

三角网格模型骨架提取算法

王洪申, 张家振[†], 张小鹏

(兰州理工大学机电工程学院, 甘肃 兰州 730000)

摘要:骨架图能够直观表达三维模型几何形状,很好地反映模型的拓扑特征,在工业机器人抓取、特征识别等领域有着广泛的应用。针对三角网格表达的工业零件给出一种骨架提取算法,该算法采用Reeb图对三角网格进行骨架的抽取运算。首先读取三角网格文件,并对复杂的三角网格进行简化处理,然后遍历所有的三角网格,采用Dijkstra算法抽取基本点集,根据定义连续函数计算每个顶点的函数值,最后根据函数值得出模型的基本骨架。实验表明,该算法具有良好的计算效果和效率,提取出的骨架图较好地保存了三维模型拓扑结构和姿态,可作为后续研究三维模型搜索的特征描述符。

关键词:骨架图;三角网格;三维模型;拓扑结构;Reeb图

中图分类号:G633.6

文献标识码:A

Triangular Mesh Model Skeleton Extraction Algorithm

WANG Hong-shen, ZHANG Jia-zhen[†], ZHANG Xiao-peng

(School of Mechanical and Electrical Engineering, Lanzhou University of Technology, Lanzhou, Gansu 730000, China)

Abstract: The skeleton diagram can visually express the geometry of the 3D model and reflect the topological features of the model well. It has a wide range of applications in the fields of industrial robot capture and feature recognition. A skeleton extraction algorithm is proposed for the industrial parts expressed by the triangle mesh. The algorithm uses the Reeb diagram to extract the skeleton from the triangular mesh. First read the triangle mesh file, and simplify the complex triangle mesh, then traverse all the triangle meshes, extract the basic point set by Dijkstra algorithm, calculate the function value of each vertex according to the defined continuous function, and finally The function deserves the basic skeleton of the model. Experiments show that the proposed algorithm has good computational efficiency and efficiency. The extracted skeleton map preserves the topology and pose of the 3D model, and can be used as a feature descriptor for the subsequent research of 3D model search.

Key words: skeleton diagram; triangular mesh; 3D model; topology; Reeb diagram

随着计算机图形学的发展,对三维模型的研究日益深入,骨架作为形状表示的一种有效形式,在三维模型的各个研究领域被运用。骨架的狭义定义最初由Blum^[1]提出,当时他称骨架为“中轴”(Medial Axis)^[2]。骨架的经典定义有两种^[3]:一种是烧草模型,如图1所示,从模型表面开始点火,火焰从物体边界上的两点同时向内部推进,轨迹随时间形成等

距的同心圆,同心圆的相遇点所构成的集合即为骨架;另外一种更直观的定义,即最大圆盘模型。如图2所示,骨架点是所有最大圆盘的圆心集合,最大圆盘即是完全包含在物体内部且至少与物体边界相切于两点的圆。骨架上的每一个点都是这些内切圆的圆心,这些圆沿着骨架分布正好填充物体的内部。由于模型的骨架很好地保留了模型的拓扑形

