doi:10.11887/j.cn.201406003 http://journal.nudt.edu.cn

基于快速点特征直方图的特征点云迭代插值配准算法*

陆 军,彭仲涛
(哈尔滨工程大学 自动化学院,黑龙江 哈尔滨 150001)

摘 要:为提高三维激光扫描点云数据的配准精度,提出了一种基于快速点特征直方图特征的迭代插值配准方法。配准过程中,点云数据获取时受扫描仪分辨率影响,点云局部或整体密度偏小,两次测量点云数据的相同位置不存在完全相同的点,以致对应点之间存在误差。为减小误差对配准精度影响,引入迭代插值方法,增加点云整体密度。通过计算关键点的快速点特征直方图的特征寻找对应关系,使用随机采样一致算法去除错误对应关系,对对应点协方差矩阵进行奇异值分解求得粗配准旋转平移矩阵,再使用迭代最近点算法进行点云的精确配准。实验结果表明,改进的配准方法简单、稳定可靠、计算速度有所增加,有效地提高了配准精度。

关键词:点云配准;迭代插值;关键点;快速点特征直方图;迭代最近点
中图分类号:TP391 **文献标志码**:A **文章编号**:1001-2486(2014)06-012-06

Iterative interpolation point cloud registration algorithm based on fast point feature histograms

LU Jun, *PENG Zhongtao*
(College of Automation, Harbin Engineering University, Harbin 150001, China)

Abstract: To improve the registration accuracy of point cloud data generated by 3D laser scanning, a new method of iterative interpolation registration based on fast point feature histograms (FPFH) was proposed. Due to the effect of the scanner's resolution in the process of registration, partial or overall density of obtained point cloud data was smaller so that there were no same points even the measuring locations of point cloud data were fixed. Therefore, errors existed between corresponding points. In order to reduce the influence of these errors on the registration accuracy, iterative interpolation method was introduced to increase the overall density of point cloud. The FPFH features of key points were used to find the corresponding relationship; random sample consensus algorithm was used to remove the false correspondence between two point clouds; then the coarse registration rotation and translation matrix was gotten by using singular value decomposition algorithm on the corresponding covariance matrix; at last, the iterative closest point algorithm was employed for the precise registration of point clouds. The experimental results show that the improved registration algorithm is simple, stable and reliable and its computation velocity is faster. This method effectively improves the accuracy of registration results.

Key words: point cloud registration; iterative interpolation; key points; fast point feature histograms; iterative closest point

三维点云数据配准技术一直是逆向工程、计算机视觉、模式识别、曲面质量检测及摄影测量学等领域的研究热点与难点。在逆向工程中,三维数字化技术是首要环节。在物体数字化测量过程中,受被测物体空间位置、几何外形和测量方式的影响,设备仪器要从不同视角对物体进行多次定位测量,然后对测量数据配准拼接,整合到同一坐标下,即点云配准。

目前点云配准主要有三类方法:基于特征求对应点法、基于统计学求刚体变换法、迭代最小误差法。基于特征求对应点法主要是通过寻找视图间的对应点,利用对应点求取视图间的坐标变换关系。这就将配准问题转化为寻找对应点的问题。Rusu 提出的基于快速点特征直方图(Fast Point Feature Histograms, FPFH)[1-4]的三维点云配准算法,以点与领域内点法向量夹角作为特征构建直方图,实现了三维点云的配准。Gelfand[5]提出的基于体积积分不变(Integral Volume Invariant, IVI)算法的三维点云配准算法根据积分不变量,提取特征,完成了三维点云数据的配准。Johnson[6]提出的旋转图像特征描述子实现了3D场景中的曲面匹配和模型识别。视点特征直方图

* **收稿日期**:2014-04-11
基金项目:黑龙江省自然科学基金资助项目(F201123);中央高校基本科研业务费专项基金资助项目(HEUCFX41304)
作者简介:陆军(1969—),男,黑龙江哈尔滨人,教授,博士,E-mail:lujun0260@sina.com

