

工业 CT 断层图像自动生成 STL 文件的方法^{*})

段黎明 林海 吴朝明 刘 炉

(重庆大学光电技术及系统教育部重点实验室 ICT 研究中心 重庆 400044)

摘 要 为了使工业 CT(computed tomography)扫描获得的工件断层图像格式转换为快速原型设备所接受的数据格式,研究了工业 CT 断层图像数据格式自动生成 STL(Stereolithography)文件格式的方法,并用 VC++ 开发了软件。实验结果表明,效果较好。

关键词 STL 文件,三角网格化,逆向工程,工业 CT

Method of Automatic Generation of STL Files from Industrial CT Slicing Image

DUAN Li-Ming LIN Hai WU Zhao-Ming LIU Lu

(ICT Research Center, Key Laboratory of Optoelectronic Technology and System of the Education Ministry of China, Chongqing University, Chongqing 400044)

Abstract To make the slicing image of industrial CT can be transformed to the data that can be accepted by rapid prototype machine. The method of automatic generation of STL files from industrial CT slicing image is researched. The software is designed with VC++. The result is good.

Keywords STL file, Triangulation, Reverse engineering, Industrial computed tomography

1 引言

随着快速原型制造(Rapid Prototyping, RP)技术的逐渐成熟和工业 CT 无损检测技术的发展,面向 RP 的工业 CT 断层图像反求技术已经成为当前反求领域的一个研究热点^[1]。

面向 RP 的工业 CT 断层图像反求技术研究,重点在于由工业 CT 断层图像数据格式生成快速原型设备所能接受的数据格式。目前,其方法主要有两种:一是由 CAD 软件读取经图像处理得到的轮廓数据,用人工交互的方法重构出样件的 CAD 实体模型,再通过相应的 CAD 软件进行格式转换得到^[2];二是通过开发专用软件实现由断层图像直接生成^[3,4]。前一种方法可对模型进行修改和再设计,但是相对周期较长;而第二种方法则缩短了开发时间,可用于产品的直接仿制^[5]。

本文对第二种方法进行了研究,开发了基于工业 CT 断层图像的 STL 文件自动生成软件。使用该软件生成的 STL 文件可直接输入快速原型设备进行原型制造,实验结果表明,效果较好。

2 基于工业 CT 断层图像的 STL 文件自动生成

由工业 CT 断层图像直接生成 STL 文件,其总体思路如图 1 所示。(1)对 CT 扫描得到的断层图像进行图像处理;(2)轮廓对应;(3)轮廓三角网格化,生成 STL 文件。

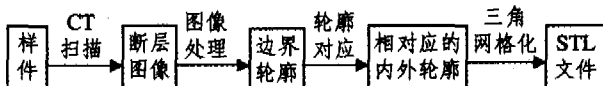


图 1 总体流程图

2.1 图像处理

首先对工业 CT 扫描得到的断层图像进行处理,其目的

是为了得到单像素宽度的边界数据,主要包括低通滤波、边界提取、细化三个步骤。图像处理后的边界轮廓是以离散的像素点来表示的,为轮廓对应及三角网格化做准备。

2.2 轮廓对应

轮廓对应的目的是识别每一断层的内轮廓和外轮廓,并将各断层图像上不同轮廓进行适当的组合,使其能够构成有意义的物体。

首先进行包容检测,即判断轮廓为内轮廓还是为外轮廓。设某一断层上的任意两个轮廓 C_i 和 C_{i+1} ,其最小矩形包围盒分别为 B_i 和 B_{i+1} 。若 B_i 和 B_{i+1} 相互不包含,则 C_i 和 C_{i+1} 不存在包含关系,同为内轮廓或同为外轮廓;若 B_i 包含 B_{i+1} ,进一步判断轮廓 C_i 的各条边是否都在 C_{i+1} 的内部,若是,则 C_i 为内轮廓, C_{i+1} 为外轮廓。包容检测后,标记轮廓为内轮廓或外轮廓。然后,根据在相邻断层上属于同一分支的轮廓之间重叠面积大于属于不同分支的轮廓之间重叠面积准则,确定内、外轮廓相应的对应关系。以“组”来记录不同的分支,并将不同的分支标记为不同组号:内轮廓组或外轮廓组。