收稿日期:2019-05-14

作者简介:王洪申(1969—),男,甘肃兰州人,博士,教授,研究方向:CAD/CAM。

[†] 通讯联系人, E-mail: 2834298258@qq.com

态及其连接特性,所以经常被用于模型渲染、模型表面重建、碰撞检测、模型检索等应用中,在工业零件的视觉识别领域也有广泛的用途。

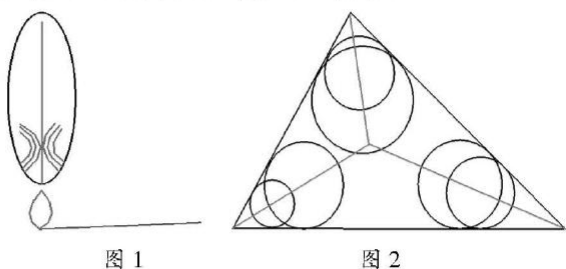


图 1

图 2

针对三维模型,骨架的提取方法主要有如下几种:(1)基于拓扑细化技术^[9]。该类算法主要用于体表示的模型上,通过不断地进行体元素的削减来实现骨架的抽取,所以整个过程比较耗时,效率不高;(2)基于距离矩阵的方法^[4]。它一般也是针对体表示的模型,通过计算每个体元素的距离来求取模型的脊点;(3)基于二维图像领域的 Voronoi 图技术。有学者将二维领域的 Voronoi 图技术引入到三维模型的骨架提取中来,Dey 等人提出过一种利用 Voronoi 图直接近似中轴的算法^[2];(4)烧草模型法。正如上面的烧草模型所述,在模型的一端点火,模型两侧同时燃烧,火焰相聚处形成连贯的烧痕,这就是模型骨架的相似轨迹,但是这种方法理论性强,实际操作起来却十分复杂;(5)基于 Reeb 图的思想。如图 3 所示,该算法首先在模型上定义一个连续函数,计算每个顶点的函数值,将相同函数值的顶点聚合成一个顶点,连接各个聚合后的顶点,形成骨架图。其中最著名的是 Hilaga 等人提出的多分辨率 Reeb 图 (Multiresolutional Reeb Graph, 简称 MRG) 算法^[5],它可以表示不同分辨率级别的 3D 形状的骨架和拓扑结构。图 3 表示了不同分辨率的 Reeb 图,在模型上定义一个连续函数 U ,通过三种不同分辨率来计算 Reeb 图,由图可见,函数区间分割的层次越多,所得的 Reeb 图越精确。

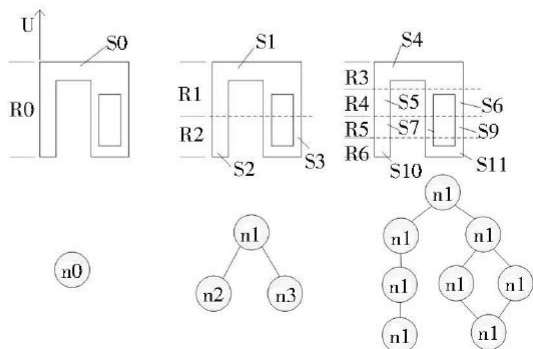


图 3 三个不同分辨率的 Reeb 图

另外,国内外的其他学者也对物体的骨架提取技术进行了一些研究。意大利学者 Jacopo Aleotti^[6]通过摄像机观测三维物体,将三维物体拓扑分解成若干部分,从而实现骨架的提取;葡萄牙学者 Diego R. Faria^[7]提出了两种模型拓扑分离方法:基于物体主轴的分离方法和基于高斯混合模型的分离方法,结果发现后者运算效率低,花费时间长,采用基于主轴的分离方法效果要好;国内学者官法明^[8]提出了基于特征点求解的骨架提取算法,能够快速地提取出模型骨架,但对于复杂的模型还存在过多的问题。

提出了针对三角网格模型的骨架提取算法,首先对复杂的三角网络进行简化处理,然后遍历所有的三角形,采用 Dijkstra 算法抽取基本点集,根据定义的 Morse 函数计算每个顶点的函数值,最后根据函数值得出模型的基本骨架。算法的程序流程图如下所示:

