

基于工业 CT 图像的自适应三维网格模型重建

黎玲^{1,2}, 金恒^{1,2}, 刘杰^{1,3}, 龙超^{1,3}, 何云勇^{1,3}, 李中明^{1,2}, 段黎明^{1,2*}¹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;²重庆大学机械与运载工程学院, 重庆 400044;³重庆大学光电工程学院, 重庆 400044

摘要 针对 X 射线扫描得到的工业 CT 图像重建三角网格模型存在尖锐特征丢失、狭长三角形和三角面片数量多等问题, 提出一种自适应三维网格模型重建算法。首先对图像进行预处理; 其次采用八叉树结构确定体元; 然后利用二次误差函数(QEF)构建自适应八叉树; 最后剖分四边形生成三角网格。使用立方体数据和两组实际扫描的 CT 数据对所提算法的性能进行验证分析, 实验结果表明: 所提算法在简化网格的同时仍能保持物体的尖锐特征, 减少了狭长三角形的数量。利用所提算法生成三角网格模型的简化率可达 90%, 简化后网格质量大于 0.3 的三角网格平均占比为 99%, 有效地提高了由工业 CT 图像重建三角网格模型的质量。

关键词 X 射线光学; 工业 CT 图像; 三维重建; 三角网格; 自适应八叉树

中图分类号 TP751

文献标志码 A

DOI: 10.3788/AOS221477

1 引言

三角网格模型已被广泛应用于增材制造^[1]、逆向设计^[2]、有限元分析^[3]等领域。目前, 基于工业 CT 图像^[4]的三维网格模型重建算法中, 使用最广泛的是移动立方体(MC)算法^[5], 但 MC 算法存在两个问题: 1) 只利用交点的位置信息, 无法保持尖锐特征; 2) 重建产生的三角面片数据规模大, 严重影响模型的实时显示、保存和传输。

针对 MC 算法无法保持尖锐特征的问题, Kobbelt 等^[6]提出了扩展 MC(EMC)算法, 通过能量方程拟合法向, 能够产生尖锐特征, 但需要显式地检测尖锐特征的存在; 袁小翠等^[7]提出一种基于曲面法向估计的算法, 但参数的设定需要根据经验, 实际操作的可行性低; 陈甜甜等^[8]提出一种保持尖锐特征的三维网格模型重建算法, 但会进一步增加三角网格的数量。

针对 MC 算法产生的三角面片数据规模大的问题, 主要有两类解决方案: 第一类是对生成的网格模型使用网格简化算法, 如顶点聚类、顶点删除、三角形折叠算法以及广泛应用的边收缩网格简化算法^[9-12], 但这类方法属于事后补救的方法, 处理费时, 并且建立辅助结构会占据大量存储空间; 第二类是直接采取避免生成较多三角面片的表面重建方法。与第一类方法相比, 第二类方法利用了图像数据的特点, 效率更高。

Vignoles 等^[13]提出了简化 MC(SMC)算法, 将插值点选在体元边的中点, 然后合并共面的三角形, 以此来减少三角面片数量, 但网格的简化率有限; Shu 等^[14]提出了自适应 MC(AMC)算法, 通过不断划分子立方体, 直至子立方体内的三角面片相对等值面的误差在一定范围内, 实现了用较少的三角面片来表示等值面, 但产生了较多狭长三角形; Shekhar 等^[15]在体元合并中使用八叉树结构评估误差, 保证误差在给定阈值之内, 但不同大小的八叉树节点交界处会产生孔洞, 需要额外修补孔洞; 熊邦书等^[16]提出一种变分辨率的曲面重建算法, 产生立方体的大小可以自适应于曲面曲率的大小, 但需要拼接立方体间产生的缝隙; 徐雷等^[17]提出一种基于八叉树的简化构型算法, 通过合并共面构型的体元实现网格简化, 但共面构型的条件限制了简化率的进一步提升。

以上研究虽对 MC 算法做了一些改进, 但处理过程繁琐。Ju 等^[18]提出了双轮廓(DC)算法, 与 MC 算法相比, DC 算法不存在二义性, 能够再现物体的尖锐特征, 但该算法目前主要用来与其他算法结合生成有限元模型^[19]、绘制隐式曲面^[20]以及简化三角网格模型^[21]。本文将 DC 算法引入工业 CT 图像的三维网格模型重建中, 提出一种自适应的三维网格重建算法, 通过图像预处理、创建八叉树、简化八叉树、生成三角网格 4 个步骤, 实现了自适应三维网格模型重建。本

收稿日期: 2022-07-14; 修回日期: 2022-08-17; 录用日期: 2022-08-29; 网络首发日期: 2022-09-10

基金项目: 国家自然科学基金(52075057)

通信作者: *duanliming163@163.com

文算法可根据物体局部特征,自适应地不同大小的体元内提取等值面,既能保留重建模型的细节特征,又能大量减少模型三角面片的数量,同时避免狭长三角形的产生,提升了三角网格模型的重建质量。

2 图像预处理

由于扫描过程中受到各种因素的干扰,工业 CT 图像通常包含大量的噪声信息,这些噪声会严重影响模型的重建质量^[22],同时等值面的值也会影响模型的

重建效果^[23]。因此,为保证重建得到的网格模型质量,需要先对图像降噪,再确定等值面的值。

2.1 降噪

常见的图像降噪算法包括均值滤波、高斯滤波、双边滤波、中值滤波、非局部均值滤波等,针对不同类型的噪声,这些算法可能有不同的降噪效果。为尽可能减少噪声对重建结果的影响,分别用这几种算法对工业 CT 图像进行处理,降噪结果如图 1 所示。

