$ 为第 $l-1$ 层的描述特征,维度为 (N,C_{l-1},K),C_{l-1} 为第 $l-1$ 层的特征通道数.多层感知机将特征升维至 (N,C_l,K) 再进行最大池化.因此,F_X^K 的维度为 (N,C_l).

受点注意力网络(PANet)的启发[16],本文采

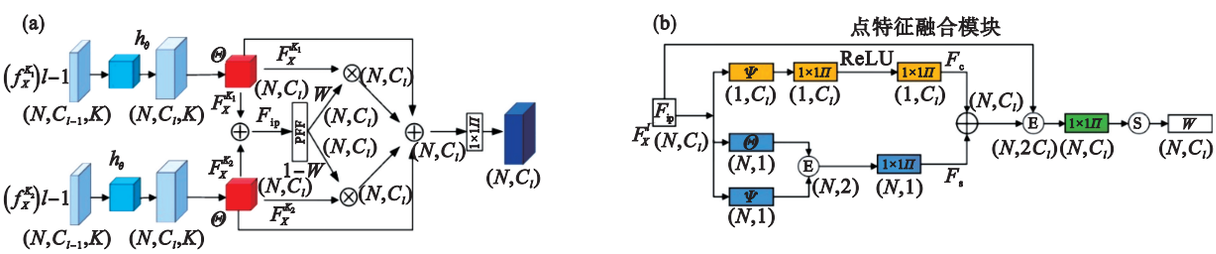
用2个不同尺度的KNN算法先对点云进行特征提取再进行特征融合.如图2a所示,构造点特征融合(PFF)模块为2个尺度的特征学习求出最佳权重.利用权重对所有的特征进行加权求和,从而获得具有更强代表性的综合特征.由于2个尺度的特征感受野存在重叠部分,如果仅用元素相加来融合特征会存在特征冗余问题,PFF模块可以自适应地识别大邻域尺度和小邻域尺度中点云特征的重要性.同时使用残差连接,以解决梯度消失的问题.特征融合的整个过程如下:

$$\hat{F}_X^l=\Pi\left(F_X^{K_1}\cdot W+F_X^{K_2}\cdot(1-W)+F_{\mathrm{ip}}\right). \tag{4}$$

式中:$\hat{F}_X^l$为第l层提取的综合特征,维度为(N,C_l);Π为卷积核大小为1×1的卷积;$F_X^{K_1}$为K_1尺度下提取的特征;$F_X^{K_2}$为K_2尺度下提取的特征,维度均为(N,C_l);W为学习得到的权重,维度为(N,C_l);F_{ip}为输入特征.

PFF模块的结构如图2b所示.将2个尺度的KNN算法所提取的聚合特征进行元素相加得到F_{ip},维度为(N,C_l),作为PFF的输入.

$$F_{\mathrm{ip}}=F_X^{K_1}+F_X^{K_2}. \tag{5}$$



注:⊕为逐元素求和;Ⓢ为Sigmoid激活函数;⊗为用矩阵元素逐一乘以权重;Θ为最大池化.

图2　特征提取层

Fig. 2　Feature extraction layer

(a)—特征提取流程;(b)—PFF模块结构.

分别提取通道注意力特征F_{c}以及空间注意力特征F_{s},过程如下:

$$F_{\mathrm{c}}=\Pi\left(\Gamma\left(\Pi\left(\Psi\left(F_{\mathrm{ip}}\right)\right)\right)\right), \tag{6}$$

$$F_{\mathrm{s}}=\Pi\left(\Omega\left(\Theta\left(F_{\mathrm{ip}}\right),\Psi\left(F_{\mathrm{ip}}\right)\right)\right). \tag{7}$$

式中:Ψ为平均池化;Γ为ReLU激活函数;Ω为特征拼接.

在提取通道注意力特征时,沿着空间维度进行平均池化.在提取空间注意力特征时,沿着通道维度进行最大池化和平均池化.因此,F_{c}的维度为$(1,C_l)$,F_{s}的维度为$(N,1)$.

利用广播机制对2个特征进行元素相加,并通过加入输入特征F_{ip}来学习权重W.

$$W=S\left(\Pi\left(E\left(F_{\mathrm{ip}},F_{\mathrm{c}}+F_{\mathrm{s}}\right)\right)\right). \tag{8}$$

式中,S为sigmoid函数.

特征提取层数设置为4.特征描述只进行1次,因此第1层的输入特征维度为$(N,7,K)$.使用多层感知机对描述的特征进行升维操作,并将升维后的特征作为下1层的输入特征.因此,第l个特征提取层的输入特征维度为(N,C_{l-1},K).每层的特征提取过程均与图2类似.最后将每层提取的特征沿着特征通道维度进行特征拼接得到总特征.

$$\boldsymbol{F}_X=r\left(F_X^{l_1},F_X^{l_2},F_X^{l_3},F_X^{l_4}\right). \tag{9}$$

式中:$\boldsymbol{F}_X$为点云总特征矩阵;r为特征拼接;$F_X^{l_i}(i=1,2,3,4)$为第l层特征提取层输出的逐点特征.