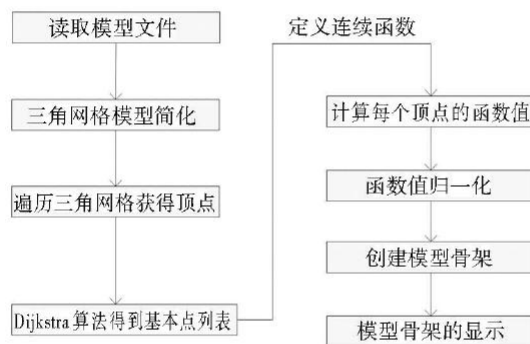


图 4 算法的程序流程图

1 读入模型文件

STL 文件是在计算机图形应用系统中用于表示三维网格模型的一种文件格式^[9]。三维模型采用 STL 文件格式进行表达。

以 vs2010 为平台,运用 VTK 开源软件包面向对象原理设计和实现算法。VTK 自身提供了一种进行网格读取的方法,它是将一个模型三角网格的所有顶点坐标以及它们的索引号直接在程序中表达出来,然后通过循环遍历读取每一个网格及顶点。这种方法具有局限性,只适用于三角网格所有顶点坐标以及每个点的索引号已知的情况下,模型三角网格的读取。而大量的三维模型网格复杂,无法得到模型三角网格所有顶点以及索引号,这时就需要利用程序读取模型的网格信息。

旨在设计出一种针对任意模型,可以读取三角网格信息的算法,这样可以处理的模型范围广泛,也能精简程序的长度,以长方体为例进行说明。

长方体有 8 个顶点,6 个面,对应的 STL 文件中有 8 个顶点和 12 个三角形网格。图 5 是一个长方体的三角网格模型,每个面被分成了两个三角网格。取其中一个三角网格进行分析,它包含了 3 个顶点,每个顶点对应着一个索引号。所以,如果想要完全表达这个长方体,分为两个步骤:1.从某一个顶点开始遍历所有的顶点,给出顶点的三维坐标,并将索引号与每个顶点相对应;2.从某一个三角网格开始遍历所有的三角网格,并给出每一个网格顶点的索引号。

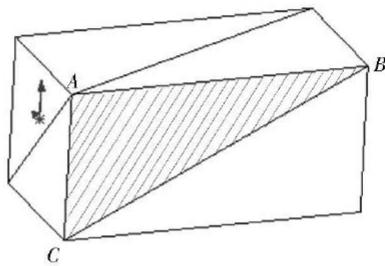


图 5 长方体的 STL 模型

根据以上的分析,可以列出求解各个顶点三维坐标的程序:

如图 6 所示,其中 i 代表第 i 个顶点, N 代表总顶点数, $P_i^{[3]}$ 代表第 i 个顶点的三维坐标,而 vId 代表该顶点对应的索引号。三维 STL 文件导入后,由第一个顶点开始遍历,遍历每个顶点时,输出顶点的三维坐标,然后将每个顶点的索引号与坐标相对应,直到遍历完所有的顶点。

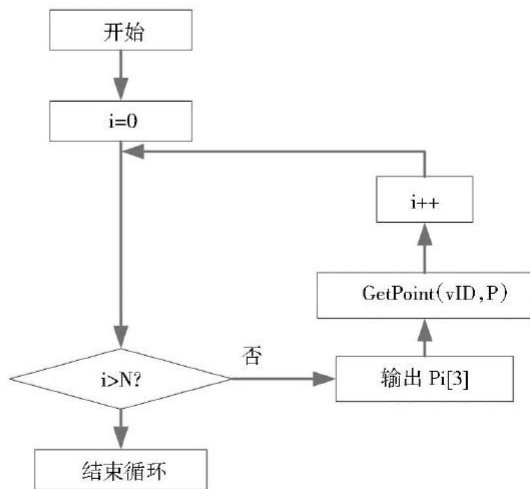


图 6 输出顶点坐标的程序流程图

同样的思路,也可以列出求解网格数据的程序流程图,如图 7 所示。

在图 7 中, j 代表第 j 个网格, M 代表总网格数目, k 代表第 j 个网格的第 k 个顶点。 $GetCellPoints()$ 将每个三角网格的顶点提取出来,然后根据提取出来的顶点进行二次遍历。 $GetId(k)$ 将第 k 个顶点的索引号提取出来。依次循环遍历,直到遍历完所有的网格。