2.3 三角网格化

边界轮廓的三角网格化就是将相邻轮廓之间各“组”内的轮廓用三角面片连接。本文采用一种基于 Delaunay 三角剖分思想的算法,按照剖分三角面片集中的最小内角在整体上尽可能大的优化准则进行剖分。其步骤如下:

设相邻层中相对应的轮廓点集分别为 $P = \{p_i | 0 \leq i \leq m\}$ 和 $Q = \{q_j | 0 \leq j \leq n\}$, ($m \geq 3, n \geq 3$),均以顺时针顺序排列,其中 $p_m = p_0, q_n = q_0$ 。设 P 为上层点集, Q 为下层点集。

(1)构建起始三角面片

1)连接下层轮廓的任意相邻两点作为起始三角面片的底边,设为 $q_0 q_1$,如图 2(a)所示;

2)计算上层轮廓上到 q_0 和 q_1 两点距离和最小的点,设

^{*}重庆市科委自然科学基金计划资助项目(CSTC2006BB2138)。段黎明 副教授,工学博士,研究方向为工业 CT 技术及应用,逆向工程。

为 p_i ;

3) 以点 p_i 及其左右临近点 p_{i-1} 和 p_{i+1} 为顶点, 组成三个三角面片 $p_{i+1}q_0q_1$ 、 $p_iq_0q_1$ 和 $p_{i-1}q_0q_1$, 比较并选择其中最小内角为最大的三角面片, 将其作为起始三角面片, 将起始三角面片的左边作为工作边。假定选择 $p_{i+1}q_0q_1$ 为起始三角面片, 则 $p_{i+1}q_1$ 为工作边。

(2) 将上层轮廓点集中的点重新排序。将点集 p 中的前 m 个点以 p_{i+1} 为第 0 点, 按照顺时针的顺序重新排序, 并将 p_{i+1} 定为新点集中的第 m 点, 如图 2(b)。重新排序后的点集

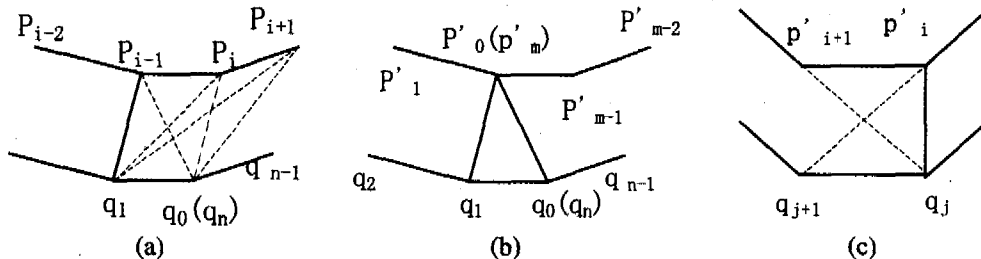


图 2 最小内角最大法三角网格化

(5) 判断工作边 $p'_i q_j$ 的某一点是否与初始三角面片的一个顶点重合

1) 若 $i=m, j \neq n$, 即工作边一个点与初始三角面片位于上层轮廓上的顶点重合, 则连接 $p'_i q_{j+1}, p'_i q_{j+2}, \dots, p'_i q_{n-1}$, 令 $j=n, p'_i q_j$ 为新工作边, 转(6);

2) 若 $j \neq m, j=n$, 即工作边一个点与初始三角面片位于下层轮廓上的顶点重合, 则连接 $p'_{i+1} q_j, p'_{i+2} q_j, \dots, p'_{m-1} q_j$ 为新工作边, 转(6)。

