

面向低维点集配准的高效最近邻搜索法*

祝继华 尹俊 邢汶铎 杜少毅

(西安交通大学 软件学院 西安 710049)

摘要 为提高点集配准效率,设计一种适用于二维/三维点集的高效最近邻搜索法.该方法根据由模型点集的各维方差所选定的维度信息,排序模型点集中的点.借助二分查找法,将数据点集中的每个点插入至排序后的模型点集中,并利用左边第一个点确定搜索范围的上确界.当在确定范围内搜索最近邻时,可根据当前结果进一步减小待搜索范围,以便快速获得各点的最近邻.最后进行的复杂度分析和实验对比结果均验证文中方法的有效性.

关键词 欧氏距离,最近邻搜索,上确界,点集配准,迭代最近点法

中图法分类号 TP 181

Efficient Nearest Neighbor Search Approach for Registration of Low Dimensional Point Sets

ZHU Ji-Hua, YIN Jun, HAN Wen-Xin, DU Shao-Yi

(Software Engineering School, Xi'an Jiaotong University, Xi'an 710049)

ABSTRACT

To improve the efficiency of point set registration, an efficient nearest neighbor search approach for 2D/3D point sets is proposed. Firstly, according to the variance of each dimension of the model points, all model points based on the selected dimensional data are sorted. By adopting the binary search strategy, each data point is inserted into the sorted model points. Then, the upper bound of search range can be obtained by calculating the distance between the data point and its first left model point. During the search process, the search range can be further reduced by the current nearest neighbor so that the final nearest neighbor can be efficiently searched. Finally, the efficiency of the approach is demonstrated by both the complexity analysis and experimental results.

Key Words Euclidean Distance, Nearest Neighbor Search, Upper Bound, Point Set Registration, Iterative Closest Point Algorithm

* 国家自然科学基金项目 (No. 61203326)、中国博士后科学基金项目 (No. 2012M512004)、陕西省自然科学基金项目 (No. 2014JM8342) 资助

收稿日期:2013-11-08;修回日期:2013-12-24

作者简介 祝继华(通讯作者),男,1982年生,博士,讲师,主要研究方向为计算机视觉、移动机器人. E-mail: zhujh@mail.xjtu.edu.cn. 尹俊,女,1988年生,硕士研究生,主要研究方向为三维重建. 邢汶铎,男,1989年生,硕士研究生,主要研究方向为三维重建、移动机器人. 杜少毅,男,1980年生,博士,副教授,主要研究方向为模式识别、图像处理.

1 引言

最近邻搜索是计算机视觉、模式识别等研究领域的基础问题. 给定两个同等维度的向量集, 该问题可描述为: 针对一个向量集中的每个向量, 基于所定义的度量距离, 从另外一个向量集中搜索获得距离最近的向量. 为解决该问题, 可采用顺序扫描法, 即将模型向量集中的每个向量与数据向量集中的待查询向量逐一进行距离对比, 以获得最近邻. 此方法虽然简单, 但效率较低.

为提高搜索效率, 研究者提出基于三角剖分的最近邻搜索法^[1-2]. 该方法将模型向量集中的所有向量转化成三角网格的数据结构, 并针对数据向量集中的每个向量, 从三角网络中搜索获得最近邻. 此方法虽能提高最近邻搜索的效率, 但数据结构较复杂.

另一类有效的搜索方法是基于 k-d 树^[3]的最近邻搜索法, 该方法采用相对简单的二叉树结构, 并利用深度优先和回溯搜索两步获得最近邻. 但由于存在回溯搜索过程, 故效率有待提高. 基于此方法, Greenspan 等^[4]提出近似最近邻搜索法, 它是一种以牺牲精度而提高搜索效率的方法, 是目前效率最高的方法, 但它获得的并非是最近邻. 此外, 研究者们还提出一些改进的基于 k-d 树的最近邻搜索法^[5-6]. 虽然基于 k-d 树的最近邻搜索法及其变异方法的效率较高, 但其效率与向量维数的大小成反比. 随着向量维数的不断增加, 此类方法的效率逐渐下降.