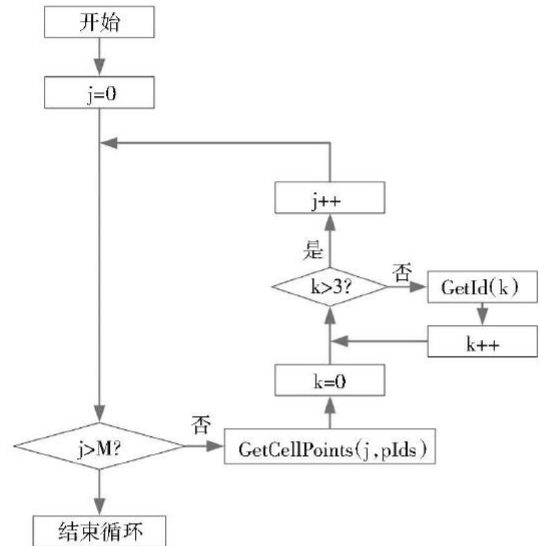


图 7 求解网格数据的程序流程图

用上述方法提取出长方体每个索引号对应的顶点坐标,如表 1 所示。(P—顶点索引号)

表 1 顶点索引号对应的顶点坐标

P	1	2	3	4	5	6	7	8
X	0	0	0	0	10	10	10	10
Y	100	100	0	0	100	0	100	0
Z	10	0	10	0	10	10	10	0

上述长方体共有 12 个三角网格,每个网格对应三个顶点,每个三角网格对应三个顶点的索引号如表 2 所示。(M—三角网格序号)

表 2 三角网格对应的顶点索引号

M	1	2	3	4
P	0 1 2	2 1 3	4 0 5	5 0 2
M	5	6	7	8
P	6 4 7	7 4 5	1 6 3	3 6 7
M	9	10	11	12
P	4 6 0	0 6 1	7 5 3	36 5 2

这样,就将长方体模型的几何特性完全表达了出来,为后续研究提供了前提。

2 三维模型网络的简化

导入网格文件后,为了便于后续处理,需要先从三维模型进行简化,采用 OpenGL 中使用层次细节(level of details,缩写 LOD)方法来实现^[10]。

LOD 方法运用最多的有两种算法,分别是压缩三角形算法和折叠三角形算法^[11]。对于第一种算法,是将要简化的三角形网格的三个顶点压缩至一个点,实现网络的简化。如图 8 所示,△M1M2M3 是要压缩的三角形,将三角形的三条边设为 0,则△M1M2M3 就变成了点 M。

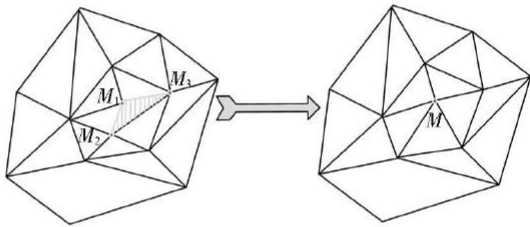


图 8 压缩三角形算法

对于第二种算法,基本思路是将两个三角形的共线端的两顶点重合,使两个三角形折叠,从而达到简化三角形网格的效果。如图 9 所示,要简化△N1M1M2 和△N2M1M2,将顶点 M1 和 M2 重合至顶点 M,则边 L1、L2、L3 和 L4 也重合至边 L。

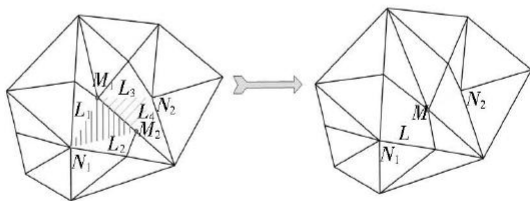


图 9 折叠三角形算法

对比两种算法,折叠三角形算法可以同时简化两个三角形,运用此方法,迭代简单,运行速度快,采用这种算法进行复杂模型的简化。

本算法在 OpenGL 平台上实现,基本数据结构由 Vertex, Triangle, Process 三个结构体组成,如下所示:

```
struct Vertex
{
    GLfloat k[]=new GLfloat[3]; //顶点的三维坐标
```

