

基于 2 维流形的 STL 曲面网格重建算法*

杨晟院^{1,2}, 陈瑶¹, 易飞¹, 刘新¹



¹(湘潭大学 信息工程学院, 湖南 湘潭 411105)

²(智能计算与信息处理教育部重点实验室(湘潭大学), 湖南 湘潭 411105)

通讯作者: 杨晟院, E-mail: yshy@xtu.edu.cn

摘要: STL(stereo lithography)作为 3D 扫描数据和快速原型制造事实上的标准,广泛应用于娱乐、制造业和 Internet 等领域.随着 3D 模型越来越复杂,数据量越来越庞大,从 STL 文件难以快速获得完整拓扑关系且其存在大量冗余信息的缺点,制约了 STL 网格模型的进一步优化处理与应用.为此,需要针对 STL 网格模型进行网格重建.针对 2 维流形的 STL 三角形曲面网格模型,提出了一种快速的网格重建方法.主要利用删除在重建过程中达到饱和的顶点,以便减少需要比对的顶点数,并结合 STL 文件数据的相关性来提高顶点搜索与比较的效率.对于非封闭的曲面网格,该算法在提高曲面网格重建效率的同时,还有效地提取了曲面网格模型的边界信息.另外,重建的曲面网格数据文件减少了存储空间,有效地去除了冗余数据.实验结果表明了该算法的高效性及鲁棒性.

关键词: 2 维流形; STL 文件; 三角形曲面网格; 网格重建; 数据相关性

中图法分类号: TP391

中文引用格式: 杨晟院,陈瑶,易飞,刘新.基于 2 维流形的 STL 曲面网格重建算法.软件学报,2017,28(12):3358-3366. <http://www.jos.org.cn/1000-9825/5243.htm>

英文引用格式: Yang SY, Chen Y, Yi F, Liu X. STL surface mesh reconstruction algorithm based on 2 dimensional manifolds. Ruan Jian Xue Bao/*Journal of Software*, 2017, 28(12):3358-3366 (in Chinese). <http://www.jos.org.cn/1000-9825/5243.htm>

STL Surface Mesh Reconstruction Algorithm Based on 2 Dimensional Manifolds

YANG Sheng-Yuan^{1,2}, CHEN Yao¹, YI Fei¹, LIU Xin¹

¹(College of Information Engineering, Xiangtan University, Xiangtan 411105, China)

²(Key Laboratory of Intelligent Computing & Information Processing of Ministry of Education (Xiangtan University), Xiangtan 411105, China)

Abstract: As 3D data scanning and rapid prototyping manufacturing standard in fact, STL (stereo lithography) is widely used in entertainment, manufacturing, Internet and other fields. Along with the 3D model is more and more complex, the data quantity of the 3D model is more and more large. It is difficult to get the complete topological relations quickly from the STL file, and it exists a large amount of redundant information in STL files, the two defects restrict the further optimization of processing and application of the STL mesh model. For these reasons, it is need to reconstruct the mesh of STL model. Based on 2-dimensional manifold model of STL triangular surface mesh, a fast mesh reconstruction method is proposed in this paper. Mainly using the saturated vertex deletion in the reconstruction process, in order to reduce the number of vertices which needed to be compared, and combined with the correlation of STL file data to improve the efficiency of vertex search and comparison. For a non-closed surface mesh, the algorithm to improve the efficiency of surface mesh reconstruction at the same time, also can effectively extract the boundary information of the surface mesh

* 基金项目: 国家自然科学基金(11571293); 湖南省“十二五”重点实验室开放课题(2015IM05)

Foundation item: National Natural Science Foundation of China (11571293); Open Project of Key Laboratory of Hu'nan Province in 12th Five-Year Plan (2015IM05)

收稿时间: 2016-09-28; 修改时间: 2016-11-11; 采用时间: 2016-11-28; jos 在线出版时间: 2017-03-24

CNKI 网络优先出版: 2017-03-24 17:52:13, <http://kns.cnki.net/kcms/detail/11.2560.TP.20170324.1752.016.html>

model. In addition, the reconstruction of the surface mesh data file is greatly reduces the storage space, and is effectively reduces the redundant data. Experimental results show that the efficiency and robustness of the algorithm in this paper.

Key words: 2-dimensional manifold; STL file; triangular surface mesh; mesh reconstruction; data correlation

STL(stereo lithography)文件是由美国 3D SYSTEM 公司于 1988 年制定的一个接口协议,其所包含的数据是表示几何模型表面的离散、独立的三角形单元片的相关信息:顶点坐标信息以及三角形单元的单位外法向量. STL 文件由于其格式简单和良好的跨平台性,被广泛应用于科学计算可视化^[1]、计算机动画、Internert、逆向工程、快速成型及 3D 打印等领域^[2,3],并被工业界认为是 3D 扫描数据和快速原型制造事实上的标准.