为解决高维向量的最近邻搜索问题, Slaney 等^[7]提出基于局部敏感哈希算法 (Locality-Sensitive Hashing, LSH) 的最近邻搜索法, 该方法利用哈希函数将输入的高维特征矢量散列至低维特征空间中, 并保证在原始空间中距离较近的点经过散列后在低维空间中依然距离较近, 以提高最近邻搜索法的效率. 此外, Ahn 等^[8]提出一种将高维数据嵌入至低维非线性子空间的策略, 并采用有效方法快速确定搜索范围的上确界, 以高效搜索获得最近邻. 上述两类方法适合于高维向量的最近邻搜索, 但当它们应用于低维向量时, 其效率并不高.

本文针对点集配准问题, 提出面向低维数据的高效最近邻搜索法. 该方法采用排序和二分查找法快速确定搜索范围的上确界, 然后进行最近邻搜索, 并在搜索过程中逐步缩小搜索边界, 以便高效地搜索获得最近邻.

2 点集配准问题及迭代最近点算法

点集配准是模式识别^[10]、计算机视觉^[11]与移动机器人^[12]等领域的关键技术之一. 当前最常用的点集配准方法是用于解决刚体配准问题的迭代最近点算法 (Iterative Closest Point, ICP), 它具有精度高、收敛速度快的优点.

基于此算法, 各种变异算法^[13-16]被提出, 解决不同类型的配准问题. 给定 \mathbf{R}^d 空间下两个完全重叠的形状点集: 数据点集

$$\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^{N_p}, N_p \in \mathbf{N}$$

和模型点集

$$\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^{N_q}, N_q \in \mathbf{N},$$

刚体配准的目标是计算一个最优的刚体变换 $\varphi = (\mathbf{R}, \mathbf{t})$, 使得经过变换后的数据点集 $\mathbf{T} = (\mathbf{P}, \varphi)$ 能较好地与模型点集匹配在一起. 基于平方和误差最小的原则, 此类问题可描述为

$$\min_{\mathbf{R}, \mathbf{t}, j \in \{1, 2, \dots, N_q\}} \sum_{i=1}^{N_p} \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_j\|_2^2,$$

$$\text{s. t. } \mathbf{R}^T \mathbf{R} = \mathbf{I}_{d \times d}, \det(\mathbf{R}) = 1,$$

其中, $\mathbf{R} \in \mathbf{R}^{d \times d}$ 为旋转矩阵, $\mathbf{t} \in \mathbf{R}^d$ 为平移变量. 目前, 解决上式最有效的方法是 ICP. 该算法在给定合适的变换初值 $(\mathbf{R}_0, \mathbf{t}_0)$ 的前提下, 采用迭代模式计算最优的变换解, 且每次迭代过程均包含如下 2 步.

1) 根据前一次迭代得到的刚体变换 $(\mathbf{R}_{k-1}, \mathbf{t}_{k-1})$, 对数据点集中的每个点 \mathbf{p}_i , 计算其最近邻点 $\mathbf{q}_{c(i)}$:

$$c_k(i) = \arg \min_{j \in \{1, 2, \dots, N_q\}} \|\mathbf{R}_{k-1}\mathbf{p}_i + \mathbf{t}_{k-1} - \mathbf{q}_j\|_2.$$

2) 根据当前建立点对关系 $(i, c_k(i))$, 求解最新的刚体变换^[17]结果:

$$(\mathbf{R}_k, \mathbf{t}_k) = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^{N_p} \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_{c_k(i)}\|_2^2.$$

重复执行上述两步, 直至满足算法收敛条件, 即可获得刚体配准结果.

由于许多点集配准算法是 ICP 的变种, 因此这些算法的效率均取决于 ICP 的效率. 根据相关文献可知, ICP 最耗时的操作为最近邻搜索, 即基于欧氏距离, 建立两点集之间点的最新对应关系 $(i, c_k(i))$.

为此针对点集配准问题, 本文给出一种高效的最近邻搜索法, 提高 ICP 及变异算法的执行效率.

3 基于单维排序的最近邻搜索方法

本节将介绍基于单维排序的最近邻搜索法的原理和具体实现,并给出详细的计算复杂度的分析结果.

3.1 搜索方法的原理及实现

在 ICP 的每次迭代中,数据点集中的每点均需从模型点集 Q 中搜索获得其最近邻点.如图 1 所示, $Q = \{q_j\}_{j=1}^{N_q}$ 是模型点集, p_i 是数据点集 P 中的一个点.