(6) 判断工作边 $p'_i q_j$ 是否与初始三角面片的边重合。若 $i=m, j=n$, 即工作边 $p'_i q_j$ 与初始三角面片的边重合, 则剖分结束; 否则, 转(4)。

2.4 自动生成 STL 文件

(1) STL 文件格式

STL 文件格式是由美国 3D system 公司于 1988 年制定的一个接口协议, 是一种为快速原型制造技术服务的三维图形文件格式, 是 CAD 实体模型与快速成型设备进行数据交换的准标准, 几乎所有类型的快速成型系统都采用 STL 数据格式^[7]。STL 文件由许多三角面片的定义组成, 每个三角面片的定义包括它的法矢量和各个顶点的三维坐标。STL 文件记载了组成 STL 实体模型的所有三角面片, 它有二进制和 ASCII 两种表示方法。本文采用 ASCII 来写 STL 文件, 其语法如下:

```

solid file name //STL 文件的开始标志及此文件的名称
facet normal //第一个三角面片的外法矢量的三个分量值
  outer loop
    vertex A //第一个三角面片的第一个顶点的三维坐标
    vertex B //第一个三角面片的第二个顶点的三维坐标
    vertex C //第一个三角面片的第三个顶点的三维坐标
  endloop //三角面片顶点定义结束
endfacet //第一个三角面片定义结束
..... //与第一个三角面片结构一样的其余各三角面片的数据
end solid file name //实体结束标志
    
```

根据接口协议, STL 文件的格式要求如下:

1) 法矢量向外: 每个三角面片都是实体模型内部与外部的分界面。对于存在内表面的实体模型, 为了生成正确的内外部结构, 协议规定: 组成 STL 数据模型的每个三角面片的法矢量的方向是朝外的。

2) 顶点规则: 相邻的两个三角面片只有一条公共边, 也就是相邻的两个三角形有且仅共用两个点。

设为 $p'_i = \{p'_i | 0 \leq i \leq m\}$, 初始三角面片为 $p'_0 q_0 q_1$ 或 $p'_m q_0 q_1$ 。

(3) 设定 $p'_0 q_1$ 为起始工作边, 令 $i=0, j=1$

(4) 构建过工作边 $p'_0 q_1$ 的三角面片, 并选择新的工作边。以 $p'_i q_j$ 为边组成的三角面片可能为 $p'_{i+1} p'_i q_j$ 和 $p'_i q_{j+1} q_j$ 如图 2(c)。若 $p'_{i+1} p'_i q_j$ 的最小内角大于 $p'_i q_{j+1} q_j$ 的最小内角, 则连接 $p'_{i+1} q_j$ 以其作为新工作边, $i=i+1$, 否则连接 $p'_i q_{j+1}$, 以其作为新工作边, $j=j+1$ 。

(2) 自动生成 STL 文件的实现

1) 定义数据结构。用 VC++ 实现上述三角网格化算法, 为了表示轮廓截面上点的三维坐标, 需要定义一个表示三维坐标的结构体。

```
struct point{float x; float y; float z;};
```

单层轮廓上所有点的集合, 可以用一个单向链表或者循环链表来表示, 结构体定义为:

```
struct link{struct point p; struct link * next;};
```

对于在每相邻两层轮廓之间构建的三角面片的集合可以采用邻接表来表示, 首先定义表示三角面片的结构体。

```
struct triangle{struct point vector; struct point p1; struct point p2; struct point p3;};
```

然后定义邻接表的结构:

```
struct trilinear{struct triangle tri; struct trilinear * next;}; trilinearlink [N-1];
```

式中: N 为采集到的工业 CT 断层图像的个数; 单向链表 $trilinearlink [i]$ 表示的是在第 i 层和第 $i+1$ 层之间构建的所有三角面片的集合。

2) 相邻层轮廓三角网格化。为了生成相邻两层轮廓间的三角面片, 在程序中建立了一个循环, 将相邻 i 和 $i+1$ 层的轮廓原始数据分别读入两个循环链表。

根据 Delaunay 三角剖分思想进行三角网格化, 得到的三角面片数据写入邻接表 $trilinearlink [i]$ 。为确保三角面片的法矢量朝外, 组成三角面片的三个点按照逆时针顺序写入 STL 文件。