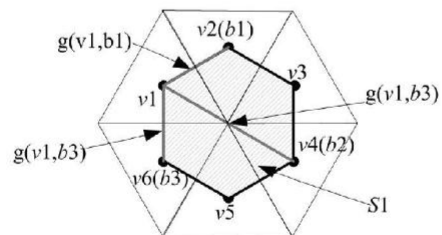
```
GLint vId; //顶点的索引号
}
struct Triangle
{
    GLint tId; //三角网格的索引号
    Bool t_flag; //标识该三角网格是否被删除
    GLTVector3 t_normal; //三角网格法向量
}
struct Process
{
    GLint process[]=new GLint[4]; //顺序存储了一次简化操作中除去的点、三角网格的索引号
```

3 计算顶点的函数值

模型的基本单元是三角形,我们计算函数值也是以三角形为单元计算的,但是为了方便,我以一顶点代替三角形,这个顶点取三角形的重心。在这一节中,首先定义顶点的连续函数,顶点的连续函数定义如下^[12]:

$$\mu(v) = \sum_{i=1}^n [g(v, b_i) \cdot area(b_i)] \tag{1}$$

如图 10 所示,六个三角网格是一个连通分量, v 为三维模型的顶点(三角网格的重心),是这个连通分量上的基本顶点, g(v, b_i) 是顶点 v 与 b_i 之间最短距离, area(b_i) 是基本顶点与周围各个顶点围成的面积,即为图中的面积 S₁。



$$area(b1)=area(b2)=area(b3)=S1$$

图 10 顶点函数的参数关系图

定义完连续函数后,接着要求每个顶点的函数值,分以下两步进行:1.通过 Dijkstra 算法求取基本点集;2.用公式(1)求得顶点的函数值^[13]。

3.1 求取基本点列

使用 Dijkstra 算法、二叉树 LIST 和基本点列表 B_LIST 来求取连通分量上所有顶点的基本点列表,图 11 列出了算法实现的流程图:

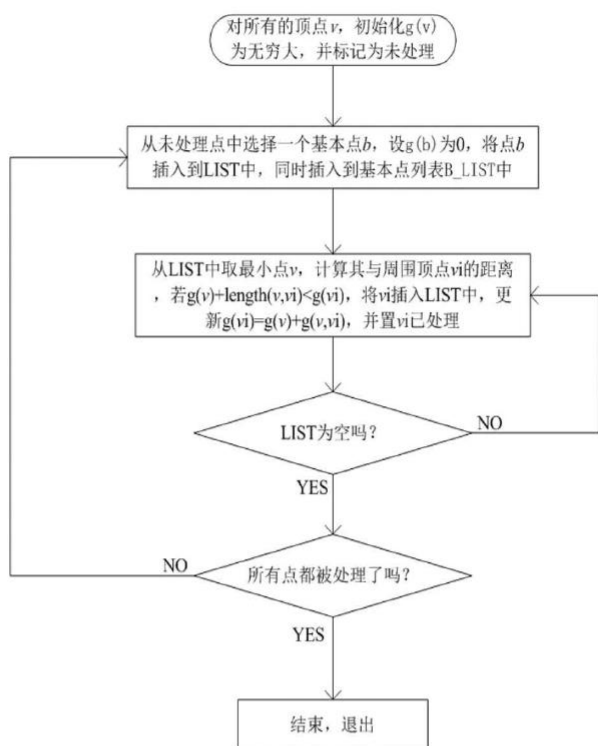


图 11 Dijkstra 算法求取基本点列流程

在这里定义一个阈值 $r = \sqrt{(0.05 \times \text{area}(b_i))}$, 发现基本点列 B_LIST 的大小与阈值 r 的大小有关, 阈值越小, 取得基本点列越大, 计算的骨架图越精确, 相应的时间越长, 效率越低。所以, 要合理取阈值, 让获得的骨架图既相对精确, 执行的效率又高, 综合考虑, 定义阈值 $r = 0.01$ 。