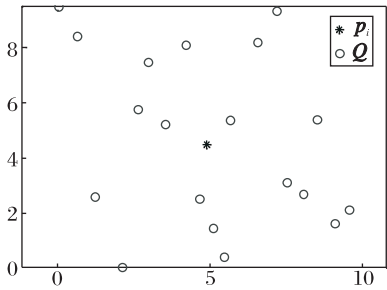


图 1 待搜索点与模型点集的示意图

Fig. 1 Sketch map of point to be matched and all points in model shape

为从 Q 中快速搜索获得与点 p_i 的欧氏距离最近的点,可先分析点 p_i 与点 q_j 之间欧氏距离 L_{ij} 的定义:

$$L_{ij} = \begin{cases} \sqrt{(p_i^x - q_j^x)^2 + (p_i^y - q_j^y)^2}, & d = 2 \\ \sqrt{(p_i^x - q_j^x)^2 + (p_i^y - q_j^y)^2 + (p_i^z - q_j^z)^2}, & d = 3 \end{cases} \quad (1)$$

如果模型点集中的所有点已按 X (也可以是 Y 或 Z) 轴的坐标值排序,且已知点 q_g 满足

$$q_g = \arg \min_{j \in \{1, 2, \dots, N_q\}} (p_i^x - q_j^x)^2, \quad (2)$$

则称该点为点 p_i 的相邻点,即 q_g 是模型点集 Q 的所有点中与待搜索点 p_i 在 X 轴坐标值差异最小的点.由式(1)可知,对于二维/三维点集,如两点之间的距离 L_{ij} 较小,则等式右边的项 $(p_i^x - q_j^x)^2$ 小的概率较小.因此可在点 q_g 附近搜索获得 p_i 的最近邻点 $q_{c(i)}$.实际上,选择不同轴的坐标值排序及搜索,对应搜索法的效率将会有所差异,后续内容将分析具体原因及所依据的选择标准.为便于叙述,下文选用 X 轴的坐标值排序及搜索.

由此可提出基于单维排序的最近邻搜索法,该

方法的思想如下:在预处理阶段,将模型点集中的所有点按 X 轴的坐标值排序;在搜索最近邻时,根据待搜索点 p_i 在 X 轴的坐标值将其插入到经过排序后的模型点集中,并在插入点附近搜索其最近邻点.该搜索法可分为 6 步,图 2 给出该方法分步实现的示意图,该图中的每个子图分别对应于搜索过程中的每一步.

step 1 利用高效的排序法,将模型点集中的点按 X 轴的坐标值升序排序,得到排序后的模型点集 Q' .

step 2 根据二分查找法将数据点集中的待搜索点 p_i 插入到经过排序的模型点集 Q' 中.设 $q_{c(i)}$ 是点 p_i 的最近邻点,则点 $q_{c(i)}$ 是包含大量点的模型点集 Q 中与点 p_i 的欧氏距离最小的点,故点 $q_{c(i)}$ 和 p_i 之间的欧氏距离 $L_{ic(i)}$ 非常小.对于二维/三维点集中,如两点的欧氏距离 $L_{ic(i)}$ 非常小,则 p_i 和 $q_{c(i)}$ 之间 X 轴的坐标值的差值较小,即 $(p_i^x - q_{c(i)}^x)^2$ 较小.借助上述搜索策略,基于单维排序的搜索法可快速地将搜索位置定位在最近邻 $q_{c(i)}$ 附近.

step 3 计算点 p_i 和它左边第一个点 q_g 之间的欧氏距离 L ,并将此距离作为搜索范围的上确界.由于点 p_i 和点 q_g 之间满足式(2),对低维数据,这两点之间距离小的概率非常大.因此借助于单维排序和二分查找法,可快速确定待搜索范围的有效上确界.另外根据欧氏距离的定义式可知,仅当模型点集中的点 q_j 与数据点集中的待搜索点 p_i 之间满足

$$abs(p_i^x - q_j^x) < L$$

时,它们之间的欧氏距离才可能小于搜索范围的上确界 L .由此可知,只需在

$$p_i^x - L < q_j^x < p_i^x + L$$

范围内搜索点 p_i 的最近邻点.

step 4 前向搜索.在 $L \geq abs(\Delta x_{ij})$ 的范围内向前搜索点 p_i 的最近邻点.在前向搜索过程中,需在

$$p_i^x - L < q_j^x < q_g^x$$

范围内,从右到左地遍历 Q' 中的各个排序点,同时计算这些点与目标点 p_i 的欧氏距离,并记录下当前最小的欧氏距离 $L_{ic(i)}$ 及对应的点 $q_{c(i)}$.在搜索过程中,可根据最新的搜索结果更新搜索范围的上确界 $L = L_{ic(i)}$,以便缩小搜索范围.