3) 判断轮廓是否全部三角网格化。若层轮廓未全部三角网格化, 则转到(2)循环执行, 否则, 转(4)。

4) 生成 STL 文件。最后, 将生成的三角面片按 STL 文件所规定的格式写入 STL 文件。

采用上述算法生成的 STL 文件可以满足接口协议规定。

3 实例验证

一个样件经过工业 CT 扫描后得到 28 层断层图像, 每层间距为 0.4mm。使用开发的软件对该样件的断层图像进行

(下转第 218 页)

A 表示。

由表 1 可知,动态消减算法没有静态消减与动态消减组合算法的检验时间理想。因为在文[4]中提出的静态消减算法在模型检验的初期通过静态分析删除了大量的无关原子公式,使得可达性分析的速度加快,虽然空间消耗与前者接近,但是检验时间通常比前者快,所以在检验过程中我们偏向于采用两者结合的优化方法。同时,在验证的过程中,我们发现如果将阈值设置得比较大的时候,检验过程所需要的时间会有所下降。

结论 目前有许多针对扩展时间自动机的符号状态来减少状态空间的优化算法。这些方法的特点是基于静态的统计信息。本文提出的算法是动态的算法,它的核心思想是在检验过程中根据相邻符号状态的前驱与后继之间的依赖关系获得可以消去的无关原子公式,然后通过删除这些静态算法不能发现的无关原子公式来扩展符号状态,从而降低模型检验过程中对内存的需求。案例研究的实验数据表明,本文提出的算法能够在状态空间超过阈值时有效地压缩状态空间,但是因为要获得所有的符号状态原子公式间的依赖关系以及合并有包含关系的符号状态,需要一定的计算。一般来说这个算法对于当前的存储空间小于状态空间、必须进行状态空间约减的情况非常适用。

参考文献

1 Clarke E M Jr, Grumberg O, Peled D A. Model Checking. Cambridge, MIT Press, 2000. 1~26

2 Alur R, Dill D L. A theory of timed automata. Theoretical Computer Science, 1994,126: 183~235
 3 Daws C, Yovine S. Reducing the number of clock variables of timed automata. In: Burns A, ed. Proc. of the 17th IEEE Real Time Systems Symposium (RTSS' 96). Washington, DC, USA; IEEE Computer Society Press, 1996. 73~81
 4 ZHAO Jianhua, LI Xuandong, ZHENG Tao, et al. Removing Irrelevant Atomic Formulas for Checking Timed Automata Efficiently. In: Niebert P, ed. Proc of First International Workshop on Formal Modeling and Analysis of Timed Systems (FORMATS). LNCS 2791. Marseille, France; Springer-Verlag, 2003. 34~45
 5 Behrmann G, Bouyer P, Larsen K G, et al. Lower and Upper Bounds in Zone Based Abstractions of Timed Automata. In: Jensen K, ed. Proc. of 10th Int Conf Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04). LNCS 2988. Barcelona, Spain; Springer-Verlag, 2004. 312~326
 6 Bengtsson J, Yi Wang. Timed Automata; Semantics, Algorithms and Tools. In: Reisig W, ed. Lecture Notes on Concurrency and Petri Nets 2003. LNCS 3098. Eichstätt, Germany; Springer-Verlag, 2004. 87~124
 7 Behrmann G, David A, Larsen K G, et al. Uppaal - Present and Future. In: Proc. of the 40th IEEE Conference on Decision and Control (CDC'2001). Orlando, Florida, USA; IEEE Computer Society Press, 2001,3: 2881~2886
 8 Daws C, Olivero A, Tripakis S, et al. The tool Kronos. In: Alur R, ed. DIMACS Workshop on Verification and Control of Hybrid Systems. Hybrid System III; Verification and Control, LNCS 1066. Berlin; Springer-Verlag, 1996. 208~219
 9 Wang F. RED: Model-Checker for Timed Automata with Clock-Restriction Diagram. In: Proc. of Workshop on Real-Time Tools, Aalborg University, Denmark; [Technical Report], 2001-014, ISSN 1404-3203, Department of Information Technology, Uppsala University
 10 Havelund K, Skou A, Larsen K G, et al. Formal Modelling and Analysis of an Audio/Video Protocol; An Industrial Case Study Using Uppaal. In: Lin KJ, ed. Proc. of 18th IEEE Real-Time Systems Symposium. San Francisco, CA, USA; IEEE Computer Society Press, 1997. 2~13