(Viewpoint Feature Histogram ,VFH)[7] 是源于 FPFH 描述子的,它是一种新的特征表示形式,应用于点云聚类识别和六自由度位姿估计问题。基于统计学求刚体变换法就是将变换空间离散化,并建立某种配准误差评价函数,使用标准最优化技术来确定两个点云间的最优匹配。典型算法是 Magnusson 提出的正态分布变换(Normal Distributions Transform ,NDT)算法[8]。迭代最小误差法最具代表性的是 Besl 等提出的迭代最近点(Iterative Closest Point ,ICP)算法[9],算法迭代查找对应关系,寻找最优三维的刚体变换。十几年来,ICP 算法被不断地完善和改进。Turk 等[10]提出统一采样法,改进 ICP 算法的初始点集采样方法。Fan 等[11]提出根据点到曲面最小距离确立对应点集。文献[12]提出了一种迭代最近线(Iterative Closest Line,ICL)算法,根据两幅点云中连线寻找对应点配准。Johnson 和 Kang[13]提出了适用于彩色点云纹理信息的数据配准方法。配准前点云预处理非常重要。因为点云质量的好坏直接影响求得的特征的准确程度。

1　迭代插值算法描述

点云是一系列离散点的集合,点云密度越大,密集程度就越高,反之越低。实物目标上的点并不完全在点云上显示。在多次测量获取的点云数据中,实物目标上的一个固定点有可能在每一次测量数据上,也可能在某一次测量数据上,更可能在任何一次测量数据上都不存在。这样寻找到的对应点存在扫描位置引起的误差,给配准精度问题埋下了隐患。

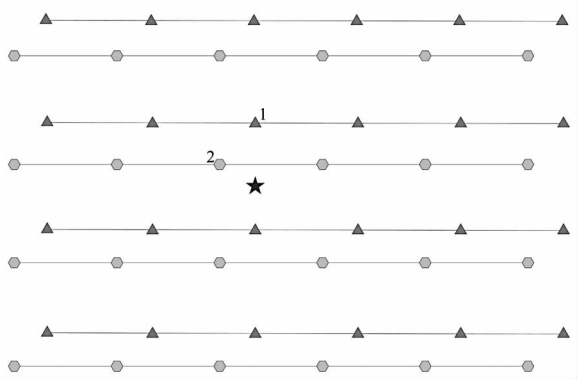


图 1　扫描错位

Fig. 1　Scanning dislocation

如图 1 所示,三角是第一次获取到的点云,多边形是第二次获取到的点云,五角星是实物目标上的固定点,点的相对位置与在实物目标上的相对位置相同。根据对应关系,可以找到五角星的对应点 1 与 2,但两次扫描都错开五角星,这样的对应点求出的旋转和平移矩阵精度必定达不到要求,存在较大误差。

造成上述问题的主要原因是三维测量数据点密度小,因此,要解决问题就要增加点云密度。迭代插值方法可解决上述问题。

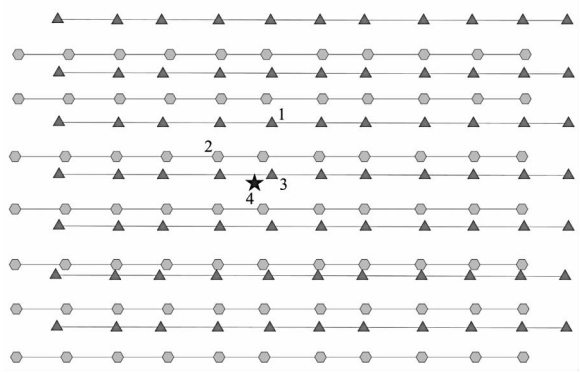


图 2　迭代插值

Fig. 2　Iterative interpolation

如图 2 所示,插值过后点密度增加,原先 1 与 2 是对应点,插值后 3 与 4 是对应点,两者之间的相对距离比较近,求得的旋转平移矩阵精度较前者高。若插值后密度没达到要求,可以继续迭代插值,直至达到密度要求。