2.4　多注意力模块

为了提高模型的泛化能力,引入了注意力机制.注意力机制能区分点对之间的重要程度,利用点云特征的点乘运算来获取点对之间的相似度.将相似度高的邻域点云赋予更高的权重,以此来提升点云之间的关联性.点云配准的主要任务是找出准确的对应关系,增强点云之间的关联性有助于突出关键点的独特性,从而更容易找到准确的对应关系.本文采用偏移注意力机制[15],它是自注意力机制的变体,利用逐元素相减计算出注意力特征与输入特征之间的差异,并且加入1个残差连接来得到最后的注意力特征.

如图3所示,计算过程主要分为3个阶段:①计算特征相似度;②计算注意力特征;③计算特征差异.首先,将可学习的线性变换矩阵作用于输入特征$\boldsymbol{F}_X$,生成查询矩阵$\boldsymbol{Q}$,键矩阵$\boldsymbol{U}$和值矩阵$\boldsymbol{V}$,公式如下:

$$\boldsymbol{Q}=\boldsymbol{F}_XW_1;\boldsymbol{U}=\boldsymbol{F}_XW_2;\boldsymbol{V}=\boldsymbol{F}_XW_3. \tag{10}$$

式中:$\boldsymbol{F}_X$维度为(N,d_1);W_1,W_2,W_3是对应可学习的线性变换,均通过梯度下降从特征中学习得到,维度均为(d_1,d_1);$\boldsymbol{Q},\boldsymbol{U},\boldsymbol{V}$的维度均为$(N,d_1)$.

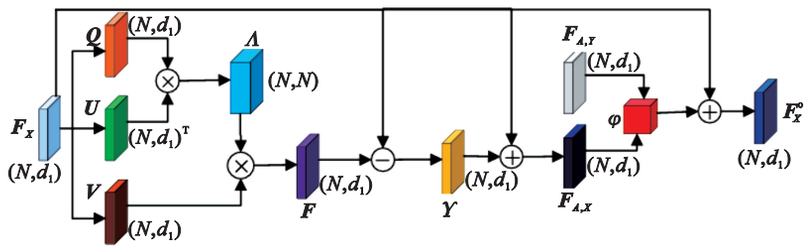


图3 多注意力机制

Fig. 3 Multi-attention mechanism

将查询矩阵和键矩阵进行点乘得到注意力矩阵$\boldsymbol{\Lambda}$,维度为(N,N).然后对注意力矩阵的元素进行指数归一化操作,将更新后的注意力矩阵作为权重矩阵.

$$\boldsymbol{\Lambda}=\boldsymbol{Q}\cdot\boldsymbol{U}^{\mathrm{T}}, \tag{11}$$

$$\lambda_{ij}:=\delta\left(\lambda_{ij}\right)=\frac{\mathrm{e}^{\lambda_{ij}}}{\sum\limits_{z}\mathrm{e}^{\lambda_{zj}}}, \tag{12}$$

$$\lambda_{ij}:=\frac{\lambda_{ij}}{\sum\limits_{z}\lambda_{iz}}. \tag{13}$$

式中:λ_{ij}为注意力矩阵$\boldsymbol{\Lambda}$中的某个元素;z为游动指标标记;λ_{zj}为注意力矩阵中某列的所有元素,此时z遍历全部行;λ_{iz}为注意力矩阵中的某行的所有元素,此时z遍历全部列;δ为指数归一化操作.

偏移注意力的输出特征$\boldsymbol{F}$是权重矩阵所对应值矩阵中向量加权和,维度为(N,d_1).

$$\boldsymbol{F}=\boldsymbol{\Lambda}\cdot\boldsymbol{V}. \tag{14}$$

将输入特征与自注意力特征进行逐元素相减得到特征差异.对特征差异依次进行线性变换,批量归一化和ReLU激活函数(Υ)处理,最后再进行残差连接得到注意力特征$F_{A,X}$,其维度为(N,d_1).

$$\boldsymbol{F}_{A,X}=\Upsilon\left(\boldsymbol{F}_X-\boldsymbol{F}\right)+\boldsymbol{F}_X. \tag{15}$$

偏移注意力加强了同一点云的特征关联性.为提升配准的准确率,再加入交叉注意力.交叉注意力通过关注不同点云中相似的特征来学习细化逐点特征,有助于网络找到更准确的对应关系.其过程与自注意力机制类似,用φ来表示.最后加入一个残差连接,得到最后的输出特征.

$$\boldsymbol{F}_X^{\circ}=\varphi\left(\boldsymbol{F}_{A,X},\boldsymbol{F}_{A,Y}\right)+\boldsymbol{F}_X. \tag{16}$$

式中:$\boldsymbol{F}_X$为原输入特征;$\boldsymbol{F}_{A,X}$为源点云的偏移注意力特征;$\boldsymbol{F}_{A,Y}$为目标点云的偏移注意力特征,计算方法与源点云相同;$\boldsymbol{F}_X^{\circ}$为整个注意力模块输出的源点云的注意力特征,维度为(N,d_1),目标点云与之类似.