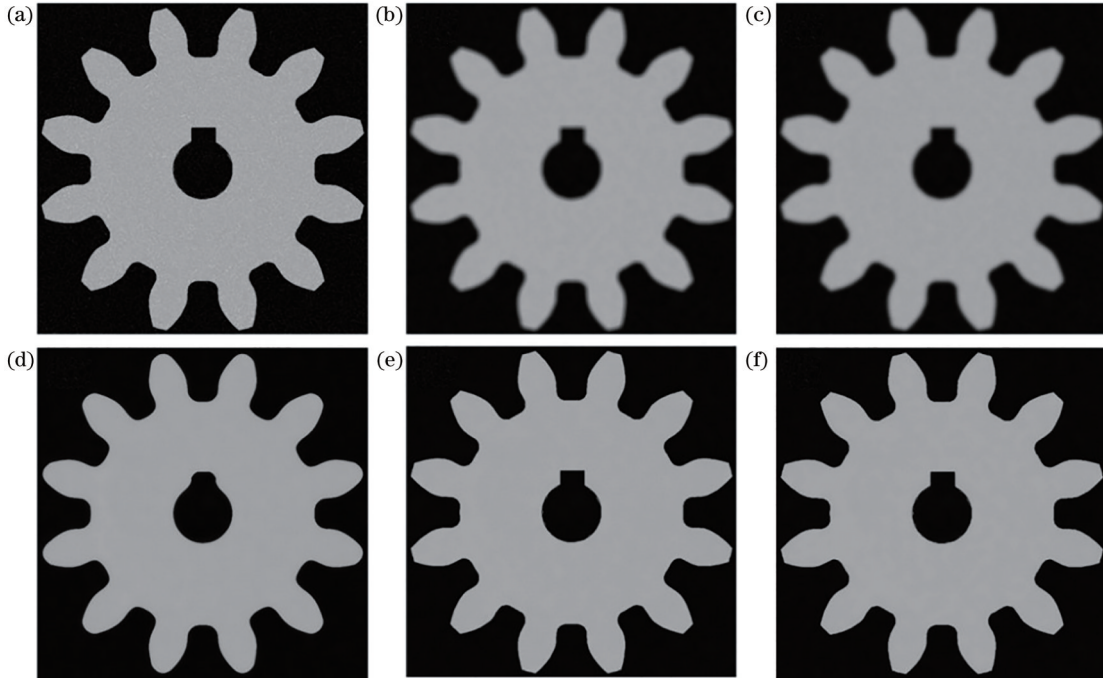


图 1 不同算法的图像降噪结果。(a)原图;(b)均值滤波;(c)高斯滤波;(d)中值滤波;(e)非局部均值滤波;(f)双边滤波

Fig. 1 Results of image denoising by different algorithms. (a) Original image; (b) mean filtering; (c) Gaussian filtering; (d) median filtering; (e) nonlocal mean filtering; (f) bilateral filtering

从图 1 可以看出:均值滤波和高斯滤波破坏了图像的细节信息,使图像变得模糊;中值滤波效果较好,但引起了图像边缘变形;非局部均值滤波和双边滤波处理效果好,但非局部均值滤波复杂度高。综合比较这几种算法的图像降噪效果可知,双边滤波可以在有效去除噪声的同时保存图像的边缘细节,因此本文选择双边滤波来对工业 CT 图像降噪。

2.2 确定等值面的值

通过工业 CT 扫描需要重建表面模型的对象多为零件级,如齿轮、引擎缸体、阀体等,这些零件的材质均匀,因此其灰度直方图一般为双峰结构。根据这个特点,使用最大类间方差^[24](OTSU)算法来确定图像的灰度阈值。

对于一幅图像 I ,所有灰度值为 i 的像素点出现的概率 P_i 为

$$P_i = N_i / N, \quad (1)$$

式中: N_i 表示灰度值为 i 的像素点数, $i \in [0, L-1]$, L 为图像灰度级; N 表示图像像素点总数。

假设像素点按灰度阈值 T 划分为目标区域 A_t 和背景区域 A_b , A_t 包括灰度级为 $[0, T]$ 的像素点, A_b 包括灰度级为 $[T+1, L-1]$ 的像素点。

按照如下步骤确定图像的灰度阈值:

1) 分别计算目标区域像素点的比例 k_t 和背景区域像素点的比例 k_b , 其中 $k_t = \sum_{i=0}^T P_i$, $k_b = \sum_{i=T+1}^{L-1} P_i = 1 - k_t$;

2) 计算目标灰度均值 μ_t 、背景灰度均值 μ_b 和总体灰度均值 μ ;

3) 计算类间方差 δ^2 : $\delta^2 = k_t(\mu_t - \mu)^2 + k_b(\mu_b - \mu)^2$;

4) 计算灰度阈值 T_h : $T_h = \operatorname{argmax}_{0 \leq T \leq L-1} \delta^2$ 。

对于一组二维序列 CT 图像,利用 OTSU 算法获取每一幅图像的灰度阈值。因为众数能够较好地反映一组数据的集中情况,所以取这组阈值数据的众数作为等值面的值。这种确定等值面的值的方法简单实用,可以避免因手动选择等值面的值造成重建效果不

好的问题。

3 创建八叉树

八叉树^[25]作为三维重建算法中一种重要的数据结构,能够用于表示隐函数^[26]和自适应提取等值面^[27],在不同的应用场合中,八叉树的构建方式可能有所不同。常规八叉树子节点的个数要么为 0,要么为 8,但这样会造成存储空间的浪费。本文创建的八叉树与常规八叉树不同,只存储边界体元,因此子节点的个数可以为 0~8 的任意值。根据工业 CT 图像的数据特点,本文提出一种适用于工业 CT 图像的八叉树构建方法:首先给出体元的定义及分类;然后以自顶向下不断递归的方式创建八叉树。

3.1 体元的定义及分类

由工业 CT 扫描得到二维序列图像属于离散的三维空间规则数据场^[28]的一种。本文将体元定义为相邻两幅 CT 图像的 8 个相邻像素点构成的立方体,如图 2 所示,CT 图像的像素点构成了体元的顶点,且每个顶点都有其对应的坐标和灰度值。

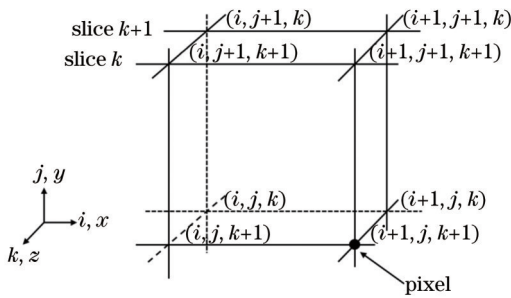


图 2 体元示意图

Fig. 2 Schematic of a voxel

根据式(2)确定体元顶点的状态,即

$$S_{\text{corner}} = \begin{cases} 1, & g \geq V_{\text{iso}} \\ 0, & g < V_{\text{iso}} \end{cases}, \quad (2)$$

式中: S_{corner} 表示顶点状态; g 表示顶点的灰度值; V_{iso} 为等值面的值。若 $S_{\text{corner}} = 1$,表示顶点位于等值面内;若 $S_{\text{corner}} = 0$,表示顶点位于等值面外。

每个体元有 8 个顶点,每个顶点有 0、1 两种状态,所以体元共有 $2^8 = 256$ 种状态值。依据不同的状态值,将体元分为三类:空体元、内部体元和边界体元。空体元的状态值为 0,用“E”标识;内部体元的状态值为 255,用“F”标识;其余状态值的体元都称为边界体元,用“P”标识。