解决方案:

1)分析数据结构,构建 KD-tree(K Dimensional tree),寻找每一个点 p_i 的 K 近邻点;

KD-tree 是一种分割 k 维数据空间的数据结构,主要应用于多维空间关键数据的搜索,如图 3 二维示例。树的根部节点是第一个指定的维度上均匀分开所有点的点,如 1 号点,如果在这一维度上小于根节点的点分在左边,大于根节点的点分在右面。树的每一级都在相应的一个维度上分开,这样所有的点之间存在枝与干的关系。建立 KD-tree 的最大作用是加快搜寻速度,寻找指定点最近点时根据根节点的维度比较大小确定下一步寻找左枝干还是右枝干,然后重复这个过程直到有一个枝干不再分支。例如,寻找图 3 中方块的最近点,首先比较主干 A 与方块 X 轴的大小,得到方块属于 A 的左枝干,重复这个过程得到最终结果方块的最近点在枝干 C 上。

2)建立 $M \times K$ 查询表矩阵,存储由 KD-tree 寻找到的每一个点的 K 近邻点索引值(M 是点云中点数);

3)遍历原点云每一个点 p_i,在 p_i 与 p_i 近邻点 q_i 之间建立插值点,在查询表矩阵中查询 q_i 和 p_i 是否已建立插值点,避免重复,减少计算量。

此处还可以构建三角网格,求取三角网格的

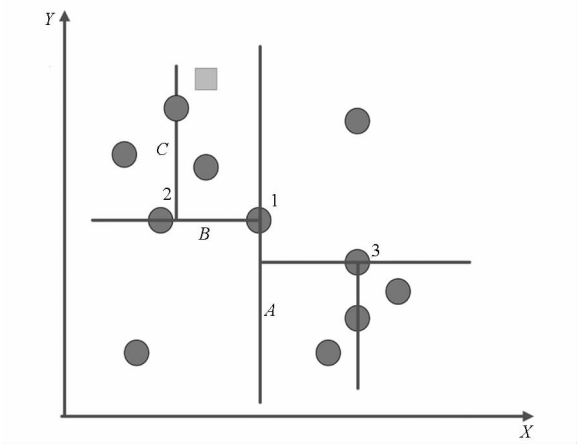


图 3　KD-tree

Fig. 3　KD-tree

中心插值。若经一次插值后点云密度仍偏小，可进行重复迭代插值，直到密度满足要求。

2　配准算法描述

2.1　概述

点云配准是两次测量点云数据之间进行整体数据融合，平移到同一个坐标系中。配准过程分为两个部分：粗配准过程和精配准过程。首先对原始点云去除离群点并进行插值增加点云密度，提高配准精确度。然后使用内在形状签名（Intrinsic Shape Signature，ISS）[14]算法求得点云数据的关键点，以减少计算量，加快计算速度，提高效率。配合 FPFH 算法获得关键点特征描述子，对两次测量获得的点云数据的关键点描述子比较，寻找对应点，并对所得的对应点使用随机采样一致性（Random Sample Consensus，RANSAC）[15]原理去除错误对应点。计算对应点协方差矩阵并对其进行奇异值分解（Singular Value Decomposition，SVD），得到旋转矩阵 $\boldsymbol{R}$、平移矩阵 $\boldsymbol{T}$。最后进行 ICP 精确配准，得到最终参数。配准原理如图 4 所示。

2.2　算法实现

1）输入点云，去除离群点。离群点是距离点云数据集较远的点，会导致计算复杂甚至计算错误。在点云数据中，用户指定的每个点周围一定范围内至少要有足够多的近邻点，没有足够近邻点的被认为是离群点而被删除。

2）点云数据迭代插值。使用本文提出方法对点云数据整体插值，提高点密度。

3）ISS 算法求关键点。关键点是点云上具有稳定性、区别性的点集，关键点的数量相比于原始点云数据量减少很多，能加速后续处理速度。本文使用 ISS 算法求得关键点：