2.5 位置估计

在部分点云的场景中,由于2个点云的点不一定一一对应,如果直接通过匹配矩阵来选择对应点对,会出现大量错误匹配的情况.为了克服直接匹配的缺点,采用虚拟对应点匹配的策略,在目标点云上计算出源点云的虚拟对应点,利用源点云与虚拟对应点计算旋转平移矩阵和平移向量.

通过计算源点云中的1个点被分配到目标点云每个点的概率矩阵,再使用该矩阵对目标点云每个点进行加权求和得到虚拟对应点.使用2个点云特征的点乘来计算特征相似度矩阵,再进行指数归一化得到概率矩阵.受跳跃注意力基础的对应过滤网络(SACF-Net)计算对应矩阵策略的启发[17],本文进行了多重点特征匹配.利用注意力模块输出的注意力特征与特征提取阶段的融合特征分别计算概率矩阵,并且将2个矩阵进行元素相乘.这样会使正确对应点的概率越大,生成的虚拟对应点越准确.得到概率矩阵$\boldsymbol{G}$的表达式如下:

$$\boldsymbol{G}=\delta\left(F_X^{l_2}\left(F_Y^{l_2}\right)^{\mathrm{T}}\odot\boldsymbol{F}_X^{\circ}\left(\boldsymbol{F}_Y^{\circ}\right)^{\mathrm{T}}\right). \tag{17}$$

式中:$F_X^{l_2}$为源点云在特征提取层第2层提取的特征,维度为(N,C_{l_2});$F_Y^{l_2}$为目标点云在特征提取层第2层提取的特征,维度为(M,C_{l_2});$\boldsymbol{F}_Y^{\circ}$为目标点云注意力特征,维度为(M,d_1);概率矩阵$\boldsymbol{G}$的维度为(N,M).

虚拟对应点$\hat{y}$的表达式如下:

$$\hat{y}=\boldsymbol{G}\cdot Y. \tag{18}$$

式中:Y为目标点云坐标,维度为$(M,3)$.

因此,虚拟对应点$\hat{y}$的维度为$(N,3)$.得到虚拟对应关系$(x_i,\hat{y}_i)$后,刚性变换可以使用SVD算法解出.

2.6 损失函数

点云配准主要是为了优化使源点云与目标点云对齐的变换参数.在点云配准研究中,最常见的损失函数是直接最小化预测值与真实值之间的误差.网络模型中的主要优化对象为位置估计模块中的概率矩阵,因此损失函数使用均方根误差.

$$L=\sqrt{\left\|\tilde{\boldsymbol{R}}^{\mathrm{T}}\boldsymbol{R}_{\mathrm{g}}-\boldsymbol{I}\right\|^{2}+\left\|\tilde{\boldsymbol{t}}-\boldsymbol{t}_{\mathrm{g}}\right\|^{2}}. \tag{19}$$

式中:L为损失函数;$\tilde{\boldsymbol{R}}$和$\boldsymbol{R}_{\mathrm{g}}$分别表示网络预测出的旋转矩阵和实际的旋转矩阵;$\boldsymbol{I}$为单位矩阵;$\tilde{\boldsymbol{t}}$和$\boldsymbol{t}_{\mathrm{g}}$分别表示网络预测出的平移向量和实际的平移向量.通过最小化损失函数来使网络预测的矩阵更接近实际的矩阵.

3　实　　验

为验证算法的有效性,将ModelNet40数据集分为完整点云与部分点云2个场景,并与传统算法及一些流行的深度学习算法进行对比.

3.1　数据集预处理

ModelNet40数据集包含40个不同类别的点云数据[18].训练集包含9 840个样本,测试集包含2 468个样本.为降低计算资源消耗,再将所有样本中的点数量从2 048均匀下采样至1 024.在训练和测试期间,网络为每个对象随机生成旋转平移变换,旋转角度的区间为[0°,45°],平移向量的值被限制在区间[−0.5,0.5].