图 2 所示的体元是一个边长为 1 的空间立方体,为叙述方便,将体元的概念扩展,即后文提到的体元都是指空间的一个立方体,边长可不为 1。

3.2 构建八叉树

八叉树根结点的边长必须为 2 的整数次幂,所以在正式创建八叉树之前,需先对输入的图像数据进行处理。

假定输入二维序列 CT 图像的分辨率为 $D_x \times D_y$, 图片的层数为 D_z ,提取数据中最大维度 D_m ,即

$$D_m = \max(D_x, D_y, D_z), \quad (3)$$

并求出不小于 D_m 的最小的 2 的幂次 d ,即

$$d = 2^{\lceil \log_2 D_m \rceil}, \quad (4)$$

式中: $\lceil \cdot \rceil$ 表示向上取整。

递归深度 V_{depth} 为

$$V_{\text{depth}} = \log_2 d. \quad (5)$$

任意八叉树的节点与三维空间的某立方体一一对应,其可以由最小的顶点坐标和节点边长唯一确定。以节点的边长为划分条件,按照如下步骤,创建八叉树:

Step 1:以 $(0, 0, 0)$ 为起点坐标、 d 为边长创建根节点,根节点的深度为 V_{depth} 。

Step 2:将根节点沿着 x, y, z 方向划分为 8 个子节点,如果子节点的边长大于 1,就继续对子节点进行递归分割,每向下递归一层,节点的深度减 1,节点的边长减半。

Step 3:遍历八叉树的所有节点,计算出节点的状态值,删除标识符为 E 或 F 的节点,只存储标识符为 P 的节点。

创建八叉树的流程如图 3 所示。

4 简化八叉树

由于模型表面不同区域的特征不同,对体元的大小要求也不同,平坦区域用较大的体元表示,非平坦区域用较小的体元表示,本文用二次误差函数(QEF)的大小来衡量体元所在区域的平坦程度。首先获取埃尔米特数据^[18](Hermite data);然后构造 QEF,求出特征点的位置;最后通过八叉树叶子节点的聚类来产生一系列具有不同大小的体元,构建出自适应八叉树。

4.1 获取埃尔米特数据

埃尔米特数据可以用来表示体元与等值面交点的位置和交点的单位法向量,通过计算隐式曲面方程或三角网格模型扫描转换可以获得精确的交点和法向量,但 CT 图像由于本身的限制,不能得到准确的数据,所以本文通过线性插值和中心差分法来获取埃尔米特数据。

1) 计算交点的位置

当体元棱边两个端点的状态分别为 0 和 1 时,等值面一定与棱边相交。

体元的棱边分别平行于 x 轴、 y 轴、 z 轴,若棱边两个端点分别为 $P_1(x_1, y_1, z_1)$ 、 $P_2(x_2, y_2, z_2)$,那么交点 $P(x, y, z)$ 的计算方式为

$$\begin{cases} x = x_1 + \frac{(V_{\text{iso}} - g_1)(x_2 - x_1)}{g_2 - g_1} \\ y = y_1 + \frac{(V_{\text{iso}} - g_1)(y_2 - y_1)}{g_2 - g_1} \\ z = z_1 + \frac{(V_{\text{iso}} - g_1)(z_2 - z_1)}{g_2 - g_1} \end{cases}, \quad (6)$$

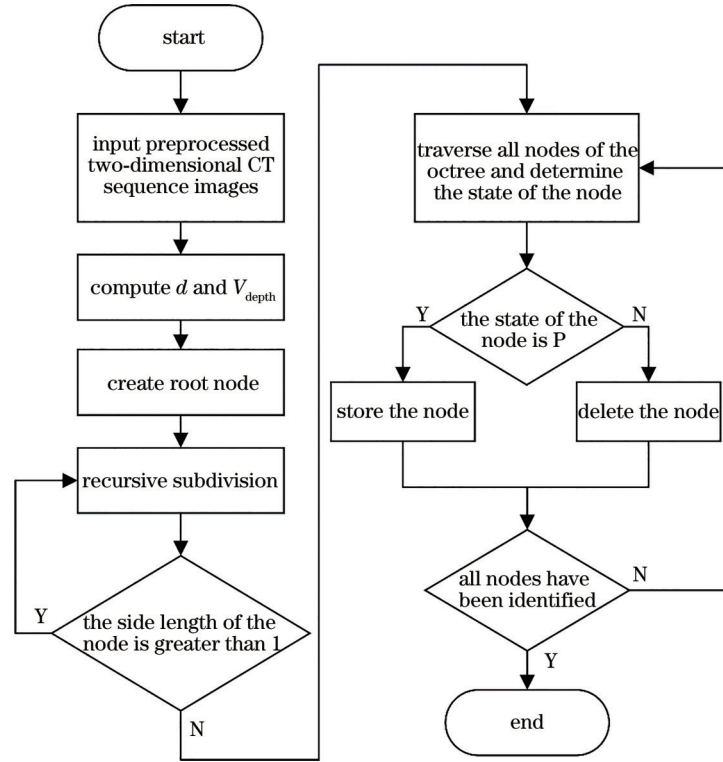


图 3 创建八叉树的流程图

Fig. 3 Flow chart for creating an octree

式中: g_1, g_2 表示两个端点的灰度值。

2) 计算交点的单位法向量

基于灰度梯度的法向量估计来计算交点的法向量。首先,用中心差分法计算体元顶点 $v(i, j, k)$ 上的灰度梯度 $G(G_x, G_y, G_z)$, 即

$$\begin{cases} G_x = \frac{g(i+1, j, k) - g(i-1, j, k)}{2} \\ G_y = \frac{g(i, j+1, k) - g(i, j-1, k)}{2} \\ G_z = \frac{g(i, j, k+1) - g(i, j, k-1)}{2} \end{cases} \quad (7)$$

式中: $g(i, j, k)$ 表示顶点 $v(i, j, k)$ 灰度值; G_x, G_y, G_z 分别表示在 x, y, z 方向的梯度。对 G 归一化后,可以得到体元顶点的单位法向量,再通过线性插值计算交点的法向量。

若棱边两个端点的法向量分别为 $N_1(N_{x1}, N_{y1}, N_{z1}), N_2(N_{x2}, N_{y2}, N_{z2})$, 那么交点的法向量 $N(N_x, N_y, N_z)$ 的计算方式为

$$\begin{cases} N_x = N_{x1} + \frac{(V_{iso} - g_1)(N_{x2} - N_{x1})}{g_2 - g_1} \\ N_y = N_{y1} + \frac{(V_{iso} - g_1)(N_{y2} - N_{y1})}{g_2 - g_1} \\ N_z = N_{z1} + \frac{(V_{iso} - g_1)(N_{z2} - N_{z1})}{g_2 - g_1} \end{cases} \quad (8)$$

4.2 求解特征点

对于每一个与等值面相交的体元,通过最小化

QEF, 在体元的内部产生一个特征点。QEF^[18]表示为

$$E(\alpha) = \sum_i [n_i \cdot (\alpha - p_i)]^2, \quad (9)$$

式中: p_i, n_i 分别表示等值面与体元交点的位置坐标和单位法向量。将式(9)改写成矩阵形式并展开,得到

$$E(\alpha) = (A\alpha - b)^T (A\alpha - b) = \alpha^T A^T A \alpha - 2\alpha^T A^T b + b^T b, \quad (10)$$