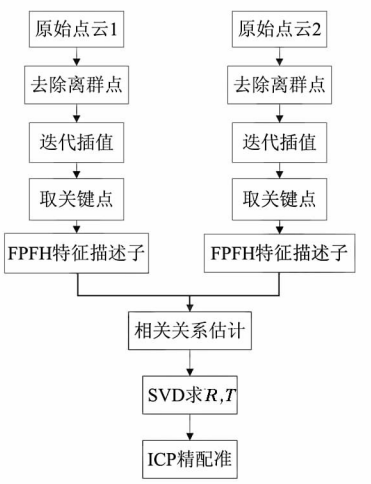


图 4　配准流程图

Fig. 4　Registration flow chart

① 查询一次点云获取得到的数据中每一个点 p_i 半径 r_{frame} 内所有点，并计算权值

$$w_{ij}=1/\|p_i-p_j\|,\ |p_i-p_j|<r_{frame} \tag{1}$$

② 根据权值计算方差矩阵

$$COV(p_i)=\frac{\sum\limits_{|p_i-p_j|<r_{frame}} w_{ij}(p_i-p_j)(p_i-p_j)^{\mathrm{T}}}{\sum\limits_{|p_i-p_j|<r_{frame}} w_{ij}} \tag{2}$$

③ 计算协方差矩阵的特征值 $\{\lambda_i^1,\lambda_i^2,\lambda_i^3\}$，特征值按照从大到小顺序排列；

④ 设置阈值 ε_1 和 ε_2 保留满足 $\lambda_i^2/\lambda_i^1\leqslant\varepsilon_1$ 和 $\lambda_i^3/\lambda_i^2\leqslant\varepsilon_2$ 的点为关键点。

4）获取 FPFH 特征描述子。点特征直方图（PFH）表示点的法线和其 k 邻域点法线之间的关系。如图 5（a）所示，PFH 特征就是计算任意一点 D_q 以及 D_q 的所有 k 近邻点（即与点 D_q 的距离小于半径 r 的所有点）之间两两相互连接的角度特征。

为了计算两点 D_s 和 D_t 及与其法线 n_s 和 n_t 之间的位置关系，在其中的一个点上根据法向量构建局部坐标系，如图 6 所示。

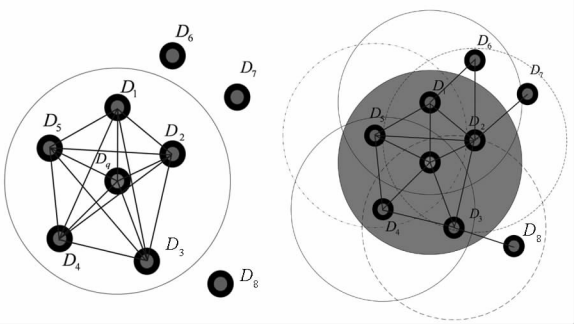
在图 6UVW 坐标系中，任意两点 D_s 和 D_t 及与其法线 n_s 和 n_t 之间特征如下表示：

$$\alpha=V\cdot n_t \tag{3}$$

$$\phi=U\cdot\frac{D_t-D_s}{d} \tag{4}$$

$$\theta=\arctan(W\cdot n_t,U\cdot n_t) \tag{5}$$

式（4）中 d 是 D_s 和 D_t 两点之间的欧氏距离，$d=\|D_t-D_s\|$。D_q 的所有 k 近邻点中任意两点之间的关系就可以用(α,ϕ,θ,d)四组值来表示。邻域内所有的四组值以统计的方式放进直方图中，就