在部分点云场景下对ModelNet40数据集的处理方式与PRNet相同.对源点云随机进行旋转平移变换生成目标点云.采用最远点采样算法[19],通过在2个点云的坐标空间中随机放置1个点,搜索N个最近点来作为局部点云场景中的源点云与目标点云.

3.2　实验环境与参数设置

计算机GPU为RTX 4090;显存24 GB;Windows11系统,使用PyTorch框架对点云进行配准.

网络整体结构如图1所示,2个尺度的KNN算法搜索的近邻点个数分别为K=10和K=20.每个特征提取层对输入特征进行升维的维度设置为[64,64,128,256].网络训练时,批大小设置为16,测试时批大小设置为16.使用Adam优化器,初始学习率设置为0.001,迭代次数设置为250,在迭代周期为50,100,200处将学习率乘以0.1.

3.3　评估方法

采用的评估标准为真实值与预测值之间的均方误差、均方根误差和平均绝对误差.误差值越接近0,代表网络的配准效果越好.

$$S_{\mathrm{ME}}=\frac{1}{n}\sum_{i=1}^{n}(V_{\mathrm{g}}^{(i)}-V_{\mathrm{p}}^{(i)})^{2}, \tag{20}$$

$$S_{\mathrm{MR}}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(V_{\mathrm{g}}^{(i)}-V_{\mathrm{p}}^{(i)})^{2}}, \tag{21}$$

$$M_{\mathrm{AE}}=\frac{1}{n}\sum_{i=1}^{n}\left|V_{\mathrm{g}}^{(i)}-V_{\mathrm{p}}^{(i)}\right|. \tag{22}$$

式中:S_{ME}为均方误差;S_{MR}为均方根误差;M_{AE}为平均绝对误差;$V_{\mathrm{g}}^{(i)}$为第i个样本的真实旋转变换$R_{\mathrm{g}}^{(i)}$或真实平移变换$t_{\mathrm{g}}^{(i)}$;$V_{\mathrm{p}}^{(i)}$为第i个样本的预测旋转变换$R_{\mathrm{p}}^{(i)}$或预测平移变换$t_{\mathrm{p}}^{(i)}$;n为样本数量.在测试过程中n=2 468.

3.4　完整点云场景

在完整点云场景中,与传统算法ICP,GO−ICP,FGR以及深度学习算法PointNetLK,DCP进行对比.实验在干净点云、未知种类点云、噪声点云3种条件下进行.

1)在干净点云的实验中,9 840个样本用于训练,2 468个样本用于测试.原始ModelNet40数据集中的点作为源点云,随机生成旋转平移变换并作用于源点云以生成目标点云.为了消除点的相对顺序对模型的影响,对点的顺序进行随机重排.配准误差如表1所示.由于ICP,GO−ICP对初始位置非常敏感,一旦初始位置相差过大,配准性能会大幅下降.FGR通过对点进行特征描述,在特征空间寻找对应点,因此效果优于ICP与GO−ICP,但误差依然较大.PointNetLK作为深度学习算法,由于它仅利用全局特征来进行配准,面对复杂结构的点云时,配准性能仍会下降得较为明显.DCP使用动态图卷积神经网络(DGCNN)的特征提取网络可有效提取全局信息与局部信息,因此配准效果明显优于其他4种算法,而本文算法的误差比DCP更小.为了验证算法的泛化性,在斯坦福数据集上进行配准.在预处理阶段,先将数据进行体素下采样,再随机采样4 096个点进行配准.可视化结果如图4所示.蓝色为源点云,红色为目标点云.结果表明,本文算法在斯坦福数据集上也具有很高的配准精度.

2)点云在未知种类点云的实验中,ModelNet40数据集中的前20个种类点云用于训练模型,后20个种类点云用于测试.其他设置均与干净点云的实验一致.配准误差如表2所示.由于传统算法不需要训练,因此与干净点云实验的配准误差相差较小.PointNetLK与DCP由于训练样本少了20个种类,配准性能下降明显.本文算法虽然相比干净点云实验的配准误差有所上升,但仍低于其他算法.

表 1 完整点云场景下干净点云配准效果比较

Table 1 Comparison of clean point cloud registration effects in the complete point cloud scenario

算法	$S_{ME}(\boldsymbol{R})$	$S_{MR}(\boldsymbol{R})$	$M_{AE}(\boldsymbol{R})$	$S_{ME}(\boldsymbol{t})$	$S_{MR}(\boldsymbol{t})$	$M_{AE}(\boldsymbol{t})$
ICP	894.897	29.914	23.544	0.084 6	0.290	0.248
GO-ICP	140.477	11.852	2.588	0.000 6	0.025	0.007
FGR	87.661	9.362	1.999	0.000 1	0.013	0.002
PointNetLK	227.870	15.095	4.225	0.000 4	0.022	0.005
DCP	1.307	1.143	0.770	0.000 0	0.001	0.001
本文算法	0.063	0.252	0.126	0.000 0	0.000	0.000