step 5 后向搜索.待前向搜索结束后,返回点 q_g ,并在 $L \geq abs(\Delta x_{ij})$ 的范围内向后搜索点 p_i 的最近邻点.在后向搜索过程中,需在

$$q_g^x < q_j^x < p_i^x + L$$

范围内,从左到右地遍历 Q' 中的各个排序点,同时

计算这些点与目标点 p_i 的欧氏距离. 同理在搜索过程中, 如搜索到最新的最近邻点, 则可进一步缩小待搜索范围.

step 6 待双向搜索结束后, 即可获得 p_i 在模型点集中的最近邻点 $q_{c(i)}$.

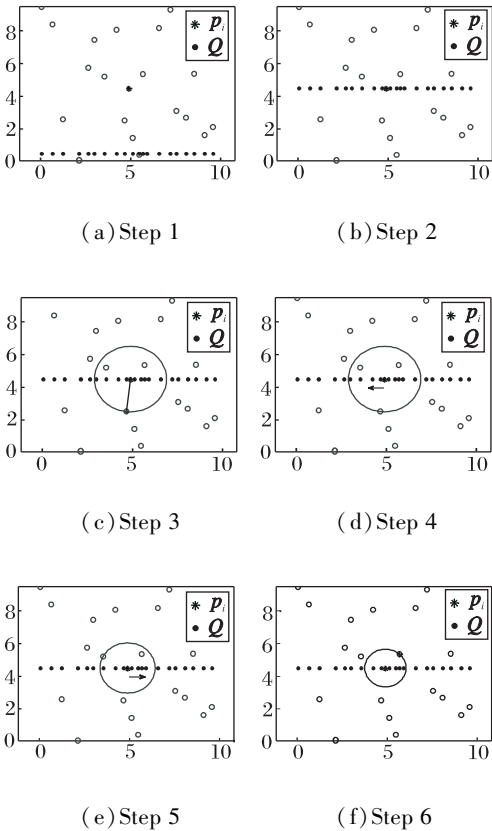


图2 本文搜索法的示意图

Fig. 2 Search process of the proposed search approach

根据式(2)可知, 在 step 3 中, 有可能点 p_i 右边的第一个点才是点 q_g , 但这并不会影响最终的最近邻搜索结果. 另外对于点 p_i , 如果模型点集中同时存在多个最近邻点, 则取其首次搜索获得的最近邻点作为结果输出. 根据上述叙述可知, 针对每个待搜索点, 上述算法可从已排序的模型点集中找到一个点, 该点与待搜索点在排序轴上的坐标值差的绝对值最小, 它与待搜索点之间的欧氏距离可作为最近邻搜索的上确界. 同时为加快最近邻的搜索速度, 上确界需根据搜索中间结果不断调整. 最后当算法搜索到待搜索点的最近邻时, 上确界更新为目标点和最近邻点之间欧氏距离.

在实际应用中, 点集中的点在各个维度的分布规律是不同的, 它们在方差大的维度上较分散. 如选取不同维度的信息排序及搜索时, 则最近邻搜索法

的效率也会发生相应变化. 假设模型点集在 X 轴分布较集中, 而在 Y 轴较分散, 则对于固定的搜索范围上确界, 沿 X 轴搜索所需遍历的点将多于沿 Y 轴搜索所需遍历的点. 为此实际应用中可根据点集在各个维度上的方差, 选取最大方差值所对应的维度排序及搜索, 从而提高最近邻搜索法的效率.

3.2 复杂度分析

根据上述描述可知, 本文方法先要对模型点集中的所有点按 X 轴的坐标值排序, 由此引入的计算复杂度为 $O(N_q \log_2 N_q)$. 为将点 p_i 插入到经过排序后的模型点集中 Q' 中, 可采用二分查找法, 其对应的计算复杂度为 $O(\log_2 N_q)$. 待将目标点 p_i 插入到经过排序后的模型点集 Q' 中后, 即可利用该点左边的第一个点确定待搜索范围的上确界 L . 假设在搜索过程中, 所需遍历的点与模型点集中点的总数之比为 ε , 则一次搜索过程为 $O(\varepsilon N_q)$. 由于相邻点在选定的维度上与目标点的坐标值差异最小, 根据欧氏距离的定义可知, 对于二维 / 三维数据, 相邻点与目标点之间欧氏距离小的概率较大. 故利用相邻点所确定的搜索范围较小, 且在搜索过程中, 还可根据最新的搜索结果进一步缩小搜索范围. 因此本文方法需遍历的点数较少, 即 ε 值较小. 实验部分将给出 ε 的统计值.