随着 3D 扫描设备精度的提高和 CAD/CAM 应用领域的拓展,需要处理的 3D 几何模型也越来越复杂.为了精确、有效地逼近复杂几何模型的边界曲面,一般需要几十万甚至几百万个三角形单元网格片来表示几何模型的面,因此生成网格模型的 STL 文件数据量也越来越大.虽然 STL 文件格式简单,但是它也存在一些缺点:(1) 缺乏拓扑关系,STL 文件中并不包含任何三角形单元之间的拓扑邻接信息,因此在实际应用中,不能从 STL 文件中直接提取所需的拓扑邻接信息;(2) 存在大量冗余,由于顶点的坐标信息会在多个三角形单元的顶点数据中重复出现,因此 STL 文件的数据冗余现象比较严重.这些缺点制约着 STL 网格模型的应用.可见,剔除冗余数据并快速建立完整的拓扑关系,是基于 STL 网格模型进一步优化处理与应用的必要前提.

针对 STL 文件格式的三角形网格模型的冗余数据滤除和快速拓扑重建,许多学者进行了大量的研究^[4-10].其中,Shin 等人^[4]利用桶排序的思想进行邻域搜索,采用存储点和面的数据结构,并不需要构建一个满的非流形拓扑结构,但是在搜索邻域信息时进行桶排序时复杂度较高,因此效率不高.张必强等人^[5]采用平衡二叉树(AVL)结构进行顶点聚合,去除了大量的冗余点,但当网格规模变大时,重建过程中过多的平衡操作会降低查找效率.安涛等人^[6]采用红黑树的方法进行模型重建,在读取小数据量的文件时所需时间比较低;但当针对大数据量的处理时,耗费时间仍旧较多.郑惠江等人^[7]与王增波^[8]采用基于哈希函数的方法,虽然在顶点滤除过程中减少了不必要的顶点比对次数,但哈希函数的选择对算法效率的影响较大,过于复杂或简单的哈希函数都会降低算法的效率.

以上这些基于 STL 文件的网格模型重建算法基本上是将冗余顶点的去除和拓扑关系的重建这两个步骤分开处理,效率不高.很少有学者从网格模型本身的几何特性出发进行网格重建处理.虽然王坚等人^[9]从三角片拼合的角度考虑了部分网格模型的几何特性,但计算效率随着网格规模的增大而降低.杨晟院等人^[10]分析了由各类 CAD,CAXA 等软件生成的大量 STL 三角形网格数据,发现前后相邻的两个三角形单元共一个顶点或共边的概率非常高,即,三角形相邻单元之间的具有一定的数据相关性,并从数据相关性的角度给出了一种有效的重建算法.但其算法时间复杂度与直接重建算法相同,随着网格规模的增大,其算法效率下降明显.

在基于 STL 文件的三角形网格模型重建过程中,顶点的比较非常耗时.事实上,对于 2 维流形的几何模型,当与某一顶点相关的所有三角形网格单元都拼合后,该点便变成饱和点,即,该点不属于后续任意三角形单元的顶点了.在后续的网络重建过程中,也不需要再与饱和点进行比对.而文献[10]中的算法效率低的原因在于进行顶点的比对时,需要与大量饱和点进行无用的比对操作.因此,在网络重建过程中剔除饱和点后,可以减少大量不必要的顶点比对操作,从而提高算法的效率.为此,本文在剔除饱和点的同时,结合 STL 文件的数据相关性,给出了一种新的高效的基于 2 维流形的 STL 曲面网格重建算法.

1 相关定义与 STL 文件格式

1.1 STL文件格式与基本定义

STL 文件有两种存储格式:文本格式和二进制格式,本文只针对文本格式进行描述.STL 文件是由表达几何模型表面的一系列离散的三角形单元片组成的集合.其中,每个三角形单元片的数据信息仅保存该三角形单元的指向模型外部的单位法向量以及三角形 3 个顶点的坐标信息,其存储格式如下:

Solid

...

facet normal 4.0213842e-001 9.0815002e-001 1.1639693e-001

outer loop

vertex 1.0915299e+001 1.1778202e+001 1.0490475e+001

vertex 1.0829356e+001 1.1819937e+001 1.0461781e+001

vertex 1.0684040e+001 1.1879385e+001 1.0500000e+001

endloop

endfacet

...

