

基于 Nelder-Mead 算法的 3D 打印模型最优化放置

秦爽¹, 黄先锋¹, 张帆¹, 陈伟², 向东清²

(1. 武汉大学 测绘遥感信息工程国家重点实验室, 湖北 武汉 430079;
2. 珠海赛纳打印科技股份有限公司, 广东 珠海 519060)

摘要:针对 3D 打印材料费用昂贵、打印成本高的问题,在保证打印模型的稳定性和表面质量的前提下提出了一种优化算法。该算法首先通过计算打印模型在任意摆放角度下的模型体积和模型顶面投影体积来计算出模型支撑耗材体积,然后利用提出的模型打印时间算法计算出任意摆放角度下的打印时间,最后构造出打印模型在任意摆放角度下的支撑结构耗材体积和打印时间的目标函数。利用经典的 Nelder-Mead 算法,计算出使目标函数最低的摆放角度,即打印模型绕 X 轴、 Y 轴和 Z 轴所旋转的最优角度,从而实现了打印模型的最优化放置。实验结果表明,该算法在保证模型表面质量和稳定性的前提下,减少了打印支撑耗材体积,缩短了打印时间,具有明显的成本优势。

关键词:3D 打印; Nelder-Mead 算法; 支撑体积; 打印时间; 智能优化

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2018)11-0012-04

doi: 10.3969/j.issn.1673-629X.2018.11.003

Orientation Optimization of 3D Printed Model Based on Nelder-Mead Algorithm

QIN Shuang¹, HUANG Xian-feng¹, ZHANG Fan¹, CHEN Wei², XIANG Dong-qing²

(1. State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China;
2. Zhuhai Seine Technology Co., Ltd., Zhuhai 519060, China)

Abstract: Aiming at the problem of expensive 3D printing materials and high printing cost, we propose an optimization algorithm on the premise of ensuring the stability and surface quality of the printing model. Firstly, we calculate the supporting materials volume by calculating the volume of the model and the projection volume of the top surface of the model at any angle, and then calculate the printing time at any angle by the model printing time algorithm proposed. Finally, we construct the objective function of the supporting materials volume and the printing time at any orientation. Using the classic Nelder-Mead algorithm, the optimal placement angle of the print model is calculated to minimize the objective function, that is, the optimal angle of rotation of the print model around the axis X , Y and Z , thus achieving the optimal placement of the print model. Experiment shows that the proposed algorithm can reduce the supporting material volume and shorten the printing time while ensuring the model's surface quality and stability with obvious advantages in the cost.

Key words: 3D printing; Nelder-Mead algorithm; supporting volume; printing time; intelligent optimization

0 引言

3D 打印是以三维模型文件为基础,运用可粘合和可固化的离散材料(粉末、液体、丝等)为打印介质,通过分层打印、逐层堆叠累积的方式来制造物体的技术^[1]。3D 打印能快速而准确地将任意复杂的模型数据制造出实物,同时也带来了如打印成本优化等新的研究问题^[2]。一般用于 3D 打印的材料价格是远远高

于传统制造方式的,因而节省打印耗材成为降低打印成本的重要方法之一^[3]。针对这一问题,许多学者通过优化 3D 打印模型的内部结构来减少打印模型耗材。例如, Wang 等提出了一种基于“蒙皮-刚架”(skin-frame)的轻质结构来减少打印耗材^[4]; Kindinger 等提出基于泡沫、轻木等轻质结构来减少 3D 模型重量^[5]; Lu 等利用蜂窝结构思想提出一种打印模型

收稿日期: 2017-11-27

修回日期: 2018-03-07

网络出版时间: 2018-05-28

基金项目: 国家自然科学基金(41571437)

作者简介: 秦爽(1994-),男,硕士研究生,研究方向为数字图像处理;黄先锋,教授,从事机载、地面激光点云数据处理、文化遗产理论与方法等的研究;张帆,副教授,研究方向为数字图像处理、精细三维重建。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.TP.20180525.1603.060.html>

内部掏空优化算法^[6];Chen 利用蜂窝六边形的网孔结构来填充内部挖空^[7]。

目前,3D 打印软件在节材优化方面上仍十分有限,如果不能合理地设计出打印模型内部结构,在对物体的镂空厚度和密度这些参数进行设定时仍然需要根据用户的经验,不合理的设计会导致打印出的模型不能满足于实际生产应用,还浪费了昂贵的打印材料^[8]。3D 打印材料消耗除了模型耗材外,还包括打印支撑结构耗材。为了让打印材料在有悬空的地方沉积,通常需要在模型有悬空部位的下方添加支撑材料^[9-11]。因此,文中是在不改变模型原有内部结构的前提下,通过对打印模型最优化放置的研究来减少打印支撑耗材。