(a)PFH 计算原理　　(b)FPFH 计算原理

图 5　PFH 和 FPFH 计算原理

Fig. 5　PFH and FPFH calculation principle

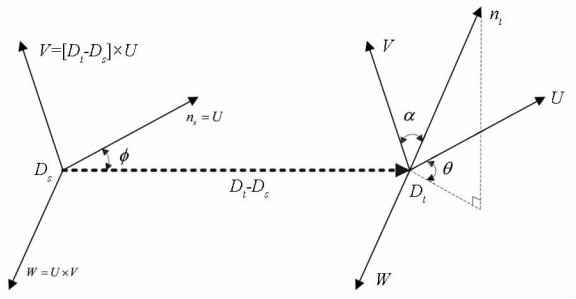


图 6　特征计算

Fig. 6　Calculation of feature

得到了 D_q 点的直方图特征。

FPFH 是由 PFH 改进而来。如图 5(b)所示,按照 PFH 计算方法计算 D_q 及其邻域内点(灰色区域)的直方图,这称之为简化的点特征直方图(Simple Point Feature Histograms,SPFH)。使用邻近的 SPFH 值来计算最终直方图(FPFH):

$$FPFH(D_q) = SPFH(D_q) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{w_k}\cdot SPFH(D_k) \tag{6}$$

式(6)中,以 D_q 和其邻近点 D_i 之间的距离作为权重 w_k。

这样很明显比 PFH 的标准计算少了邻域点之间的互联。只有一些重要对点(与 D_q 直接相连的点)被重复计数两次(灰色区域内)。

由图 5(b)以及式(6)可知 FPFH 特征计算与目标点邻域内点的数量有关。当关键点个数较少时,在关键点的邻域中的其他关键点也很少,这样按照其邻域内关键点计算 FPFH 特征就会丢失信息,必须依据关键点在原点云邻域计算特征向量,这就造成原点云中参与计算的点数量较大,计算速度降低。迭代插值算法使得点云密度增加,求得的关键点数量比较多,可依据关键点周围的其他关键点计算特征向量。这样相比于在原点云中的计算,可以减少点的计算量。

5)对应点估计。由上一步我们获得了一组点云数据关键点的 FPFH 特征,根据特征向量寻找这组点云上的对应点。

① 初步对应关系估计。寻找目标点云与源点云特征最相似的点对作为初步对应关系。

② 确定特征向量一一对应关系。初步对应关系中存在一对多的情况,选取距离最近的一个作为对应点,确定一一对应关系。

③ RANSAC 算法去除错误对应关系。

一对对应关系中的两个点,一个称为源点,另一个称为目标点。从已获得的对应关系样本集中随机地选择出一个大小为 n 的样本子集 S(在三维点云中 $n\geqslant 3$),并根据样本子集评估总样本集参数模型(在点云数据中参数模型是指源点与目标点之间的变换矩阵)。计算总样本集中的所有对应关系的源点经参数模型变换后与目标点的距离偏差,如果偏差小于设定的阈值,则此对应关系属于模型内样本,否则即为模型外样本,保存所有内样本。再次随机选择样本子集 S,重复上述过程。每一次重复都保存内样本,并统计内样本数量,直到迭代次数达到设定值,迭代结束。之后,找到内样本个数最大的参数模型,并令其为最终参数模型,认为此时的内样本为精确的对应关系。

6)求对应点协方差矩阵并进行奇异值分解,获取 $\boldsymbol{R}$(旋转矩阵),$\boldsymbol{T}$(平移矩阵)。求点云 M 和点云 N 的质心

$$C_m = \frac{1}{K}\sum_{i=1}^{K} m_i \tag{7}$$

$$C_n = \frac{1}{K}\sum_{i=1}^{K} n_i \tag{8}$$

构造矩阵

$$\boldsymbol{E}_{3x3}\frac{1}{K}\sum_{i=1}^{K}(m_i - C_m)(n_i - C_n)^{\mathrm{T}} \tag{9}$$

解 $\boldsymbol{E} = U\Lambda V^{\mathrm{T}}$,得 $\boldsymbol{X} = UV^{\mathrm{T}}$,即得:

$$\boldsymbol{R} = \boldsymbol{X},\ \boldsymbol{T} = C_n - \boldsymbol{R}C_m \tag{10}$$