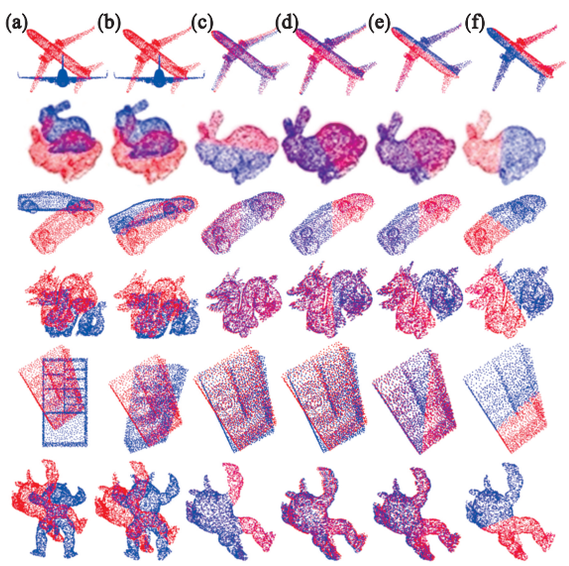


图 4 完整点云配准的可视化

Fig. 4 Visualization of complete point cloud registration

(a)—初始位置;(b)—ICP;(c)—FGR;(d)—PointNetLK;(e)—DCP;(f)—本文算法.

3) 与 DCP 算法中的实验一样,对源点云添加 $\mathcal{N}(0,0.01)$ 的高斯噪声并限制在 $[-0.05,0.05]$ 作为随机抖动,其中 $\mathcal{N}$ 表示高斯分布的记号.利用干净点云训练实验中得到的模型直接对噪声点云进行配准.配准误差如表 3 所示.FGR 对噪声很敏感,GO-ICP 与 PointNetLK 对噪声有鲁棒性,但是配准误差较大.DCP 展现出很好的配准精度,而本文算法的配准误差低于 DCP.

3.5 部分点云场景

为了模拟部分点云场景,采用最远点采样算法,将包含 1 024 个点的完整点云切割成包含 768 个点的部分点云.在部分点云场景中,除了与 ICP, GO-ICP, FGR, PointNetLK, DCP 等算法对比,还加入 1 个用于处理部分点云场景的 PRNet,迭代距离感知相似度矩阵卷积与相互监督点消除(IDAM)[20]、深度最佳伙伴选择(DeepBBS)[21]、PANet 进行对比.实验同样在干净点云、未知类别点云、噪声点云 3 种条件下进行.

表 2 完整点云场景下未知种类点云配准效果比较

Table 2 Comparison of unknown-category point cloud registration effects in the complete point cloud scenario

算法	$S_{ME}(\boldsymbol{R})$	$S_{MR}(\boldsymbol{R})$	$M_{AE}(\boldsymbol{R})$	$S_{ME}(\boldsymbol{t})$	$S_{MR}(\boldsymbol{t})$	$M_{AE}(\boldsymbol{t})$
ICP	892.601	29.876	23.626	0.086 0	0.293	0.251
GO-ICP	192.258	13.865	2.914	0.000 4	0.022	0.006
FGR	97.002	9.848	1.445	0.000 1	0.013	0.002
PointNetLK	306.323	17.502	5.280	0.000 7	0.028	0.007
DCP	9.923	3.150	2.007	0.000 0	0.005	0.003
本文算法	0.521	0.721	0.504	0.000 0	0.001	0.000

表 3 完整点云场景下噪声点云配准效果比较

Table 3 Comparison of noisy point cloud registration effects in the complete point cloud scenario

算法	$S_{ME}(\boldsymbol{R})$	$S_{MR}(\boldsymbol{R})$	$M_{AE}(\boldsymbol{R})$	$S_{ME}(\boldsymbol{t})$	$S_{MR}(\boldsymbol{t})$	$M_{AE}(\boldsymbol{t})$
ICP	882.564	29.707	23.557	0.084 5	0.290	0.249
GO-ICP	131.182	11.453	2.534	0.000 5	0.023	0.004
FGR	607.694	24.651	10.055	0.011 8	0.108	0.027
PointNetLK	256.155	16.004	4.595	0.000 4	0.021	0.005
DCP	1.169	1.081	0.737	0.000 0	0.001	0.001
本文算法	0.043	0.208	0.121	0.000 0	0.000	0.000

1) 在干净点云的实验中,对已生成的源点云与目标点云分别进行分割,以确保2个点云的点不一一对应.配准误差如表4所示.传统算法在面对部分点云时,误差会变得更大,这是因为2个点云的点不是相互对应的,传统算法更容易陷入局部最优.DCP在部分点云场景下误差也变得很大,这是由于无法为丢失部分的点云生成更准确的虚拟对应点.可视化结果如图5所示.为了验证本文算法的泛化性,在3DMatch数据集上进行配准[22].在预处理阶段,先将数据进行体素下采样,再随机采样4 096个点进行配准.蓝色为源点云,红色为目标点云.由于没有使用3DMatch数据集训练,因此配准的误差较大,但精度优于其他几种算法.