ICP 只需在迭代前对模型点集进行一次排序, 因此排序过程的计算复杂度为 $O(N_p \log_2 N_q)$. 由于数据点集包含 N_p 个点, 这些点均需从经过排序后的模型点集 Q' 中搜索获得最近邻点, 故插入排序点的总计算复杂度为 $O(N_p \log_2 N_q)$, 在插入点附近搜索最近邻点的总计算复杂度为 $O(\varepsilon N_p N_q)$.

表 1 列出本文方法的各操作过程的计算复杂度, 为便于对比, 该表中还给出基于三角剖分和基于 k-d 树的最近邻搜索法中各个步骤的计算复杂度, 其中 d 表示数据维数.

表 1 3 种搜索方法的计算复杂度分析结果

Table 1 Analysis of computational complexity of 3 search approaches		
方法	操作步骤	复杂度
三角剖分	创建三角网络	$O(N_q \log_2 N_q + N_q^{\lceil d/2 \rceil})$
	搜索最近邻点	$O(N_p \log_2 N_q) \leq O(\cdot) \leq O(N_p N_q)$
k-d 树	创建二叉树	$O(d N_q \log_2 N_q)$
	深度优先查找回溯	$O(N_p \log_2 N_q) \leq O(d N_p N_q^{(1-1/d)})$
单维排序	单维排序	$O(N_q \log_2 N_q)$
	插入顺序点搜索最近邻点	$O(N_p \log_2 N_q) \leq O(\varepsilon N_p N_q)$

4 实验及结果分析

为验证本文方法的有效性,本节将利用二维形状和三维深度数据集进行效率测试,并与基于三角剖分的最近邻搜索法和基于 k-d 树的最近邻搜索法对比. 本文方法采用 C++/Matlab 混合编程实现,即用 C++ 实现最近邻搜索法,并生成动态连接库供 Matlab 调用. 其他 2 种搜索方法均直接调用 Matlab 命令,其底层实现均采用 C/C++ 语言. 所有实验均运行在一台双核主频 2.7 GHz,内存为 2.91 GB 的笔记本电脑上.

实验随机地从公开数据集中选择 3 组二维形状数据和 3 组三维深度数据. 这些形状数据的模型点集与数据点集之间均存在一个最优的刚体变换. 为实现两者之间匹配,可采用 ICP 进行刚体配准. 图 3 (浅色和深色形状分别代表模型和数据点集) 给出所选择的二维形状数据及 ICP 的配准结果,图 4 (浅色和深色形状分别代表模型和数据点集) 给出所选择的三维深度数据以及 ICP 的配准结果.

为验证本文方法的合理性,可分别根据不同轴的信息排序模型点集,以获得相应的最近邻搜索策略,并应用于 ICP 中. 实验计算模型数据在各维度上

方差,并分别记录不同搜索策略的程序预处理和搜索耗时、方差最大的轴和表 1 中参数 ε 的统计值. 为消除随机性,每组数据均进行 100 次的独立测试,并将获得的平均耗时记录在表 2 中.