式中: 矩阵 A 的行向量为 n_i ; 列向量 b 的每个元素为 $n_i \cdot p_i$; $A^T A$ 是一个 3×3 的对称矩阵; $A^T b$ 是一个 3×1 的列向量; $b^T b$ 为标量。

特征点可以通过求解正规方程得到:

$$A^T A \hat{\alpha} = A^T b, \quad (11)$$

式中: $\hat{\alpha}$ 为 $E(\alpha)$ 的最小值。式(11)存在数值不稳定的情况,可以通过QR分解^[29]来解决:计算正交矩阵 Q , 使得 Q 与 $[A b]$ 相乘为如下上三角矩阵形式^[18]:

$$\begin{bmatrix} \alpha & \alpha & \alpha & \alpha \\ 0 & \alpha & \alpha & \alpha \\ 0 & 0 & \alpha & \alpha \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{b} \\ 0 & r \\ 0 & 0 \\ \vdots & \vdots \end{bmatrix}, \quad (12)$$

式中: \hat{A} 为 3×3 的上三角矩阵, \hat{b} 为 3×1 的列向量, r 为标量。那么 QEF 可以变换为

$$\begin{aligned} (A\alpha - b)^T (A\alpha - b) &= (A\alpha - b)^T Q^T Q (A\alpha - b) = \\ &= (QA\alpha - Qb)^T (QA\alpha - Qb) = \\ &= (\hat{A}\alpha - \hat{b})^T (\hat{A}\alpha - \hat{b}) + r^2. \end{aligned} \quad (13)$$

根据式(13)计算特征点,但在曲率变化较大的区域,如图4(a)所示,利用式(13)计算出来的特征点可能

位于体元外部,针对这种情况,如图4(b)所示,取交点的平均值为特征点,确保生成的特征点一定在体元内部。

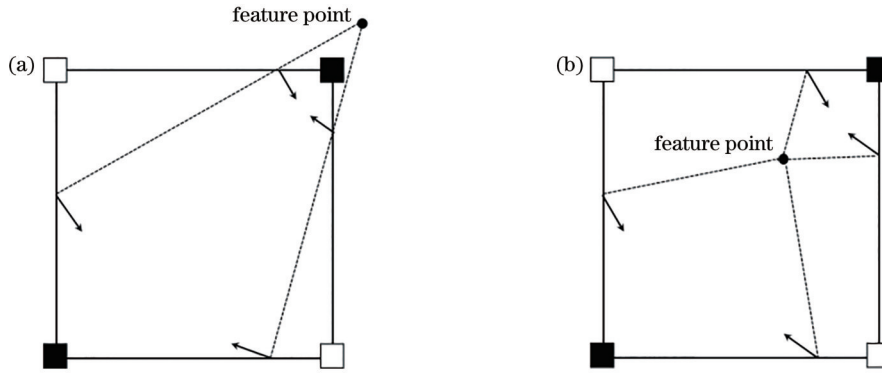


图4 特征点的位置。(a)特征点位于体元之外;(b)特征点位于体元之内

Fig. 4 Location of feature point. (a) Feature point is outside the voxel; (b) feature point is inside the voxel

针对每个边长为1的非空叶子节点,计算出这个节点与等值面交点的位置和法向量,构造出节点的QEF数据,即 $A^T A$ 、 $A^T b$ 、 $b^T b$,利用QEF数据,在节点的内部生成一个特征点。至此,该节点携带了八叉树简化的所有数据,利用这些数据就可对八叉树进行简化,生成自适应八叉树。

4.3 构建自适应八叉树

将八叉树的节点类型分为三类:内部节点、叶子节点和伪叶子节点。叶子节点的度为0,内部节点的度不为0,伪叶子节点由内部节点转换而来,其度也为0。为进一步节省存储空间,针对不同的节点类型,本文设计了不同的数据结构。

八叉树简化是一个自底向上迭代的过程。从倒数第2层开始,尝试将子节点的信息归并到父节点。若节点能简化,删除该节点的子节点,并将该节点的类型由内部节点修改为伪叶子节点。

八叉树任意一个节点能简化必须同时满足以下两个条件:

- ① 其子节点的节点类型为叶子节点或伪叶子节点;
- ② 根据子节点的QEF数据计算出的误差值必须小于给定的简化参数 φ 。

表1展示了齿轮模型在不同简化参数(φ 为负表示不简化)下各类节点的占比。从表1可以看出,简化之后一部分内部节点转为伪叶子节点,同时有一部分叶子节点被删除,随着简化参数的增大,伪叶子节点的占比越来越高。

5 生成三角网格

在每个与等值面相交的体元内部生成一个特征点,每条与等值面相交的棱边上生成一个四边形,该四边形连接共享此棱边的4个体元内的特征点。除去边界部分,所有内部的棱边都一定被4个体元共享,因此

表1 不同简化参数下各类型节点的比例

Table 1 Proportion of each type of node under different simplified parameters

Simplified parameter φ	Proportion of internal node / %	Proportion of leaf node / %	Proportion of pseudo leaf node / %
-0.01	24.81	75.19	0
0.01	24.15	71.49	4.36
0.1	24.14	71.33	4.53
1	23.20	65.80	11.00
10	20.58	0.24	79.18
100	20.41	0	79.59

能保证提取出的面片一定是四边形。首先确定四边形的连接方式,然后按照一定的规则对四边形进行剖分,进而生成三角网格。

5.1 连接四边形

通过判断相交边首端点(坐标值小的端点为首端点)的状态来决定四边形的连接方式。规定网格面的法向量必须朝外,所以法向量须由状态为1的端点指向状态为0的端点。如果首端点状态为1,根据右手法则,如图5所示,此时四边形的连接方式为 $\{0,1,3,2\}$;如果首端点状态为0,则四边形的连接方式为 $\{0,2,3,1\}$ 。

5.2 剖分四边形

因为最终生成的是三角网格,所以需要对四边形进行剖分。以首端点状态为1为例来解释,如图6所示,此时,四边形的剖分方式为 $\{\{0,3,2\},\{0,1,3\}\}$ 或 $\{\{0,1,3\},\{1,2,3\}\}$ 。如果按照上述任意一种剖分方式对四边形进行剖分,那么重建过程中可能会产生少量正则度较差的网格。为了减少狭长三角形的数量,采用面积-边长度量方法,定义三角网格的质量评价指标 q ^[30]