表4　部分点云场景下干净点云配准效果比较

Table 4　Comparison of clean point cloud registration effects in partial point cloud scenarios

算法	$S_{ME}(\boldsymbol{R})$	$S_{MR}(\boldsymbol{R})$	$M_{AE}(\boldsymbol{R})$	$S_{ME}(\boldsymbol{t})$	$S_{MR}(\boldsymbol{t})$	$M_{AE}(\boldsymbol{t})$
ICP	1 134. 552	33. 683	25. 045	0. 085 6	0. 293	0. 250
GO-ICP	195. 985	13. 999	3. 165	0. 001 1	0. 033	0. 012
FGR	126. 288	11. 238	2. 832	0. 000 9	0. 030	0. 008
PointNetLK	280. 044	16. 735	7. 550	0. 002 0	0. 045	0. 025
DCP	45. 005	6. 709	4. 448	0. 000 7	0. 027	0. 020
PRNet	10. 235	3. 199	1. 454	**0. 000 3**	**0. 016**	0. 010
IDAM	8. 726	2. 954	1. 346	0. 000 4	0. 021	**0. 005**
DeepBBS	4. 567	2. 137	1. 541	0. 000 5	0. 022	0. 017
PANet	5. 919	2. 433	1. 654	0. 000 5	0. 022	0. 017
本文算法	**2. 290**	**1. 513**	**1. 013**	**0. 000 3**	0. 017	0. 012

注:表中粗体数据表示最好的结果.

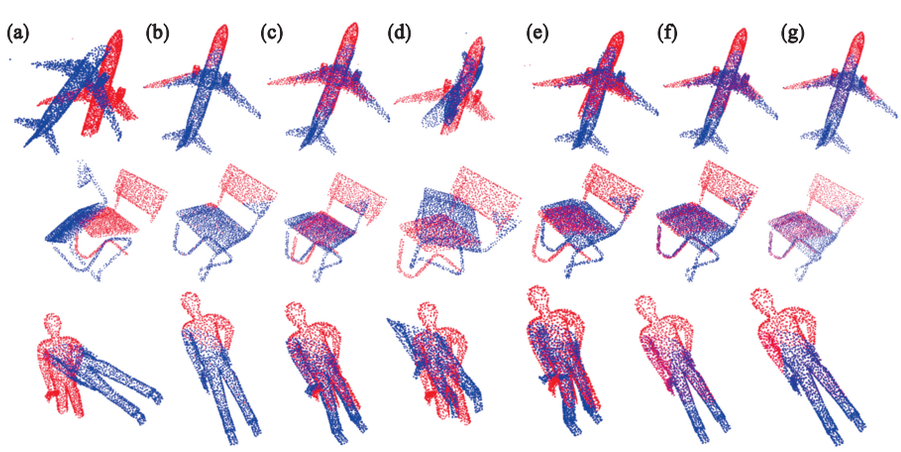


图5　部分点云配准的可视化

Fig. 5　Visualization of partial point cloud registration

(a)—初始位置;(b)—真实变换;(c)—FGR;(d)—DCP;(e)—PRNet;(f)—IDAM;(g)—本文算法.

2) 在未知种类点云的实验中,模型利用前20个种类训练,后20个种类用于测试.其他参数设置与完整点云场景相同.配准误差如表5所示.由于传统算法不需要训练,因此与干净点云实验的配准效果相差较小.另外3个深度学习算法误差明显变大.但本文算法的误差与干净点云实验相差很小,也小于其他算法,可见算法在部分点云场景下依然有良好的泛化性.

3) 在噪声点云实验中,与完整点云场景不同,对已经生成的源点云与目标点云独立添加$\mathcal{N}(0,0.01)$的高斯噪声并限制在$[-0.05,0.05]$作为随机抖动,然后将其送入网络训练.配准误差如表6所示.所有算法的误差相比干净点云均明显变大,但本文算法的误差依然是所有算法中最小的.