7)ICP 精配准。根据曲率等特征确立对应点集 P 与 Q,其中对应点对的个数为 n。三维空间中两个对应点 p_i 和 q_j,它们的欧式距离表示为

$$d(p_i,\ q_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{11}$$

根据对应点由步骤 6 可找到 P 和 Q 变化的矩阵 $\boldsymbol{R}$ 和 $\boldsymbol{T}$,利用最小二乘法求解最优解,使:

$$e = \sum_{i=1}^{n}\left|(\boldsymbol{R}p_i + \boldsymbol{T}) - q_i\right|^2 \tag{12}$$

每次迭代都可以求出对应的 $\boldsymbol{R}$,$\boldsymbol{T}$ 和误差 e,直到达到迭代次数或者误差满足设定阈值。

3 配准实例与分析

本文采用 Bunny 和 Happy_backdrop 三维点云数据在 visual studio 2010 环境下结合进行仿真实验,对比点云插值与未插值配准结果。两组点云的数据量分别在 4 万和 8 万左右,并且重合区域大于 50%。

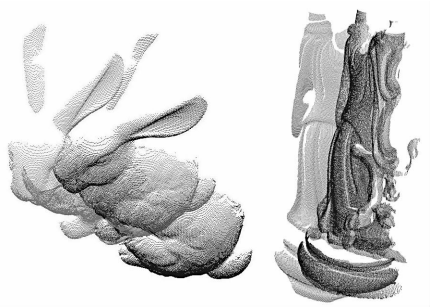


图 7 两组原始点云

Fig. 7 Two sets of the original point cloud

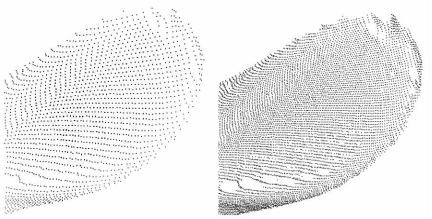


图 8 Bunny 数据插值对比

Fig. 8 Comparison of Bunny data with and without interpolation

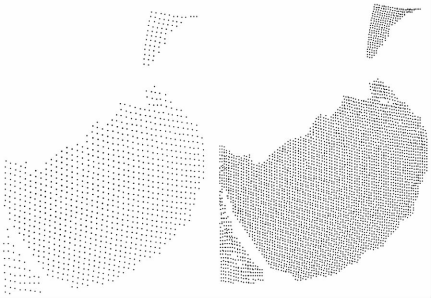
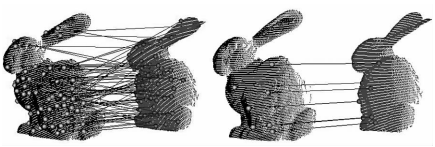


图 9 Happy_backdrop 数据插值对比

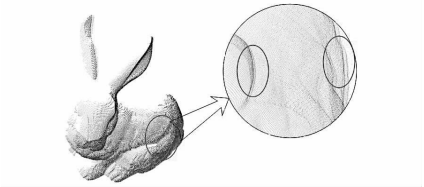
Fig. 9 Comparison of Happy_backdrop data with and without interpolation



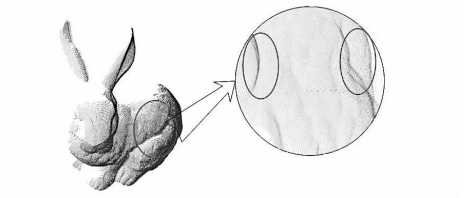
(a) (b)