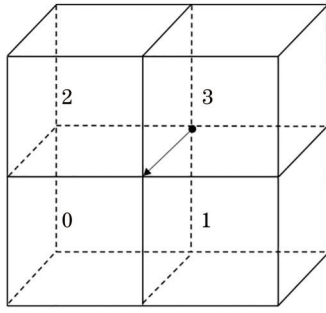


图 5 四边形的连接方式

Fig. 5 Connection mode of quadrilateral

$$q = \frac{4\sqrt{3} \cdot S}{l_1^2 + l_2^2 + l_3^2}, \quad (14)$$

式中: S 为三角形的面积; l_1, l_2, l_3 分别为三角形三条边的边长; q 的取值范围为 $0 \leq q \leq 1$, q 越大 ($q=1$ 时, 三角形为等边三角形), 产生的三角形越规范, 其质量也就越好。如图 6 所示, 本文通过比较两种方式产生 q 值的大小来决定选择哪一种方式对四边形进行剖分。

6 实验结果及分析

本文算法在 Visual Studio 2019 环境下利用 C++

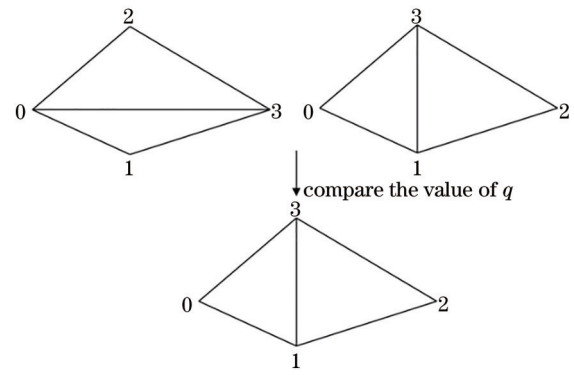
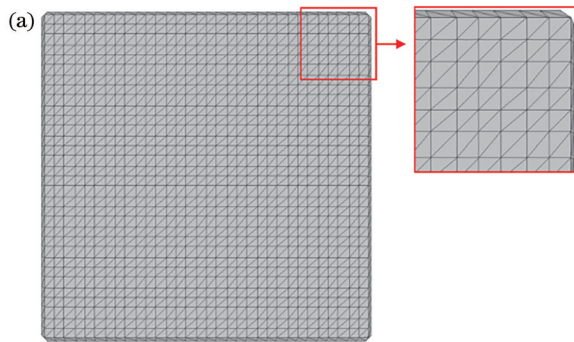


图 6 剖分四边形的方式

Fig. 6 Mode of dividing quadrilateral

编程语言实现, 所有的实验结果均在一台 AMD Ryzen 5 3600 6-Core 处理器、16 GB 内存、NVIDIA GeForce GTX 1650 显卡的 PC 机上测试得到。

6.1 立方体数据实验

为更好展现本文算法保持尖锐特征的能力, 借助 OpenCV 图像处理库生成一组标准的不含噪声的立方体数据集, 数据集的大小为 $128 \times 128 \times 46$, 分别使用 MC 算法与所提算法进行重建, 重建结果如图 7 所示。

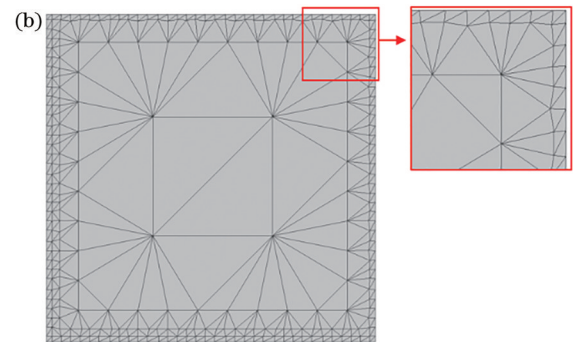


图 7 不同算法的重建结果。(a) MC 算法; (b) 所提算法

Fig. 7 Reconstruction results of different algorithms. (a) MC algorithm; (b) proposed algorithm

如图 7(a) 所示, 因为 MC 算法没有利用交点的法向量, 所以无法保持尖锐特征; 如图 7(b) 所示, 本文算法不仅可以构造出立方体的棱边, 还可以利用更小的三角网格来表示棱角这种尖锐特征。

6.2 齿轮 CT 数据实验

为展示本文算法的简化效果, 使用一组实际扫描的齿轮 CT 序列图像数据进行实验, 图像的分辨率为 $1024 \text{ pixel} \times 1024 \text{ pixel}$, $D_z = 143$, $V_{\text{depth}} = 10$, $V_{\text{iso}} = 87$ 。如图 8(a) 所示, 先利用本文算法生成未简化的网格模型, 然后采用文献 [9] 算法和文献 [12] 算法对模型进行简化。文献 [9] 采用的是一种均匀简化算法, 不区分平坦区域和非平坦区域, 如图 8(b) 所示, 简化之后会造成模型细节特征丢失。图 8(c) 展示了文献 [12] 算法的简化效果, 文献 [12] 虽然能做到保持特征, 但产生了大量的狭长三角形, 生成的网格质量较差。与这

两种算法相比, 如图 8(d) 所示, 本文算法在模型的平坦区域和非平坦区域都能获得较好的简化效果, 在保持模型的细节特征的同时有效避免了狭长三角形的产生, 保证了简化后模型的质量。此外, 文献 [9] 和文献 [12] 算法都需要根据 CT 图像先生成未简化的网格模型, 再对生成的模型进行简化, 而本文算法可以由 CT 图像直接得到简化的网格模型。

6.3 引擎 CT 数据实验

为进一步验证本文算法的有效性, 使用一组具有复杂结构的引擎 CT 序列图像数据进行实验, 图像的分辨率为 $256 \text{ pixel} \times 256 \text{ pixel}$, $D_z = 100$, $V_{\text{depth}} = 8$, $V_{\text{iso}} = 106$ 。简化参数的大小决定了八叉树的简化程度, 选取不同的简化参数, 重建出图 9 所示的网格模型。图 9(a)~(f) 产生三角网格的数量依次为 563700、475826、470240、384968、114528、44912。如图 9 所示,