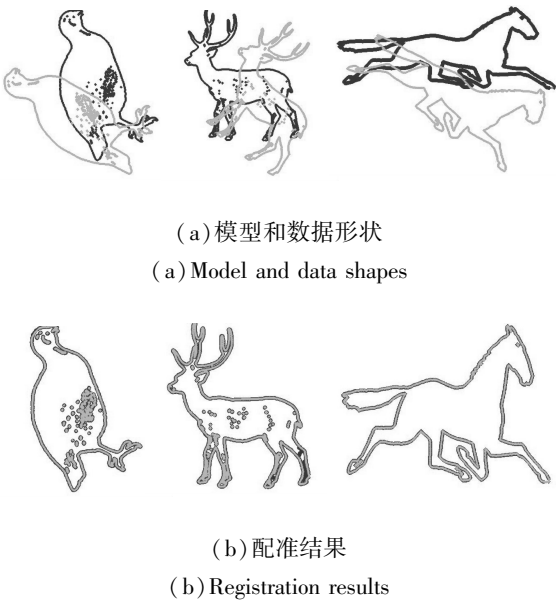


图 3 二维形状及 ICP 算法的配准结果

Fig. 3 2D shapes and the corresponding registration results

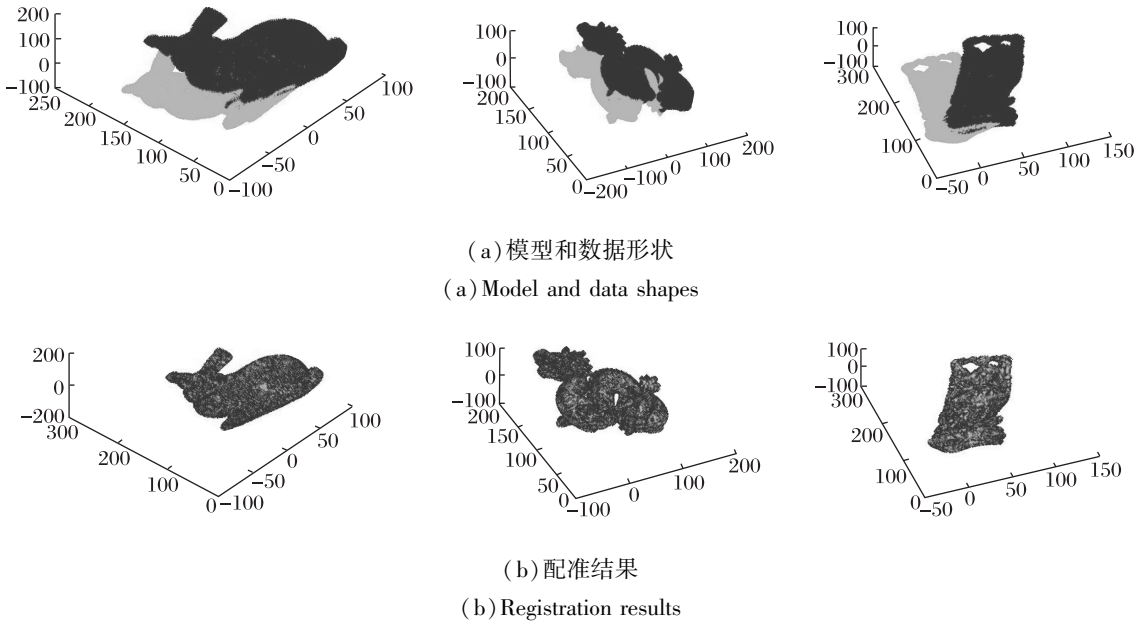


图 4 三维形状及 ICP 算法的配准结果

Fig. 4 3D shapes and the corresponding registration results

由表 2 可知,虽采用不同搜索策略的预处理耗时差别较小,但基于方差最大轴搜索策略的耗时最小. 因此通过选择方差最大轴的信息排序模型数据,可提高本文方法的搜索效率. 另外从表 2 中还可看

出,如选择方差最大的轴排序和搜索,则 ε 的统计值接近于 0.03,即在模型点集 Q 中搜索点 p_i 的最近邻时,所需遍历的点数较少,从而进一步验证本文方法的有效性.

为对比各种最近邻搜索法的效率,实验记录 ICP 程序执行过程中各种最近邻搜索法的程序耗时.为消除随机性,每组数据均进行 100 次独立测试,并将获得的平均耗时记录在表 3 中.

为便于对比,表 3 中还给出各模型点集和数据

点集的点数、ICP 的迭代次数及采用顺序扫描法搜索最近邻点的平均耗时.同时为分析不同搜索法的精度,表 4 还记录采用不同最近邻搜索法所获得的 ICP 配准结果的均方根误差 (Root Mean Square Error, RMS).