图 10 RANSAC 算法去除错误对应关系

Fig. 10 RANSAC algorithm to remove the false correspondence



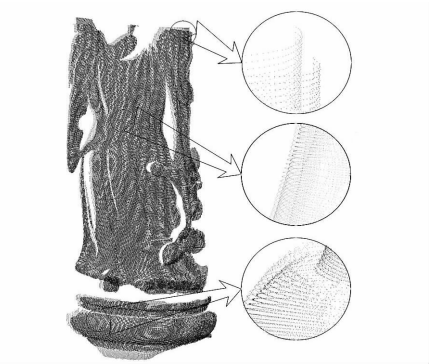
(a)未插值配准结果



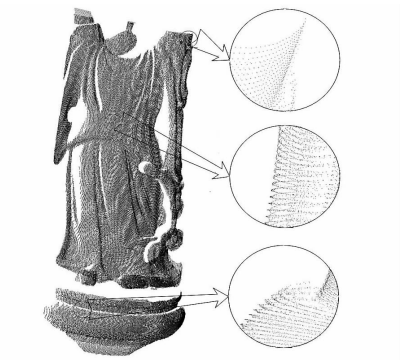
(b)插值配准结果

图 11 Bunny 数据配准结果对比

Fig. 11 Comparison of registration results in Bunny data



(a) 未插值配准结果



(b) 插值配准结果

图 12 Happy_backdrop 数据配准结果对比

Fig. 12 Comparison of registration results in Happy_backdrop data

表 1 Bunny 数据配准结果参数对比

Tab. 1 Comparison of registration result parameters in Bunny data

点云	未插值	插值
点云数据	$M=40\,097$ $N=40\,256$	$M=232\,331$ $N=233\,398$
关键点数	$R=686$ $S=746$	$R=1414$(插值 416) $S=1433$(插值 407)
对应点对	15	25
配准评估	6.845 68E − 6	4.398 34E − 6
旋转矩阵	$\begin{bmatrix} 0.9999 & -0.0026 & -0.0066 \\ 0.0026 & 0.9999 & -0.0008 \\ 0.0066 & 0.0008 & 0.9999 \end{bmatrix}$	$\begin{bmatrix} 0.9998 & -0.0021 & -0.0151 \\ 0.0021 & 0.9999 & -0.0028 \\ 0.0151 & 0.0028 & 0.9998 \end{bmatrix}$
平移矩阵	[0.0005 −0.0001 0]	[0.0009 0.0001 0.0003]
时间/s	52.453	40.234

表 2　Happy_backdrop 数据配准结果参数对比

Tab. 2　Happy_backdrop data registration results

点云	未插值	插值
点云数据	$M=79\,801$ $N=76\,983$	$M=447\,016$ $N=462\,086$
关键点数	$R=3150$ $S=3234$	$R=4936$(插值 632) $S=4904$(插值 616)
对应点对	59	92
配准评估	9.515 51E－7	4.635 13E－7
旋转矩阵	$\begin{bmatrix} 0.9999 & -0.0015 & -0.0059 \\ 0.0015 & 0.9999 & -0.0019 \\ 0.0059 & 0.0019 & 0.9999 \end{bmatrix}$	$\begin{bmatrix} 0.9999 & 0.0002 & 0.0096 \\ -0.0003 & 0.9999 & -0.0041 \\ -0.0095 & 0.0041 & 0.9999 \end{bmatrix}$
平移矩阵	[0.0014　0.0004　−0.0005]	[−0.0017　0.0007　−0.0008]
时间/s	60.906	52.688