EndSolid

为了更好地理解 2 维流形曲面三角形网格上的点、线、面等几何元素,本文给出如下一些定义.

定义 1(边的定义与分类). 连接两个顶点的线段即为边,连接两个三角形单元的边称为内部边,只属于一个三角形单元的边称为边界边.

定义 2(片的定义). 若干共点 i 的三角形单元集合称为点 i 的片.如果点 i 的片中所有与点 i 相连的边都是内部边,则称该片为点 i 的饱和片.

定义 3(点的分类). 顶点为边界边上的点,称为边界点;如果顶点所连的所有边都是内部边,或者顶点所对应的片为饱和片,则称该点为内部点或者饱和点.

在 2 维流形曲面三角形网格重建过程中,逐步将某一顶点 i 所连的三角形单元进行拼合时,当点 i 变成内部点时,后续拼合的三角形单元则不可能再与点 i 共点,即可认为点 i 达到饱和状态了.因此,也称内部点为饱和点.图 1 中,顶点 1 为饱和点,其他顶点 2,3,...,9 都是边界点.和顶点 1 相连的边都是内部边,如边 $e(1,2),e(1,3)$ 等是内部边,其他边如 $e(2,3),e(3,4)$ 等是边界边.饱和点 1 所对应的饱和片由三角形①~三角形⑥组成,顶点 3 所对应的片由三角形①、三角形②组成,顶点 7 所对应的片由三角形⑤~三角形⑦组成.

本文仅对 2 维流形的几何曲面网格模型进行处理,为此,下面给出流形的定义.

定义 4(d 维拓扑流形). 设 M 是 n 维的 Hausdorff 拓扑空间,若对每一点 $p \in M$ 都有 p 的一个开邻域 U 和 R^d 中点 q 的一个开子集同胚,则称 M 是 d 维拓扑流形,简称为 d 维流形.当取 $n=3, d=2$ 时,如图 2 所示的三维空间曲面 M 即为 2 维流形.

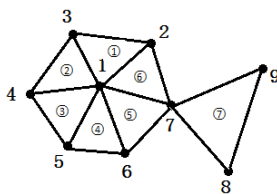


Fig.1 Triangular mesh

图 1 三角形网格示意图

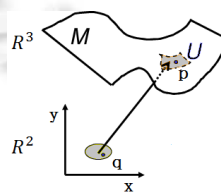


Fig.2 2-Dimensional manifolds surface M

图 2 2 维流形的曲面 M

根据 2 维流形的定义可知:属于 2 维流形的曲面三角形网格中,内部边只可能连接两个三角形单元;饱和点所对应的片只能为一个饱和片.对于 2 维流形的封闭曲面三角形网格而言,所有的顶点都是饱和点,所有的边都是内部边.

1.2 STL重建的文件格式

STL 文件重建的文件格式有许多种,常见有将顶点信息和拓扑结构分开存储为*.cor 和*.elm 的文件格式^[10]和 OFF(object file format)格式^[11]等.

本文采 OFF 文件格式,OFF 文件通过描述物体表面的多边形来表示一个模型的几何结构,这些多边形可以包含任意数量的顶点.OFF 文件遵从以下标准.

- 1) OFF 文件为 ASCII 文件,以 OFF 关键字开头;
- 2) 下一行是模型的顶点数、面数和边数.边数可以忽略,对模型不会有影响(可以为 0);
- 3) 顶点以 x,y,z 坐标列出,每个顶点占一行;
- 4) 在顶点列表之后是面列表,每个面占一行.对于每个面,首先指定其包含的顶点数,随后是这个面所包含的各顶点在前面顶点列表中的索引.

其格式即为:

OFF

顶点数 面数 边数

$x y z$

$x y z$

...

n 个顶点 顶点 1 的索引 顶点 2 的索引 ... 顶点 n 的索引

...

下面是一个四面体的例子.

OFF

4 4 0

0.0 0.0 0.0

1.0 0.0 0.0

0.0 1.0 0.0

0.0 0.0 1.0

3 0 2 1

3 2 0 3

3 1 3 0

3 1 2 3

OFF 文件中顶点的编号是从 0 开始的,而本文为了便于算法描述,顶点编号从 1 开始.写入 OFF 文件时将顶点编号减 1 即可.