3.2 计算每个顶点的函数值

在获得基本点集后, 就可以使用公式(3-1)来计算每个顶点的函数值了, 计算过程分为以下三步:

1. 遍历所有的顶点, 根据上述的 Dijkstra 算法来计算顶点 v 和每个基本点 b_i 的最短距离 $g(v, b_i)$;
2. 计算每个顶点到各个基本点的最短距离 $g(v, b_i)$ 和基本点面积的积, 并将其累加起来。把最小函数值 $\min(\mu(v))$ 和最大函数值 $\max(\mu(v))$ 记录下来;
3. 利用下列的公式将计算出来的函数值进行归一化^[5]:

$$\mu(v)' = (\mu(v) - \min(\mu(v))) / (\max(\mu(v)) - \min(\mu(v))) \quad (2)$$

将函数值归一化的目的是把函数值都分配在 (0, 1) 的区间内, 以便于更方便地将区间分成若干部分, 为每个三角网格的函数值找到其对应的区间。

3.3 生成模型骨架图

将函数值区间 (0, 1) 分成若干个子区间, 分成的子区间数越多, 计算的骨架图越接近模型拓扑结构, 相应的效率越低。将计算得到的各个网格根据其函数值分配到每个子区间内^[2]。同时根据网格的连接关系, 将各个子区间内的网格合并成相应的子连通分量。最后通过下式计算每个子连通分量的合成节点^[3]:

$$V_p = \sum_{i=0}^n V_i \cdot \text{area}(S_i) / \sum_{i=0}^n \text{area}(S_i) \quad (3)$$

其中: V_p 是要计算的合成节点, V_i 是子连通分量中网格的顶点, $\text{area}(S_i)$ 是子连通分量中每个网格的面积。

最后根据子连通分量的邻接关系添加节点之间的连线, 这些合成节点和它们的连线构成了模型的骨架图。

4 实验过程及结果

运用 VTK, 在 vs2010 平台上编程对该算法进行实现。本次试验分别在工业零件(工字钢)和圆环模型上进行算法验证。

首先在 Solidworks 三维建模软件上分别画出了一个工字钢模型和圆环模型, 保存为 STL 格式文件。

读取 STL 网格文件, 使用循环结构对每个网格和顶点进行遍历, 读取每个网格和网格中的顶点。接着按照前面所述的算法生成连续函数、计算每个网格的函数值和骨架图的生成。经过运行程序, 可以得出各个连通分量的合成节点和连接节点的弧。工字钢的节点有四个, 对应的索引值分别是 0, 73, 76 和 77, 连接弧有 3 条; 圆环的节点有六个, 对应的索引值分别是 0, 476, 711, 1091, 2065, 2445, 连接弧有 6 条。构成的骨架图如图 12 所示:

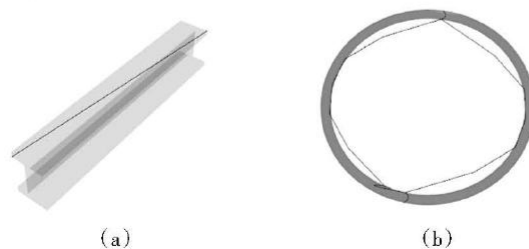


图 12 模型的骨架图

图 12(a) 是工字钢的骨架图, 图 12(b) 是圆环的骨架图。

(下转第 154 页)

机数目过多造成的资源浪费的问题。实验结果表明,提出的方法在使用支持向量回归及多层感知器算法时能有效预测出主机数目。

参考文献

- [1] MENASCE D A. Load testing of web sites [J]. IEEE Internet Computing, 2002, 6(4): 70-74.
- [2] Load Runner [EB/OL](2018-10-23)[2019-05-04]. <https://www.microfocus.com/zh-cn/products/loadrunner-load-testing/overview>.
- [3] Apache Jmeter [EB/OL](2018-12-21)[2019-05-04]. <https://jmeter.apache.org>
- [4] ALI A, BADR N. Performance testing as a service for web applications[C]//2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS). IEEE,

2015: 356-361.