表 2 基于不同排序轴的搜索效率对比

Table 2 Comparison of search efficiency under different sorted axes

点集名称	方差 (最大轴)	X 轴/s			Y 轴/s			Z 轴/s		
		预处理	搜索	ε	预处理	搜索	ε	预处理	搜索	ε
Bird2D	Y	0.0005	0.1222	0.0550	0.0003	0.0912	0.0330	—	—	—
Deer2D	X	0.0008	0.2060	0.0348	0.0004	0.2750	0.0549	—	—	—
Horse2D	Y	0.0006	0.1051	0.0424	0.0003	0.0769	0.0299	—	—	—
Bunny3D	Y	0.0031	0.6899	0.0406	0.0034	0.5962	0.0349	0.0035	0.9128	0.0522
Happy3D	Y	0.0058	4.1122	0.0506	0.0055	2.5010	0.0295	0.0054	5.4346	0.0727
Dragon3D	X	0.0084	2.8874	0.0288	0.0085	3.4947	0.0349	0.0086	5.1958	0.0566

表 3 不同搜索方法在 ICP 中的效率对比

Table 3 Efficiency comparison of different search approaches in ICP algorithm

点集名称	N_q	N_p	ICP 迭代次数	顺序扫描/s	三角剖分/s		k-d 树/s		单维排序/s	
					预处理	搜索	预处理	搜索	预处理	搜索
Bird2D	3260	3034	39	6.4420	0.0738	0.4465	0.0071	0.2399	0.0003	0.0912
Deer2D	5312	5252	37	15.0648	0.1158	1.0923	0.0085	0.4531	0.0008	0.2060
Horse2D	3422	3586	31	6.3468	0.0622	0.3738	0.0063	0.2241	0.0003	0.0769
Bunny3D	8987	8171	41	41.3182	0.6932	5.4129	0.0155	0.9755	0.0034	0.5962
Happy3D	15102	24108	45	229.8463	1.1579	12.1759	0.0334	4.1588	0.0055	2.5010
Dragon3D	20841	20005	49	285.0948	1.6879	15.8766	0.0532	5.0984	0.0084	2.8874

在表 3 中,基于三角剖分的搜索法的预处理对应为建立模型点集的三角网格;基于 k-d 树搜索法的预处理对应为建立模型点集的二叉树数据结构;本文方法的预处理对应为计算模型点集的方差,并根据方差按选定的维度排序模型点集.需说明的是,在 ICP 中,各搜索方法的预处理只需执行一次,而搜索过程则需在每次迭代中执行.

表 3 中给出的结果是各种搜索方法的全部迭代中的平均总耗时.从表 3 中可看出,如果采用顺序扫描法搜索最近邻点,则程序的耗时太长.而采用后 3 种搜索法则可大幅提升程序的效率.在后 3 种搜索法中,无论是预处理过程还是搜索过程,基于三角剖分的最近邻搜索法均耗时最多,其次是基于 k-d 树的最近邻搜索法.而本文方法的耗时最少、效率最高.与其他 2 种方法相比,本文方法对二维形状数据的搜索效率提升较明显,而对于三维深度数据搜索效率提升有所下降.这是当数据维数较低时,如果点 p_i 和点 q_j 之间 X 轴的坐标值的差值 $(p_i^x - q_j^x)^2$ 较小,则对应的欧氏距离 L_{ij} 较小的概率非常大,故利用二

分查找法所确定的搜索范围较小,搜索最近邻点时所需遍历的点非常少.随着数据维数的升高,在 $(p_i^x - q_j^x)^2$ 不变的前提下,其对应的欧氏距离 L_{ij} 较小的概率将逐渐下降,在搜索过程中所需遍历的点数将逐渐增加,故搜索过程的计算复杂度变大,从而导致本文方法的效率有所下降.

从表 4 中可看出,ICP 采用不同的最近邻搜索法所获得的配准精度是相同的,原因是这些最近邻搜索法均属于精确的搜索方法.

表 4 不同搜索方法在 ICP 中的精度对比

Table 4 Precision comparison of different search methods in ICP algorithm

点集名称	顺序扫描	三角剖分	k-d 树	单维排序
Bird2D	0.0569	0.0569	0.0569	0.0569
Deer2D	0.0604	0.0604	0.0604	0.0604
Horse2D	0.0621	0.0621	0.0621	0.0621
Bunny3D	1.2481	1.2481	1.2481	1.2481
Happy3D	1.1780	1.1780	1.1780	1.1780
Dragon3D	1.0537	1.0537	1.0537	1.0537

由上述实验对比结果可知,本文方法可获得精确的最近邻搜索结果,且对于二维/三维数据,其搜索效率高于以往的最近邻搜索法.因此该方法适用于一般的点集配准算法.