图 7 是两组点云原始图像。表 1 和表 2 分别是两组数据配准结果参数对比。同一组数据中,M 表示第一幅点云数据点数,N 表示第二幅点云数据点数;R 表示第一幅点云数据关键点数,S 表示第二幅点云数据关键点数。两组数据插值后数据量约为插值前的 5.6 倍,极大地增加了点云的个体密度。由于插值所得点数据是临近两点的平均值,所以保证了点云局部细节化且不会增加误差。插值过程中使用了 KD-tree 数据结构,查询 K 近邻点时循着主干和枝干经过少量比较即可找到所需要的点,极大减少了紧邻查询数量,提高计算效率。图 8 和图 9 是两组数据插值前后对比。由表 1 和表 2 可得出迭代插值后寻找的关键点数要多,并且有一部分关键点是经过插值得来的。迭代插值配准算法所求得的关键点数量增加,避免了在原点云中计算关键点 FPFH 特征,减少了计算量。表 1 和表 2 显示插值算法比未插值算法时间上提高了 10s 左右,计算速率稍有提高。迭代插值后根据关键点处 FPFH 特征向量相似度得到的对应点数量有所增加,证明迭代插值方法发挥了作用。图 10 中展示了错误对应关系去除前后情况的对比,图 10 (a)中的对应关系是通过关键点特征向量欧氏距离最小选取的,得到 84 对对应点,其中存在大量的错误对应关系。经过 RANSAC 算法后保存了 7 对对应关系,如图 10 (b)所示,可以看出获得了正确的对应关系。图 11 所示的 Bunny 数据配准结果中,图 11(a)从右侧圆圈内的两个小圆圈可以看出两次扫描获得的点云数据没有重合,存在明显的距离;相对于图 11(b),右侧圆圈内的两个小圆圈内数据不存在距离。在图 12(a)中雕塑的肩部、背部、底座存在明显的配准误差,在图 12(b)中相应部位误差均得到了提高。从表 1 和表 2 两组点云数据配准评估中也可以看出迭代插值方法能提高点云配准的精度。

4　结论

迭代插值方法填补了点与点之间的空缺,增加了点云密度。插值过程中使用了 KD-tree 数据结构,加快了插值速度,提高了插值的准确性。使用 ISS 算法获得了具有稳定性和可区别性的关键点集合。获取关键点这种方法减少了后续计算所使用的点云数据量,提高了计算效率。结合 FPFH 特征描述子方法寻找对应点,使用了随机采样一致性算法去除错误的对应关系。迭代差值配准方法使关键点数量增加,减小了由点云获取设备分辨率较低带来的实际对应关系的偏差对匹配结果带来的影响,提高了点云配准的精度。

参考文献(References)

[1] Rusu R B, Marton Z C, Blodow N, et al. Persistent point feature histograms for 3D point clouds[C]//Proceedings of the 10th International Conference on Intelligent Autonomous Systems, 2008: 119－128.

[2] Rusu R B, Blodow N, Marton Z C, et al. Aligning point cloud views using persistent feature histograms[C]//Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, 2008: 3384－3391.

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

[4] Rusu R B, Holzbach A, Blodow N, et al. Fast geometric point labeling using conditional random fields[C]//Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, 2009: 7－12.

[5] Gelfand N. Feature analysis and registration of scanned surface[D]. America, Stanford University, 2006.

[6] Johnson A E, Hebert M. Surface matching for object recognition in complex 3D scenes[J]. Image and Vision Computing, 1998, 16(9－10): 635－651.

[7] Rusu R B, Bradski G, Thibaux R, et al. Fast 3D recognition and pose using the viewpoint feature histogram[C]// Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 2010: 2155－2162.

[8] Magnusson M. The three dimensional normal distributions transform-an efficient representation for registration, surface analysis and loop detection[D]. Örebro: Örebro University, 2009.

[9] Besl P J, Mckay N D. A method for registration of 3－d shapes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14(2): 239－256.

[10] Turk G, Levoy M. Zippered polygon meshes from range images[C]//Proceedings of SIGGRAPH, 1994: 311－318.

[11] Fan K C, Tsai T H. Optimal shape error analysis of the matching image for a free-form surface[J]. Robotic and Computer-Integrated Manufacturing, 2001, 17(3): 215－222.

[12] Li Q. Griffiths J G. Iterative closest geometric objects registration[J]. Computers and Mathematics with Applications, 2000, 40(10－11): 1171－1188.

[13] Johnson A E, Kang S B. Registration and integration of textured 3D data[J]. Image and Vision Computing, 1999, 17: 135－1471

[14] Zhong Y. Intrinsic shape signatures: a shape descriptor for 3D object recognition[C]//Proceedings of 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), Kyoto, Japan, 2009: 689－696.

[15] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[C]//Readings in Computer Vision Issues, Problem, Principles, and Paradigms, 1987: 726－740.