3.6　消融实验

为了验证算法各模块的有效性,分别进行消

融实验.首先验证特征提取网络(PF),将其替换为PointNet(PN)和DGCNN(DG)的特征提取网络.然后验证多注意力机制(MA),在网络中去除MA.最后验证位置估计模块(PE),仅使用输出的注意力特征来计算虚拟对应点(PE2).在部分点云场景下的干净点云数据集上,分别采用以下不同的网络结构进行配准:PN+MA+PE,DG+MA+PE,PF+PE,PF+MA+PE2,PF+MA+PE(本文结构).配准的误差如表7所示.因为PointNet的最大池化操作仅提取出点云的全局特征,丢失了许多局部的关键性特征.本文算法通过2个尺度的KNN算法分层地提取特征,所提取的特征比DGCNN更具有代表性.去除了注意力模块后误差变得很大,多注意力模块有效地加强了点云特征的关联性.使用PE2进行配准的误差略大于使用PE,说明位置估计模块生成了更准确的虚拟对应点.

表5 部分点云场景下未知种类点云配准效果比较

Table 5 Comparison of unknown-category point cloud registration effects in partial point cloud scenario

算法	$S_{ME}(\boldsymbol{R})$	$S_{MR}(\boldsymbol{R})$	$M_{AE}(\boldsymbol{R})$	$S_{ME}(\boldsymbol{t})$	$S_{MR}(\boldsymbol{t})$	$M_{AE}(\boldsymbol{t})$
ICP	1 217.618	29.915	25.455	0.086 0	0.293	0.251
GO-ICP	157.072	34.894	2.940	0.000 9	0.031	0.010
FGR	98.635	12.533	1.952	0.001 4	0.038	0.007
PointNetLK	526.401	9.932	9.665	0.003 7	0.061	0.033
DCP	95.431	22.943	6.954	0.001 0	0.034	0.025
PRNet	15.624	3.953	1.712	**0.000 3**	0.017	0.011
IDAM	11.710	3.422	1.761	0.000 4	0.022	**0.005**
DeepBBS	6.342	2.518	1.968	0.000 8	0.028	0.021
PANet	13.608	3.689	2.499	0.000 9	0.030	0.019
本文算法	**2.696**	**1.642**	**1.105**	**0.000 3**	**0.016**	0.012

表6 部分点云场景下噪声点云配准效果比较

Table 6 Comparison of noisy point cloud registration effects in partial point cloud scenario

算法	$S_{ME}(\boldsymbol{R})$	$S_{MR}(\boldsymbol{R})$	$M_{AE}(\boldsymbol{R})$	$S_{ME}(\boldsymbol{t})$	$S_{MR}(\boldsymbol{t})$	$M_{AE}(\boldsymbol{t})$
ICP	1 229.670	35.067	25.564	0.086 0	0.294	0.250
GO-ICP	150.320	12.261	2.845	0.000 8	0.028	0.029
FGR	764.671	27.653	13.794	0.004 8	0.070	0.039
PointNetLK	397.575	19.939	9.076	0.003 2	0.057	0.032
DCP	47.378	6.883	4.534	0.000 8	0.028	0.021
PRNet	18.691	4.323	2.051	**0.000 3**	**0.017**	**0.012**
IDAM	13.898	3.728	1.852	0.000 5	0.023	0.011
DeepBBS	12.568	3.548	**0.974**	0.002 2	0.047	0.017
PANet	**7.728**	**2.780**	1.887	0.043 2	0.021	0.016
本文算法	12.966	3.601	2.308	**0.000 3**	**0.017**	**0.012**

表7 不同网络结构的配准效果比较

Table 7 Comparison of registration effects of different network structures

算法	$S_{ME}(\boldsymbol{R})$	$S_{MR}(\boldsymbol{R})$	$M_{AE}(\boldsymbol{R})$	$S_{ME}(\boldsymbol{t})$	$S_{MR}(\boldsymbol{t})$	$M_{AE}(\boldsymbol{t})$
PN+MA+PE	21.504	4.637	2.856	0.000 5	0.024	0.017
DG+MA+PE	9.075	3.012	1.954	0.000 3	0.019	0.014
PF+PE	40.711	6.380	4.378	0.000 7	0.027	0.019
PF+MA+PE2	3.583	1.893	1.127	0.000 4	0.019	0.014
PF+MA+PE	**2.290**	**1.513**	**1.013**	**0.000 3**	**0.017**	**0.012**

3.7 网络参数量对比

比较了部分点云场景中对比算法的网络参数量与计算复杂度.在部分点云场景中测试点云图像中点的个数N为768.ICP,GO-ICP,FGR为传统算法,没有网络参数.如表8所示,PointNetLK的网络参数量最小.