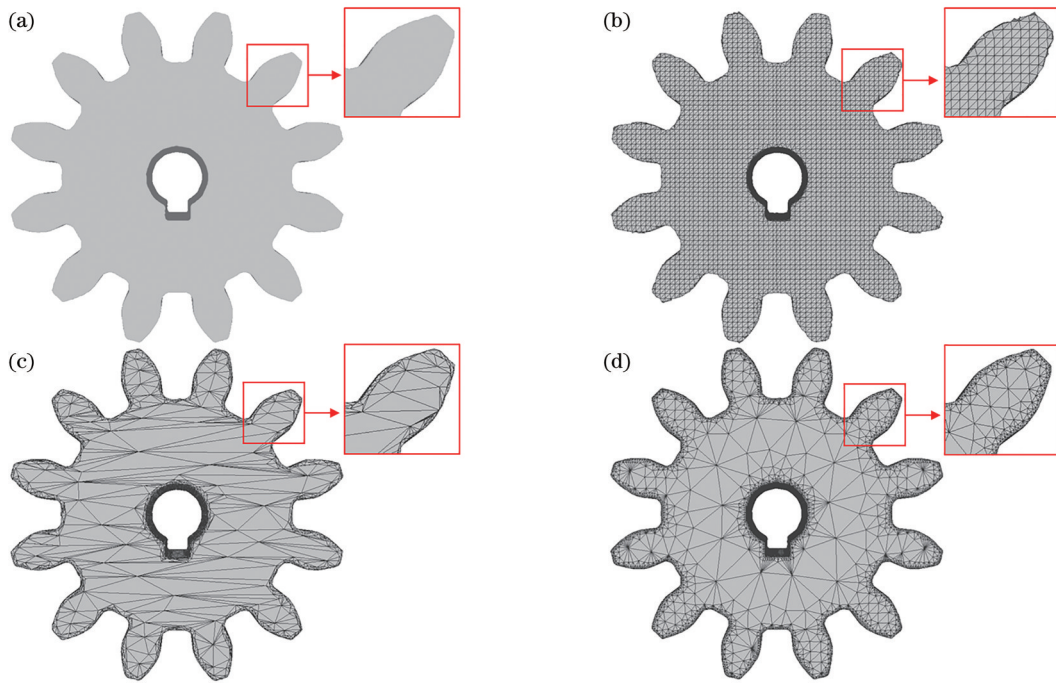


图 8 采用不同的简化算法对齿轮模型进行简化。(a)网格模型;(b)文献[9]算法;(c)文献[12]算法;(d)所提算法
 Fig. 8 Using different simplification algorithms to simplify the gear model. (a) Mesh model; (b) algorithm of Ref. [9]; (c) algorithm of Ref. [12]; (d) proposed algorithm

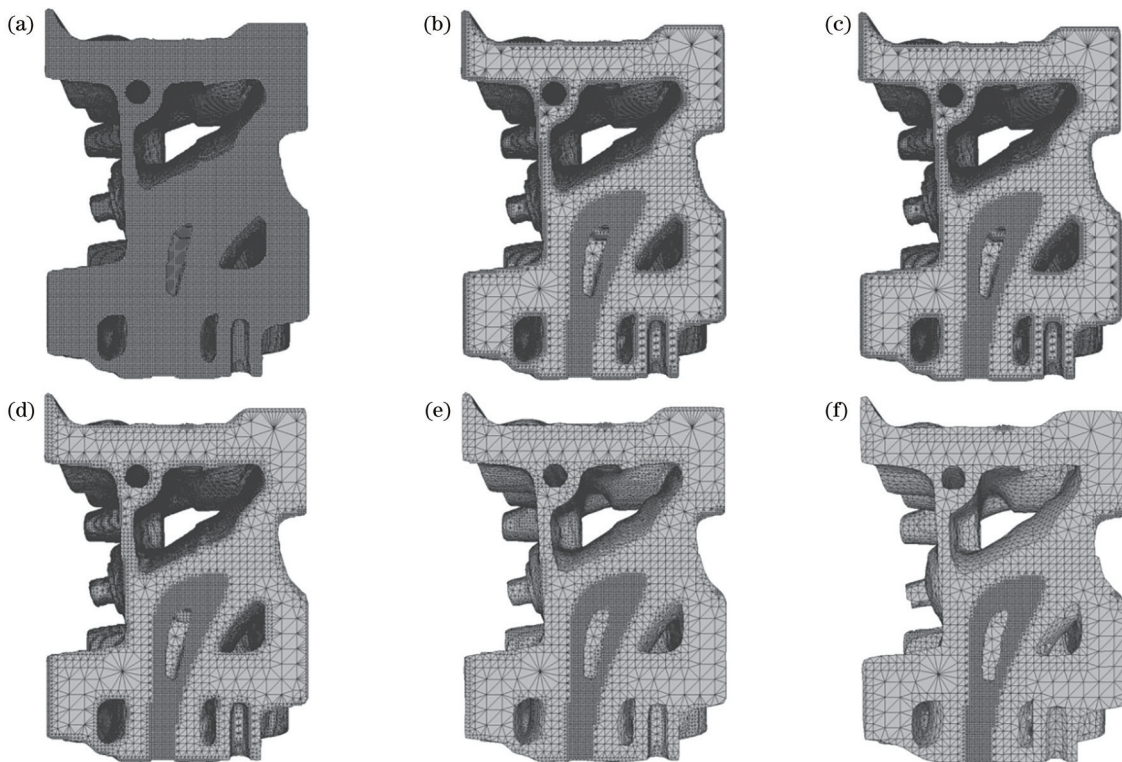


图 9 不同简化参数下的引擎模型。(a)未简化模型($\varphi = -0.01$);(b)简化模型 1($\varphi = 0.01$);(c)简化模型 2($\varphi = 0.1$);(d)简化模型 3($\varphi = 1$);(e)简化模型 4($\varphi = 10$);(f)简化模型 5($\varphi = 100$)

Fig. 9 Engine models under different simplified parameters. (a) Unsimplified model ($\varphi = -0.01$); (b) simplified model 1 ($\varphi = 0.01$); (c) simplified model 2 ($\varphi = 0.1$); (d) simplified model 3 ($\varphi = 1$); (e) simplified model 4 ($\varphi = 10$); (f) simplified model 5 ($\varphi = 100$)

本文算法会优先简化较为平坦的区域,因为该区域特征相对不明显,简化后有利于保持原始模型的几何特性。同时,由于该区域的网格单元面积一般较大,因此

简化后会引入一定的视觉误差,而在模型特征明显的区域,其网格分布较为密集且网格单元的面积相对较小,通过改变简化参数,可以获取到不同的简化效果。

在给定的简化参数下,本文算法可根据物体局部特征,自适应地不同大小的体元内提取等值面,实现自适应三维网格模型重建。

最后,选取 $q=0.3$ 为三角网格质量阈值,表 2 统计了在不同简化参数下齿轮和引擎数据,重建产生网格模型的简化率和网格的质量。由表 2 可知,针对不同分辨率的工业 CT 序列图像,本文算法的三角网格简化率可达 90%,简化后网格质量大于 0.3 的三角网格平均占比为 99%。因此,可根据实际需要设定不同的简化参数。

表 2 所提算法在不同参数下的性能表现

Table 2 Performance of proposed algorithm with different parameters

Model	Simplified parameter φ	Simplification rate / %	Mesh quality / %
Gear	-0.01	0	100.00
	0.01	42.90	99.77
	0.1	43.20	99.76
	1	60.36	99.57
	10	90.09	99.17
	100	96.74	99.05
Engine	-0.01	0	100.00
	0.01	15.59	99.95
	0.1	16.58	99.94
	1	31.71	99.90
	10	79.68	99.60
	100	92.03	99.56

7 结 论

为提高工业 CT 图像重建三角网格模型的质量、减少产生三角面片的数量、同时保持重建模型的尖锐特征,提出一种自适应的三维网格模型重建算法。与 MC 算法对比,验证了本文算法保持尖锐特征的能力;与其他的网格简化算法对比,验证了本文算法在保持特征的同时也能很好地保证简化后网格的质量;修改不同的简化参数,可以获取不同的简化效果。本文算法可以同时解决由工业 CT 图像重建三角网格模型存在的尖锐特征丢失、狭长三角形和三角面片数量多等问题,提高模型的重建质量。进一步提升重建精度,实现亚体素精度的表面提取是下一步研究的重点。

参 考 文 献

[1] 曹增欢, 黄常标, 郑红. 三角网格模型的特征保持混合折叠简化[J]. 光学精密工程, 2019, 27(4): 971-983.
Cao Z H, Huang C B, Zheng H. Feature preserving mesh simplification based on hybrid collapse[J]. Optics and Precision Engineering, 2019, 27(4): 971-983.