2 算法原理

在基于 STL 文件的三角形网格模型重建过程中,主要是顶点的比较非常耗时.杨晟院等人^[10]从 STL 数据相关性的角度给出了一种有效的重建算法,但该算法时间复杂度与直接重建算法相同,随着网格规模的增大,其算法效率明显下降.该算法效率低的原因在于进行顶点的比对时,需要与大量饱和点进行无用的比对操作.事实上,对于 2 维流形的几何模型,在网格重建过程中,当某一顶点变成饱和点,后续任意三角形单元的顶点都不需要再与饱和点进行比对.因此,在网格重建过程中剔除饱和点后,可以减少大量不必要的顶点比对操作,从而提高算法的效率.为了有效剔除饱和点,可建立一个广义的顶点链表,将依次读取的新顶点添加到链表的尾部,并依次对顶点进行编号.同时,为顶点链表中的每一个顶点建立子链表,将所有与该点相连的边界边的另一顶点放到该点的子链表中,而子链表中为内部边上的点则删除.如果顶点的子链表为空,即该点变成饱和点,则删除该点.具体操作如下.

首先,读取图 1 中的第 1 个三角形单元的顶点坐标信息,将该三角形单元编为 1 号三角形单元,并依次将读取的顶点编为 1,2,3 号顶点.为此,可以构建读取第 1 个三角形单元时的初始广义顶点链表结构(如图 3 所示).图 3 中,左边垂直的列为依次进行编号的顶点链表,右边横向的各个子链表为每一个顶点所连接的边界点的子链

表.此时,所重建的网格还只有一个三角形单元,所以所有的点都还是边界点.根据文献[10]的数据相关性,将这个三角形单元的顶点信息存储在一个临时的 temp 结构中,以便于下一个三角形单元顶点坐标信息的快速比对.

接着,读取下一个三角形单元的顶点信息(如图 1 中的第 2 号三角形单元),将 3 个顶点依次与 temp 结构中的顶点信息进行快速比对,发现其中有两个顶点已经分别编为 1 号点和 3 号点;另一顶点在 temp 中没有匹配点,则需与广义顶点链表的顶点进行逆序比对,发现该点为新顶点,则依次编为 4 号顶点并将其插入顶点链表尾部,并更新 1 号,3 号,4 号顶点各自对应的链表信息(如图 4 所示).同时,将新读取的三角形单元的顶点信息存储到 temp 结构中.

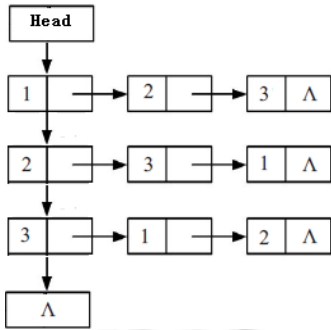


Fig.3 Initial generalized vertex list

图 3 初始广义顶点链表

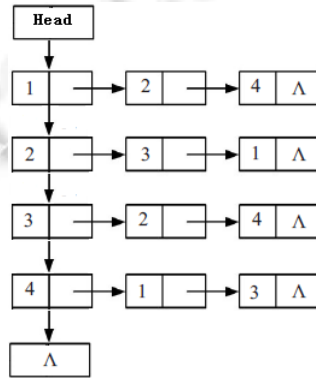


Fig.4 Splice vertex list

图 4 拼接一个单元的顶点链表

新读取的三角形单元 3 个顶点各自子链表信息的更新规则是:如果是新顶点,则创建新顶点的子链表,并将另外两个顶点的编号信息插入子链表中;如果不是新顶点,则将该顶点所连接的另两个顶点与该顶点的子链表中的顶点进行比对,删除子链表中的重复点,同时添加非重复的顶点到子链表中.例如,4 号点是新顶点,则创建 4 号顶点的子链表,并将 1 号和 3 号顶点的编号插入其子链表中;与 1 号顶点相连接的两个顶点分别为 3 号、4 号顶点,将 3 号、4 号顶点与 1 号顶点的子链表进行比对,发现 3 号点已经存在,则将子链表中的 3 号点删除,而子链表中没有 4 号顶点,因此,将 4 号顶点插入 1 号顶点的子链表中;与 3 号顶点相连接的两个顶点分别为 1 号、4 号顶点,将 1 号、4 号顶点与 3 号顶点的子链表进行比对,发现 1 号点已经存在,则将子链表中的 1 号点删除,而子链表中没有 4 号顶点,因此,将 4 号顶点插入 3 号顶点的子链表中.图 4 为拼接第 2 号三角形单元后顶点链表的结构示意图.

最后是删除饱和点.由上面的更新规则可知:每一次拼接三角形单元后,只有新读取的三角形单元的 3 个顶点有可能成为新的饱和点.因此,只需检测这 3 个顶点的子链表是否为空就行了.如果其子链表为空(不包含该顶点本身),则该点一定是饱和点,删除.如图 5 中的顶点 1,在紧接着拼接一个三角形单元后就变成饱和点,因此在拼接完这个三角形单元后,需要在顶点链表中删除顶点 1.