表8　网络参数量比较

Table 8　Comparison of network parameter sizes

算法	网络参数×10^{-6}	计算复杂度
ICP	—	$O(N^2)$
GO-ICP	—	$O(N^2)$
FGR	—	$O(N^2)$
PointNetLK	**0.16**	$O(N)$
DCP	5.57	$O(N^2)$
PRNet	10.81	$O(N^2)$
IDAM	1.87	$O(N^2)$
DeepBBS	5.67	$O(N^2)$
PANet	2.42	$O(N^2)$
本文算法	5.15	$O(N^2)$

4　结　　语

本文提出一种多尺度特征融合的点云配准算法.首先,利用KNN算法来搜索2个尺度的点云局部邻域信息,并使用PFF模块自适应地融合2个尺度的特征信息;然后,引入了注意力机制来提高点云之间的特征关联性,增强了网络对关键点特征的学习能力;最后,采用多重特征匹配策略,使生成的虚拟对应点更加准确.在2个场景中的仿真实验以及消融实验结果表明,该算法对完整点云数据和部分点云数据均具有很高的配准精度,泛化能力也更强.但是算法在部分噪声的点云时配准精度明显降低.未来需要进一步优化每个模块,调整网络结构,使其对部分噪声点云有更好的鲁棒性.

参考文献:

[1] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Sensor Fusion IV: Control Paradigms and Data Structures.Boston:SPIE,1992: 586-606.

[2] Yang J L, Li H D, Campbell D, et al. Go-ICP: a globally optimal solution to 3D ICP point-set registration[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 38(11): 2241-2254.

[3] Chetverikov D, Svirko D, Stepanov D, et al. The trimmed iterative closest point algorithm [C]//2002 International Conference on Pattern Recognition. Quebec: IEEE, 2002: 545-548.

[4] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration [C]//2009 IEEE International Conference on Robotics and Automation. Rome: IEEE, 2009: 3212-3217.

[5] Zhou Q Y, Park J, Koltun V. Fast global registration[C]// Computer Vision-ECCV 2016. Cham: Springer International Publishing, 2016: 766-782.

[6] Magnusson M, Lilienthal A, Duckett T. Scan registration for autonomous mining vehicles using 3D-NDT [J]. *Journal of Field Robotics*, 2007, 24(10): 803-827.

[7] Aiger D, Mitra N J, Cohen-Or D. 4-points congruent sets for robust pairwise surface registration[C]// Special Interest Group on Computer Graphics and Interactive Techniques Conference. New York: Association for Computing Machinery,2008: 1-10.

[8] Charles R Q, Hao S, Mo K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, 2017: 77-85.

[9] Aoki Y, Goforth H, Srivatsan R A, et al. PointNetLK: robust & efficient point cloud registration using PointNet [C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, 2020: 7156-7165.

[10] Sarode V, Li X Q, Goforth H, et al. PCRNet: point cloud registration network using PointNet encoding [EB/OL]. (2019-08-21) [2023-10-12]. https://arxiv. org/abs/1908.07906.

[11] Wang Y, Solomon J. Deep closest point: learning representations for point cloud registration[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, 2020: 3522-3531.

[12] Wang Y, Sun Y B, Liu Z W, et al. Dynamic graph CNN for learning on point clouds[J]. *ACM Transactions on Graphics (TOG)*, 2018, 38: 1-12.

[13] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]//31st Conference on Neural Information Processing Systems (NIPS 2017). Long Beach, 2017:1-15.

[14] Wang Y, Solomon J M. PRNet: self-supervised learning for partial-to-partial registration [EB/OL]. (2019-10-27) [2021-01-03]. https://arxiv.org/abs/1910.12240.

[15] Wu Y, Yao Q L, Fan X L, et al. PANet: a point-attention based multi-scale feature fusion network for point cloud registration[J]. *IEEE Transactions on Instrumentation and Measurement*, 2023, 72: 2512913.

[16] Guo M H, Cai J X, Liu Z N, et al. PCT: point cloud transformer[J]. *Computational Visual Media*, 2021, 7(2): 187-199.

[17] Wu Y, Hu X D, Zhang Y, et al. SACF-net: skip-attention based correspondence filtering network for point cloud registration[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023, 33(8): 3585-3595.

[18] Wu Z R, Song S R, Khosla A, et al. 3D ShapeNets: a deep representation for volumetric shapes [C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, 2015: 1912-1920.

[19] Eldar Y, Lindenbaum M, Porat M, et al. The farthest point strategy for progressive image sampling [J]. *IEEE Transactions on Image Processing*, 1997, 6(9): 1305-1315.

[20] Li J H, Zhang C H, Xu Z Y, et al. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration [C]// Computer Vision—ECCV 2020. Cham: Springer, 2020: 378-394.

[21] Hezroni I, Drory A, Giryes R, et al. DeepBBS: deep best buddies for point cloud registration[C]//2021 International Conference on 3D Vision (3DV). London, 2022: 342-351.

[22] Zeng A, Song S R, Nießner M, et al. 3Dmatch: learning local geometric descriptors from RGB-D reconstructions [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, 2017: 199-208.