(上接第 202 页)

图像处理、三角网格化、自动生成 STL(.ast)文件。图 3 为用

本文软件对该断层图像各阶段的处理结果在反求软件 Imageware 中的显示效果,图 4 为自动生成的 STL 文件。

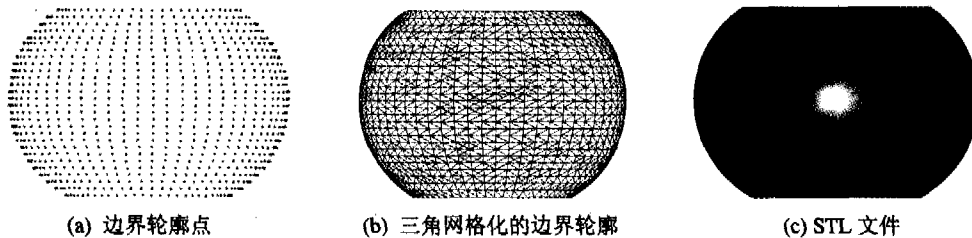


图 3

```
solid stl-1
facet normal -0.867126 -0.015804 0.497838
  outer loop
    vertex 0.000001 56.576290 -7.500054
    vertex 0.000001 52.221680 -10.000140
    vertex -3.588281 56.462379 -7.500054
  endloop
endfacet
.....
facet normal -0.174225 0.031245 0.984210
  outer loop
    vertex 5.360947 84.349602 35.000271
    vertex 0.000001 84.077232 32.500229
    vertex 0.000001 84.519791 35.000271
  endloop
endfacet
endsolid stl-1
```

图 4 生成的 STL 格式文件

结论 本文实现了由工业 CT 断层图像到 STL 文件的自动生成,实例验证了其正确性。与通过 CAD 软件重构出样件的实体模型,再将其转换成 STL 文件的方法相比,该方法缩短了开发时间,充分发挥了快速原型制造的优越性。

从理论上讲,构造的三角面片越多,生成的 STL 文件与

样件的符合程度就会越高,但是,这会使运算量变大,对计算机的运算速度提出更高的要求。此外,生成的 STL 文件中三角面片过多,亦会给后继处理带来一定的困难,如何在保持精度和数据精简之间取得平衡,这将是下一步需要继续深入研究的问题。

参考文献

1 Lee K H, Woo H. Direct integration of reverse engineering and rapid prototyping [J]. Computer & Industry Engineer, 2000, 38: 21~28
 2 王霄. 逆向工程技术及应用[M]. 北京:化学工业出版社, 2004. 103~111
 3 Liu S, Ma W. Seed-growing segmentation of 3-Dsurface from CT-contourdata [J]. Computer-Aided Design, 1999, 31(8):517~536
 4 Park H, Kim K. Smooth surface approximation to serial cross-sections [J]. Computer-Aided Design, 1996,28(12):995~1005
 5 李发致. 模具先进制造技术[M]. 北京:机械工业出版社, 2003. 208~242
 6 纪凤成,秦绪佳,等. 一种基于 CT 图像反求技术的实体几何造型方法[J]. 机械科学技术, 2002, 21(1),165~168
 7 金涛,董水光,等. 逆向工程技术[M]. 北京:机械工业出版社, 2003. 271~277