- [5] BANZAI T, KOIZUMI H, KANBAYASHI R, et al. D-cloud: design of a software testing environment for reliable distributed systems using cloud computing technology [C]//2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. IEEE, 2010: 631-636.
- [6] 阿里云性能测试工具 PTS [EB/OL](2019-01-21)[2019-05-04]. <https://www.aliyun.com/product/pts>
- [7] 压测大师 LM [EB/OL](2019-01-21)[2019-05-04]. <https://cloud.tencent.com/product/lmRapid> elasticity and the cloud.
- [8] 胡越, 罗东阳, 花奎, 等. 关于深度学习的综述与讨论[J]. 智能系统学报, 2019, 14(01): 1-19.
- [9] SMOLA A J, SCHÖLKOPF B. A tutorial on support vector regression[J]. Statistics and Computing, 2004, 14(3): 199-222.
- [10] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning internal representations by error propagation[M]// Neurocomputing: foundations of research. MIT Press, 1988.

(上接第 149 页)

5 结论

实现了一种三角网格模型骨架提取的算法。首先读取模型文件,对于三角网格比较复杂的模型,还要简化模型,遍历网格及其顶点,构造连续函数,使用 Dijkstra 算法提取基本点,然后求得所有顶点的函数值。按照每个网格的函数值分配函数区间,在每个子连通分量中遍历网格寻找合成节点,最后连点成弧,生成模型骨架。此方法生成的骨架效率高,基本基于模型的拓扑结构,但对于网格比较复杂的模型,效果不是太好,这就需要后续的不断完善。提取模型的骨架图对于工业上的智能视觉识别和机器人抓取技术非常有意义,抽取的骨架保留了原来模型的姿态,连通性,也可用于三维模型的检索。

参考文献

- [1] MARAGOS P A, SCHAFFER R W. Morphological skeleton representation and coding of binary image [J]. IEEE Transaction on Acoustics, Speech, and Signal Processing, (S0096-3518), 1986, 34(5): 1228-1244.

- [2] DEY T K, ZHAO W. Approximating the medial axis from the voronoi diagram with a convergence guarantee [J]. Algorithmica, (S0178-4617), 2001, 38(1): 179-200.
- [3] 黄坤武, 唐杰, 武港山, 等. 改进的多分辨率 Reeb 图骨架抽取算法[J]. 计算机应用, 2006(02): 415-418.
- [4] 黄坤武, 唐杰. 针对面片的 Reeb 图骨架抽取算法[J]. 系统仿真学报, 2006(S1): 52-56.
- [5] HILAGA M, SHINAGAWA Y. Topology matching for fully automatic similarity estimation of 3D shapes. International Conference on Computer Graphics and Interactive Techniques [C]// Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. ACM, 2001: 203-212.
- [6] ALEOTTI J, CASELLI S. A 3D shape segmentation approach for robot grasping by parts [J]. Robotics and Autonomous Systems, 2011, 60(3): 358-366.
- [7] FARIA D R, TRINDADE P, LOBO J, et al. Knowledge-based reasoning from human grasp demonstrations for robot grasp synthesis [J]. Robotics and Autonomous Systems, 2014, 62(6): 794-817.
- [8] 宫法明, 陈依心, 李广丽. 一种改进的基于特征点求解的骨架提取算法[J]. 微型电脑应用, 2010, 26(04): 53-55.
- [9] BERRETTI S, BIMBO A D, PALA P. 3D Mesh decomposition using Reeb graphs[J]. Elsevier B.V., 2009, 27(10): 1540-1554.
- [10] 车武军, 杨勋年. 动态骨架算法[J]. 软件学报, (S1000-9825), 2003, 14(4): 818-821.
- [11] 姚童仙. LOD 模型简化算法的研究与实现[D]. 兰州: 兰州交通大学, 2014.