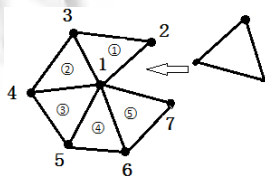


Fig.5 Splice the triangular meshes

图 5 三角形网格拼接

对于封闭的曲面三角形网格模型,重建结束后,所有的顶点都将变成饱和点.可见,顶点链表中的顶点随着网格模型的重建逐渐变成饱和点而被删除,导致顶点链表最后会变成空链表.如果是非封闭的曲面三角形网格,

网格重建结束后,链表中留存的顶点就是曲面三角形网格的边界点,即,本文的曲面网格重建算法能够有效地提取网格的边界信息.

3 算法实现

根据以上算法原理的分析,本节给出基于 STL 文件的拓扑重建算法流程.其主流程图如图 6 所示:

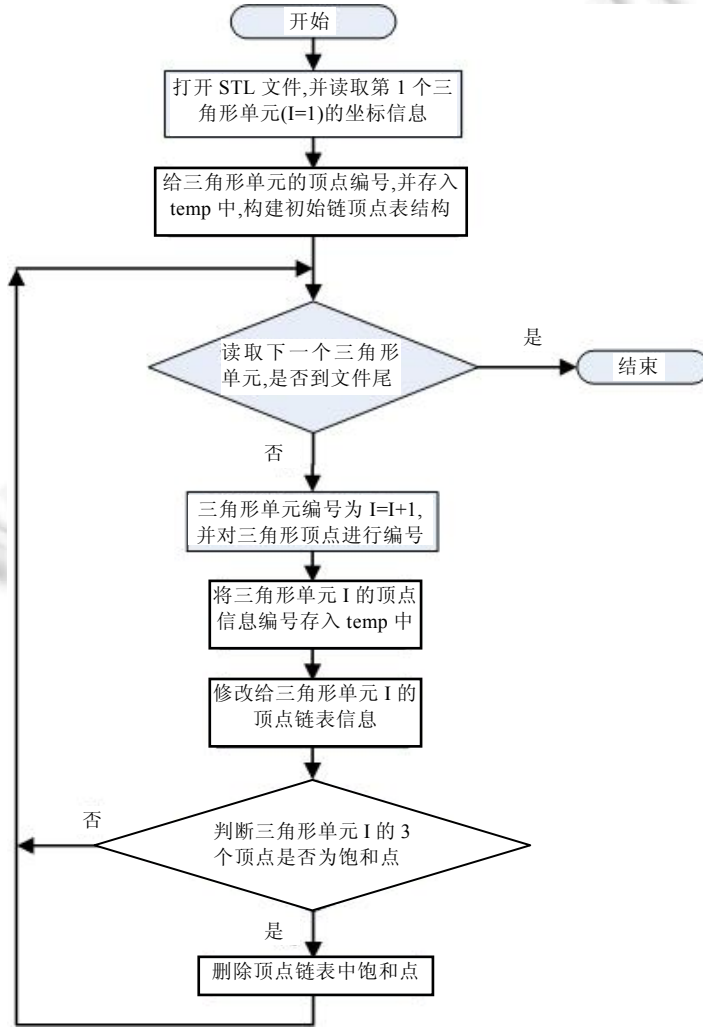


Fig.6 Main program flow diagram

Fig.6 主程序流程图

基于 2 维流形的 STL 曲面网格重建算法.

- Step 1. 打开 STL 文件,读取文件中的第 1 个三角形单元的顶点数据.将该三角形单元索引编号设定为 $i=1$,其 3 个顶点的全局索引编号以逆时针方向依次设定为 1,2,3;建立广义表的初始表结构(如图 3 所示),并建立初始的“点-边-面”拓扑关系.并将三角形单元的顶点坐标以及其索引编号存入临时结构 temp 中.
- Step 2. 继续读取 STL 文件数据,如果到文件结尾则停止;否则,顺序读取下一个三角形单元的顶点数据,该三角形单元索引编号为 $i=i+1$,并进行如下操作:

- S1. 将最新读取的 3 个顶点的坐标数据与上一个三角形单元的顶点数据(在 temp 中)进行比较, 如果与 temp 中的顶点相同,则采用其编号;如果顶点不在 temp 中,再将其与顶点链表进行比较:
- 如果该点已经属于链表,则采用其编号;
 - 如果不在链表中,则该点为新顶点,将新顶点的编号设定为链表中的最大编号值加 1; 同时,将新读取的三角形单元的顶点坐标以及其索引编号存入临时结构 temp 中.
- S2. 如果出现新编号的顶点,则将其插入顶点链表尾部.更新最新读取三角形 3 个顶点各自对应的子链表信息.更新规则为:如果是新顶点,则创建新顶点的子链表,并将另外两个顶点的编号信息插入子链表中;如果不是新顶点,则将该顶点所连接的另两个顶点与该顶点的子链表中的顶点进行比对,删除子链表中的重复点,同时添加非重复的顶点到子链表中.
- S3. 检查这 3 个更新链表信息的顶点,如果其尾指针为空,则其为饱和点,将其删除.
- S4. 返回 Step2.

4 算法比较与分析

本文分别采用基于数据相关性的算法^[10]、AVL 算法^[5]、关联-散列算法^[7]和本文算法分别对不同规模的 STL 模型进行拓扑重建(测试环境为:4G 内存,Win7 Sp1 操作系统,i3-3220 CPU@3.30GHz,VC++6.0),实验结果见表 1.

Table 1 Results for the algorithms
表 1 三角形网格拓扑重建对比结果

| STL 模型 | 三角形单元(个) | 顶点数(个) | 时间(s) | | | |
|--------|-----------|---------|-----------|---------|---------|--------|
| | | | 数据相关性算法 | AVL 算法 | 关联-散列算法 | 本文算法 |
| 模型 1 | 7 616 | 3 806 | 0.171 | 1.33 | 0.136 | 0.08 |
| 模型 2 | 16 054 | 8 029 | 0.798 | 1.89 | 0.203 | 0.169 |
| 模型 3 | 87 072 | 43 549 | 18.574 | 4.49 | 1.253 | 0.7985 |
| 模型 4 | 212 508 | 106 264 | 107.737 | 10.93 | 3.473 | 2.23 |
| 模型 5 | 235 524 | 118 309 | 142.387 | 56.931 | 8.468 | 5.049 |
| 模型 6 | 515 668 | 258 475 | 677.362 | 129.794 | 27.394 | 19.83 |
| 模型 7 | 616 184 | 308 773 | 964.276 | 143.906 | 47.017 | 20.743 |
| 模型 8 | 798 622 | 400 034 | 1 643.397 | 176.557 | 65.76 | 31.994 |
| 模型 9 | 1 125 268 | 563 439 | 3 382.146 | 308.741 | 97.654 | 56.129 |

实验结果显示:基于数据相关性算法的时间复杂度是呈指数关系增长,其本质上就是一种直接法.AVL 算法的时间复杂度是 $O(\log_2 n)$,该算法对于数据量较大的情形调整树的平衡所花费的时间会增多.关联-散列算法与本文算法的时间复杂度基本呈线性关系(如图 7 所示),但本文算法在没有采用复杂数据结构的情形下,比关联-散列算法的效率更高.

本文算法实现网格模型的重构过程中,随着读取的三角形单元数量的增加,顶点链表中的顶点数也会随之增加.但随着饱和点的删除,顶点链表中顶点的个数并不随着三角形单元数的增加而一直增加,它会达到一个最大值后,顶点个数就不会超过该值并且会逐渐波动变少(如图 8 所示).对于封闭的 2 维流形曲面网格,重建过程中,顶点链表内的顶点数会从 0 波动变化到一个最大值后,会再波动变化为 0.对于非封闭的 2 维流形曲面网格,重建完成时,顶点链表中仍然存在顶点,这些顶点都是边界点.因此,本文的重建算法在有效地重建曲面网格的同时,又能很方便地提取非封闭 2 维流形曲面网格的边界信息.

图 8 为模型 1 在重构模型过程中,顶点链表中内顶点个数的统计图.模型 1 的三角形单元数为 7 616,顶点数为 3 806.随着三角形单元的读取,顶点链表中顶点数开始增加,当读取到第 1 849 个三角形单元时,顶点数达到最大值 673 个.但是直至读取完,该模型顶点链表中的顶点数也没有超过最大顶点数,都是在 673 个顶点以下波动,最后波动减少为 0.这说明在模型 1 的网格重构过程中,每一个顶点的比对最大次数不超过 673.表 2 为采用本文算法重构各模型时顶点链表中存储的顶点曾达到的最大值以及所需的重构时间.