5 结 束 语

针对点集配准中的最近邻搜索问题,本文提出一种基于单维排序的精确最近邻搜索法,并与其他两种相关的搜索法进行点集配准实验对比.实验结果表明,本文方法能提高低维数据的最近邻搜索效率,从而提高点集配准算法的效率.另外该方法没有采用特殊数据结构,具有程序易实现的优点.虽然本文在低维数据的最近邻搜索研究方面取得一些进展,但方法的效率将随着数据维数的增加而降低.后续的研究工作将关注于面向高维数据的最近邻搜索法.

参 考 文 献

- [1] Barber C B, Dobkin D P, Huhdanpaa H. The Quickhull Algorithm for Convex Hulls. *ACM Trans on Mathematical Software*, 1996, 22 (4): 469–483
- [2] Mulchrone K F. Application of Delaunay Triangulation to the Nearest Neighbor Method of Strain Analysis. *Journal of Structural Geology*, 2003, 25 (5): 689–702
- [3] Bentley J L. Multi-dimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*, 1975, 18 (9): 509–517
- [4] Greenspan M, Yurick M. Approximate k-d Tree Search for Efficient ICP // *Proc of the 4th International Conference on 3-D Digital Imaging and Modeling*. Banff, Canada, 2003: 442–448
- [5] Nuchter A, Lingemann K, Hertzberg J. Cached k-d Tree Search for ICP Algorithms // *Proc of the 6th International Conference on 3-D Digital Imaging and Modeling*. Montreal, Canada, 2007: 419–426
- [6] Liu Y, Xiong Y L. Algorithm for Searching Nearest-Neighbor Based on the Bounded k-d Tree. *Journal of Huazhong University of Science and Technology: Natural Science Edition*, 2008, 36 (7): 73–77 (in Chinese)
- [7] 刘宇,熊有伦.基于有界 k-d 树的最近点搜索算法. *华中科技大学学报:自然科学版*, 2008, 36 (7): 73–77
- [8] Slaney M, Casey M. Locality-Sensitive Hashing for Finding Nearest Neighbors. *IEEE Signal Processing Magazine*, 2008, 25 (2): 128–131
- [9] Hwang Y, Han B, Ahn H K. A Fast Nearest Neighbor Search Algorithm by Nonlinear Embedding // *Proc of the IEEE Conference on Computer Vision and Pattern Recognition*. Providence, USA, 2012: 3053–3060
- [10] Besl P J, McKay N D. A Method for Registration of 3D Shapes. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 1992, 14 (2): 239–256
- [11] Lu X G, Jain A K, Colbry D. Matching 2.5D Face Scans to 3D Models. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2006, 28 (1): 31–43
- [12] Shih S W, Chuang Y T, Yu T Y. An Efficient and Accurate Method for the Relaxation of Multiview Registration Error. *IEEE Trans on Image Processing*, 2008, 17 (6): 968–981
- [13] Zhu J H, Zheng N N, Yuan Z J, *et al.* A SLAM Approach by Combining ICP Algorithm and Particle Filter. *Acta Automatica Sinica*, 2009, 35 (8): 1107–1113 (in Chinese)
- [14] 祝继华,郑南宁,袁泽剑,等.基于 ICP 算法和粒子滤波的未知环境地图创建. *自动化学报*, 2009, 35 (8): 1107–1113
- [15] Du S Y, Zheng N N, Ying S H, *et al.* Affine Iterative Closest Point Algorithm for Point Set Registration. *Pattern Recognition Letter*, 2010, 31 (9): 791–799
- [16] Ying S H, Peng J G, Du S Y, *et al.* A Scale Stretch Method Based on ICP for 3D Data Registration. *IEEE Trans on Automation Science and Engineering*, 2009, 6 (3): 559–565
- [17] Zhu J H, Zheng N N, Yuan Z J, *et al.* Robust Scaling Iterative Closest Point Algorithm with the Bidirectional Distance Measurement. *Electronics Letters*, 2010, 46 (24): 1604–1605
- [18] Ying S H, Peng J G, Zheng K J, *et al.* Lie Group Method for Data Set Registration Problem with Anisotropic Scale Deformation. *Acta Automatica Sinica*, 2009, 35 (7): 867–874 (in Chinese)
- [19] 应时辉,彭济根,郑开杰,等.含各向异性尺度形变数据集匹配问题的 Lie 群方法. *自动化学报*, 2009, 35 (7): 867–874
- [20] Nüchter A, Elseberg J, Schneider P, *et al.* Study of Parameterizations for the Rigid Body Transformations of the Scan Registration Problem. *Computer Vision and Image Understanding*, 2010, 114 (8): 963–980