[2] 余浩松, 邹永宁, 张智斌, 等. 利用 CAD 模型的不完全扫描 CT 图像重建[J]. 光学学报, 2021, 41(6): 0611002.
Yu H S, Zou Y N, Zhang Z B, et al. Image reconstruction of

incomplete CT scanning using a CAD model[J]. Acta Optica Sinica, 2021, 41(6): 0611002.

[3] 段黎明, 方诚, 罗雪清, 等. 工业 CT 图像直接生成三角网格有限元模型[J]. 光学学报, 2021, 41(14): 1411002.
Duan L M, Fang C, Luo X Q, et al. Direct generation of triangular mesh finite element model from industrial CT images[J]. Acta Optica Sinica, 2021, 41(14): 1411002.

[4] 龙超, 金恒, 黎玲, 等. 基于特征融合的非局部均值 CT 图像降噪[J]. 光学学报, 2022, 42(11): 1134024.
Long C, Jin H, Li L, et al. CT image denoising with non-local means based on feature fusion[J]. Acta Optica Sinica, 2022, 42(11): 1134024.

[5] Lorensen W E, Cline H E. Marching cubes: a high resolution 3D surface construction algorithm[J]. ACM SIGGRAPH Computer Graphics, 1987, 21(4): 163-169.

[6] Kobbelt L P, Botsch M, Schwanecke U, et al. Feature sensitive surface extraction from volume data[C]//SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, August 12-17, 2001, Los Angeles, California. New York: ACM Press, 2001: 57-66.

[7] 袁小翠, 吴禄慎, 陈华伟. 尖锐特征曲面散乱点云法向估计[J]. 光学精密工程, 2016, 24(10): 2581-2588.
Yuan X C, Wu L S, Chen H W. Normal estimation of scattered point cloud with sharp feature[J]. Optics and Precision Engineering, 2016, 24(10): 2581-2588.

[8] 陈甜甜, 赵罡. 半规则三角网格模型细分曲面重构[J]. 北京航空航天大学学报, 2012, 38(9): 1245-1249.
Chen T T, Zhao G. Semi-regular triangular mesh subdivision surface reconstruction[J]. Journal of Beijing University of Aeronautics and Astronautics, 2012, 38(9): 1245-1249.

[9] Low K L, Tan T S. Model simplification using vertex-clustering [C]//I3D '97: Proceedings of the 1997 Symposium on Interactive 3D Graphics, April 27-30, 1997, Providence, RI, USA. New York: ACM Press, 1997: 75-82.

[10] Schroeder W J, Zarge J A, Lorensen W E. Decimation of triangle meshes[C]//SIGGRAPH '92: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, July 27-31, 1992, Chicago, IL, USA. New York: ACM Press, 1992: 65-70.

[11] Hamann B. A data reduction scheme for triangulated surfaces[J]. Computer Aided Geometric Design, 1994, 11(2): 197-214.

[12] Hoppe H. Quadric metric for simplifying meshes with appearance attributes: US6362820[P]. 2002-03-26.

[13] Vignoles G L, Donias M, Mulat C, et al. Simplified marching cubes: an efficient discretization scheme for simulations of deposition/ablation in complex media[J]. Computational Materials Science, 2011, 50(3): 893-902.

[14] Shu R B, Zhou C, Kankanhalli M S. Adaptive marching cubes [J]. The Visual Computer, 1995, 11(4): 202-217.

[15] Shekhar R, Fayyad E, Yagel R, et al. Octree-based decimation of marching cubes surfaces[C]//Proceedings of Seventh Annual IEEE Visualization '96, October 27-November 1, 1996, San Francisco, CA, USA. New York: IEEE Press, 1996: 335-342.

[16] 熊邦书, 魏江. 变分辨率的曲面重建算法[J]. 系统仿真学报, 2007, 19(18): 4126-4129, 4133.
Xiong B S, Wei J. Variable resolution reconstruction of surface [J]. Journal of System Simulation, 2007, 19(18): 4126-4129, 4133.

[17] 徐雷, 王华锋, 潘海侠, 等. 基于八叉树的简化构型三角片等值面削减算法[J]. 北京航空航天大学学报, 2018, 44(4): 851-861.
Xu L, Wang H F, Pan H X, et al. Octree based decimation algorithm for triangle isosurface using simplified patterns[J]. Journal of Beijing University of Aeronautics and Astronautics, 2018, 44(4): 851-861.

[18] Ju T, Losasso F, Schaefer S, et al. Dual contouring of Hermite data[C]//Proceedings of the 29th Annual Conference on