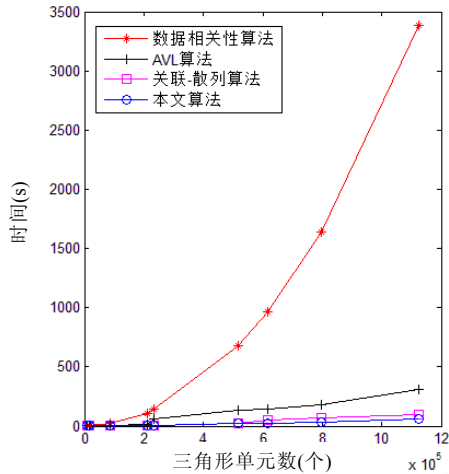


Fig.7 Times for the for four algorithms
图 7 4 种算法计算时间比较图

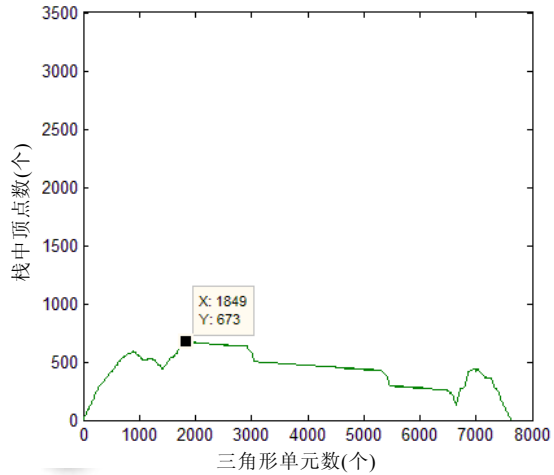


Fig.8 Vertex number of vertex table in model 1
图 8 模型 1 顶点链表的顶点数

Table 2 Time of model reconstruction and the maximum number in the table of vertex chain

表 2 模型重构时间与顶点链表中的最大顶点数

| STL 文件 | 三角片面(个) | 顶点数(个) | 时间(s) | 栈中最大顶点数(个) |
|--------|-----------|---------|--------|------------|
| 模型 1 | 7 616 | 3 806 | 0.08 | 673 |
| 模型 2 | 16 054 | 8 029 | 0.169 | 830 |
| 模型 3 | 87 072 | 43 549 | 0.7985 | 736 |
| 模型 4 | 212 508 | 106 264 | 2.23 | 1 119 |
| 模型 5 | 235 524 | 118 309 | 5.049 | 1 915 |
| 模型 6 | 515 668 | 258 475 | 19.83 | 2 237 |
| 模型 7 | 616 184 | 308 773 | 20.743 | 2 065 |
| 模型 8 | 798 622 | 400 034 | 31.994 | 2 114 |
| 模型 9 | 1 125 268 | 563 439 | 56.129 | 2 622 |

另外,根据文献[10]的数据相关性,STL 文件中有超过 80%的顶点在与 temp 进行一次比对就能快速确定编号;只有少于 20%的顶点需要与顶点链表中的顶点进行比对.而这 20%的顶点,采用逆序的方式与顶点链表内的顶点进行比对:对于重复顶点的比对,也能较快地进行确定(顶点链表中的顶点数不会超过某个最大值,并且还会逐渐减少);只有新顶点则需与顶点链表内的所有顶点进行一次比对.从统计概率的角度,本文算法也很大程度上降低了顶点数据的比对次数,从而提高算法效率.

5 结论与展望

本文针对 2 维流形的 STL 三角形曲面网格模型,利用饱和点不会再出现在后续三角形单元中的特性,将重建过程中顶点链表内变成饱和点的顶点删除,以便减少后续需要比对的顶点数,并结合 STL 文件数据的相关性来提高顶点搜索与比较的效率,从而给出了一种高效的 STL 曲面网格重建算法.对于非封闭的曲面网格,本文算法在提高曲面网格重建效率的同时,还能有效地提取曲面网格模型的边界信息;并且极大地降低了重建曲面网格数据文件所占的存储空间,有效地去除了冗余数据.

本文算法在没有采用复杂数据结构的情形下,比关联-散列算法的效率更高,实验结果表明了本文算法的高效性及鲁棒性.在后续研究中,将在数据的存储结构上进行改进来提高重构的效率,也可拓展用于非 2 维流形的处理.

References:

- [1] Béchet E, Cuilliere JC, Trochu F. Generation of a finite element mesh from stereo lithography (STL) files. *Computer-Aided Design*, 2002,34(1):1-17. [doi: 10.1016/S0010-4485(00)00146-9]
- [2] He HG, Tian J, Zhang XP, Zhao MC, Li GM. A survey on mesh simplification. *Ruan Jian Xue Bao/Journal of Software*, 2002,13(12):2215-2224 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/13/2215.htm>
- [3] Sun XP, Gai Y, Xu N, Li Z. 3D ear mesh merge algorithm based on patch removing. *Ruan Jian Xue Bao/Journal of Software*, 2016,27(10):2612-2621 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5088.htm> [doi: 10.13328/j.cnki.jos.005088]
- [4] Shin HY, Park JC, Choi BK, Chung YC, Rhee S. Efficient topology construction from triangle soup. In: *Proc. of the Geometric Modeling and Processing*. Beijing, 2004. 359-364. [doi: 10.1109/GMAP.2004.1290060]
- [5] Zhang BQ, Xing Y, Ruan XY. Fast generation of the topological information in STL for mesh simplification. *Journal of Shanghai Jiaotong University*, 2004,38(1):39-42 (in Chinese with English abstract). [doi: 10.3321/j.issn:1006-2467.2004.01.010]
- [6] An T, Dai N, Liao WH, Yuan TR. An efficient algorithm for topological reconstruction of STL data. *Mechanical Science and Technology for Aerospace Engineering*, 2008,27(8):1031-1034 (in Chinese with English abstract). [doi: 10.3321/j.issn:1003-8728.2008.08.012]
- [7] Zheng HJ, Wang TY, He GY. Research of topological reconstruction for triangular mesh in OMV. *Computer Engineering and Applications*, 2010,46(30):212-215 (in Chinese with English abstract). [doi: 10.3778/j.issn.1002-8331.2010.30.061]
- [8] Wang ZB. Fast topological reconstruction algorithm for a STL file. *Journal of Computer Applications*, 2014,34(9):2720-2724 (in Chinese with English abstract). [doi: 10.11772/j.issn.1001-9081.2014.09.2720]
- [9] Wang J, Zhou LS, Zhang WZ. STL mesh model reconstruction by assembling triangles. *Journal of Computer-Aided Design & Computer Graphics*, 2006,18(11): 1758-1764 (in Chinese with English abstract). [doi: 10.3321/j.issn:1003-9775.2006.11.021]
- [10] Yang SY, Shu S. Fast reconstruction of STL surface meshes based on data correlation. *Journal of Computer-Aided Design & Computer Graphics*, 2009,21(1):68-71 (in Chinese with English abstract).
- [11] http://shape.cs.princeton.edu/benchmark/documentation/off_format.html

附中文参考文献:

- [2] 何晖光,田捷,张晓鹏,赵明昌,李光明. 网格模型化简综述. *软件学报*, 2002,13(12):2215-2224. <http://www.jos.org.cn/1000-9825/13/2215.htm>
- [3] 孙晓鹏,盖宇,徐南,李志. 耳廓三维网格去补丁合并算法. *软件学报*, 2016,27(10):2612-2621. <http://www.jos.org.cn/1000-9825/5088.htm> [doi: 10.13328/j.cnki.jos.005088]
- [5] 张必强,邢渊,阮雪榆. 面向网格简化的 STL 拓扑信息快速重建算法. *上海交通大学学报*, 2004,38(1):39-42. [doi: 10.3321/j.issn:1006-2467.2004.01.010]
- [6] 安涛,戴宁,廖文和,袁天然. 基于红黑树的 STL 数据快速拓扑重建算法. *机械科学与技术*, 2008,27(8):1031-1034. [doi: 10.3321/j.issn:1003-8728.2008.08.012]
- [7] 郑惠江,王太勇,何改云. 在机检测中三角网格拓扑重建方法的研究. *计算机工程与应用*, 2010,46(30):212-215. [doi: 10.3778/j.issn.1002-8331.2010.30.061]
- [8] 王增波. STL 格式文件的快速拓扑重建算法. *计算机应用*, 2014,34(9):2720-2724. [doi: 10.11772/j.issn.1001-9081.2014.09.2720]
- [9] 王坚,周来水,张维中. 基于三角片拼合的 STL 网格模型重建算法. *计算机辅助设计与图形学学报*, 2006,18(11):1758-1764. [doi: 10.3321/j.issn:1003-9775.2006.11.021]
- [10] 杨晟院,舒适. 基于数据相关性的 STL 曲面网格快速重建算法. *计算机辅助设计与图形学学报*, 2009,21(1):68-71.



杨晟院(1974—),男,湖南绥宁人,博士,副教授,CCF 专业会员,主要研究领域为图形图像处理,网格生成与优化.



易飞(1992—),男,硕士生,主要研究领域为图像配准.



陈瑶(1992—),女,硕士生,主要研究领域为图形图像.



刘新(1975—),男,博士,副教授,CCF 专业会员,主要研究领域为信息安全,智能计算.