- Computer Graphics and Interactive Techniques-SIGGRAPH, July 23-26, 2002, San Antonio, Texas. New York: ACM Press, 2002: 339-346.
- [19] Zhang Y J, Bajaj C. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data[J]. *Computer Methods in Applied Mechanics and Engineering*, 2006, 195: 942-960.
- [20] 陈俊杰, 金小刚. 基于 GPU 的高质量隐式曲面四边形化[J]. *计算机辅助设计与图形学学报*, 2016, 28(4): 549-555.
Chen J J, Jin X G. GPU-based high quality quadrangulation for implicit surfaces[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2016, 28(4): 549-555.
- [21] Lobello R U, Dupont F, Denis F. Multi-resolution dual contouring from volumetric data[C]//GRAPP/IVAPP: Proceedings of the International Conference on Computer Graphics Theory and Applications, February 24-26, 2012, Rome, Italy. Setúbal: SciTePress, 2016: 163-168.
- [22] 彭守建, 吴斌, 许江, 等. 基于 CGAL 的岩石裂隙面三维重构方法研究[J]. *岩石力学与工程学报*, 2020, 39(S2): 3450-3463.
Peng S J, Wu B, Xu J, et al. Research on 3D reconstruction method of rock fracture surface based on CGAL[J]. *Chinese Journal of Rock Mechanics and Engineering*, 2020, 39(S2): 3450-3463.
- [23] 马瑜, 赵九龙, 余婷, 等. 基于三维分数阶微分增强的边缘曲面重构算法[J]. *光子学报*, 2014, 43(10): 1011004.
Ma Y, Zhao J L, Yu T, et al. The reconstruction algorithm of boundary surfaces based on the 3D fractional differential enhancement[J]. *Acta Photonica Sinica*, 2014, 43(10): 1011004.
- [24] Otsu N. A threshold selection method from gray-level histograms [J]. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979, 9(1): 62-66.
- [25] Hunter G M. Efficient computation and data structure for graphics[D]. Princeton: Department of Electrical and Computer Science, Princeton University, 1978.
- [26] Ohtake Y, Belyaev A, Alexa M, et al. Multi-level partition of unity implicits[C]//SIGGRAPH '05: ACM SIGGRAPH 2005 Courses, July 31-August 4, Los Angeles, California, USA. New York: ACM Press, 2005: 173.
- [27] Kazhdan M, Hoppe H. Screened Poisson surface reconstruction [J]. *ACM Transactions on Graphics*, 2013, 32(3): 29.
- [28] 唐泽圣. 三维数据场可视化[M]. 北京: 清华大学出版社, 1999.
Tang Z S. Visualization of three-dimensional data field[M]. Beijing: Tsinghua University Press, 1999.
- [29] Schaefer S, Ju T, Warren J. Manifold Dual Contouring[J]. *IEEE Transactions on Visualization & Computer Graphics*, 2007, 13(3): 610-619.
- [30] 段黎明, 邵辉, 李中明, 等. 高效率的三角网格模型保特征简化方法[J]. *光学精密工程*, 2017, 25(2): 460-468.
Duan L M, Shao H, Li Z M, et al. Simplification method for feature preserving of efficient triangular mesh model[J]. *Optics and Precision Engineering*, 2017, 25(2): 460-468.

Adaptive 3D Mesh Model Reconstruction Based on Industrial CT Images

Li Ling^{1,2}, Jin Heng^{1,2}, Liu Jie^{1,3}, Long Chao^{1,3}, He Yunyong^{1,3}, Li Zhongming^{1,2},
Duan Liming^{1,2*}

¹*ICT Research Center, Key Laboratory of Optoelectronic Technology & Systems, Ministry of Education, Chongqing University, Chongqing 400044, China;*

²*College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China;*

³*College of Optoelectronic Engineering, Chongqing University, Chongqing 400044, China*

Abstract

Objective As a typical numerical representation of geometric models, the triangular mesh is widely used in additive manufacturing, inverse design, and finite element analysis. The triangular mesh model is directly reconstructed based on industrial CT images, which allows for the reconstruction of 3D representations of parts with complicated internal cavity structures. However, current algorithms for reconstructing triangular mesh models based on industrial CT images, for example, marching cube (MC) algorithm, have problems such as loss of sharp features, many long-narrow triangles, and a large number of triangular surfaces. In this paper, we propose an adaptive 3D mesh model reconstruction method to simultaneously address these issues while improving the quality of the reconstructed triangular mesh model from industrial CT images.

Methods First, a bilateral filter and an OTSU algorithm are utilized to preprocess industrial CT images, so as to denoise and determine the value of the isosurface. Second, an octree structure is used to confirm the voxels; the octree is created top-down recursively, while non-boundary voxels are deleted to save storage space. The quadratic error function (QEF) is then applied to each boundary voxel of the octree to produce a feature point, and the octree is simplified by merging the feature points from the bottom up. Third, a quadrilateral formed by four adjacent feature points is divided into two triangular meshes. In order to validate the performance of the proposed algorithm, experiments are performed using a cubic dataset and two groups of real industrial CT images.

Results and Discussions To begin with, a cubic dataset with no noise is utilized and reconstructed using the MC algorithm and the approach proposed in this study, as shown in Fig. 7. Sharp features including angles are lost by the MC algorithm [Fig. 7(a)]. The approach in this article not only generates the cube's edges but also a smaller triangular mesh to

represent sharper features such as angles [Fig. 7(b)]. Next, in order to demonstrate the algorithm's simplification effect in this paper, an experiment is performed utilizing a set of real scanned gear CT sequence image data. This study's algorithm generates an unsimplified mesh model [Fig. 8(a)], which is then processed using vertex clustering and edge-shrinking algorithms. The vertex clustering approach results in the loss of the model's detailed features after simplification [Fig. 8(b)], whereas the edge-shrinking algorithm saves the details but generates a large number of long-narrow triangles with poor mesh quality [Fig. 8(c)]. The method used in this study efficiently prevents the formation of long-narrow triangles while preserving the model's detailed features and guaranteeing the quality of the model [Fig. 8(d)]. Besides, in order to further validate the effectiveness of the algorithm presented in this study, a set of engine CT sequence image data with a complicated internal cavity structure is applied. The mesh models shown in Fig. 9 are reconstructed by varying the simplified parameters. Different simplification effects can be obtained by changing the simplified parameters (Fig. 9). The method used in this study can adaptively extract the isosurface in voxels of different sizes according to the local characteristics of the object and achieve the reconstruction of an adaptive 3D mesh model under the given simplified parameters. The experimental results demonstrate that the method presented in this study can preserve object sharpness while simplifying the mesh, reducing the number of long-narrow triangles, and improving the quality of the triangular mesh model reconstructed from industrial CT images. In this study, the simplification rate of the triangular mesh model generated by the algorithm can achieve 90%, with an average proportion of mesh quality greater than 0.3 after simplification of 99% (Table 2).

Conclusions This paper proposes an adaptive 3D mesh model reconstruction algorithm to deal with the problems of sharp feature loss, many long-narrow triangles, and a large number of triangular surfaces in the reconstructed triangular mesh model of industrial CT images obtained from X-rays. Firstly, the image is denoised by the bilateral filter, and the value of the isosurface is determined using the OTSU algorithm. Then, voxels are organized by an octree structure, and the octree is generated top-down; feature points are generated by minimizing a quadratic error function (QEF), and an adaptive octree is constructed by merging feature points bottom-up. Finally, triangular meshes are generated by dividing the quadrilateral formed by four adjacent feature points. The algorithm in this paper checks its ability to keep sharp features compared with the MC algorithm. Compared with vertex clustering and edge-shrinking mesh simplification algorithms, the algorithm in the present paper can keep features and guarantee the quality of simplified meshes. Under the given simplified parameters, the method used in this research can adaptively extract the isosurface in voxels of different sizes based on the local characteristics of the object and achieve the reconstruction of an adaptive 3D mesh model. From the experimental results, it is found that the simplification rate of the triangular mesh model generated by the algorithm in this paper can be as high as 90%, and the average proportion of mesh quality higher than 0.3 after simplification reaches 99%. Compared with the conventional mesh method, the proposed method can maintain the sharp features of the model while simplifying the mesh, reducing the number of long-narrow triangles, and improving the quality of the reconstructed triangular mesh model from industrial CT images.

Key words X-ray optics; industrial CT images; 3D reconstruction; triangular mesh; adaptive octree