此外,与传统的制造技术相比,3D 打印技术已大大缩短了模型输出时间,但是打印一个物体仍需花费不少的时间^[2,12-13]。在打印过程中,还需考虑打印时间问题,通过缩短 3D 打印时间从而实现对产品对象的快速打印^[2,14-15]。

文中针对打印耗材和打印时间两个方面进行优化,利用打印模型的支撑结构耗材体积和打印时间这两个参数以及各自所占的权重系数构造出目标函数,采用经典的 Nelder-Mead 算法,在不改变模型内部结构的前提下,计算出打印模型的最优摆放角度,以实现对打印模型的最优化放置,达到降低成本的目的。

1 模型支撑耗材体积计算

如图 1 所示,支撑耗材体积(V_s)计算思路为,将模型的上顶面投影到打印平面的体积(V_p)减去模型实体体积(V_m),即 $V_s = V_p - V_m$,因此,可拆分为模型顶面投影体积计算和模型实体体积计算两个部分。

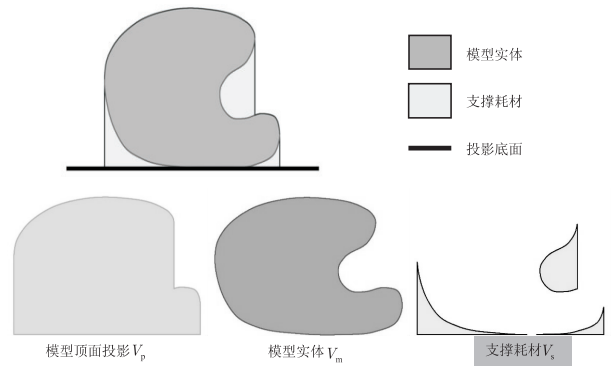


图 1 模型耗材显示图

1.1 模型实体体积计算

文中分析的三维模型数据为三角网模型,由顶点和三角形构成,三角形的法向量满足右手准则,即所有三角形面片的法向一致,均指向体外。

设三维模型中的任意一个三角形顶点为 $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$, $C(x_3, y_3, z_3)$, 如图 2 所示,以坐标

原点 $O(0,0,0)$ 为顶点,以 ΔABC 为底的三棱锥的体积为:

$$V_{oabc} = \frac{1}{6} \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

如果 ΔABC 为顺时针排序,根据右手准则,右手握拳方向为 ΔABC 的排序方向,大拇指指向为三角形法向的方向,那么原点 O 在 ΔABC 法向的正方向上,则 V_{OABC} 为正数,反之 V_{OABC} 为负数^[10]。

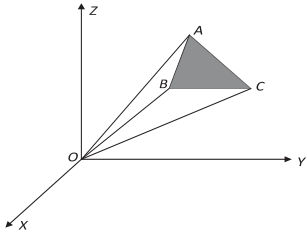


图 2 模型体积计算

对任一给定三维模型数据,它的模型表面是由 n 个三角形面片组成,如果以坐标原点 O 为顶点,依次以这 n 个三角形为底可以构成 n 个三棱锥,那么该三维模型的体积计算公式为:

$$v_T = \sum_{i=1}^n v_i$$

其中, V_i 为第 i 个三角形与坐标原点构成的带符号的体积,其正负符号根据坐标原点 O 和该三角形法向量的正负方向确定。

1.2 模型顶面投影体积计算

体积计算需要逐个单元进行分析,三维显示引擎 OpenGL 提供了快速的几何模型投影计算能力。利用 OpenGL 的成像机理和深度缓存机制来计算打印物体在正射投影下的体积。

1.2.1 模型变换矩阵的计算

假设模型变换矩阵为 M_e ,模型绕 X 轴、 Y 轴和 Z 轴旋转的角度分别为 DX 、 DY 和 DZ 。为了使模型围绕自身坐标轴进行旋转,首先对模型进行平移变换,将模型移动到坐标原点,然后进行旋转变换,最后又利用平移变换将模型移动到模型的中心点。由于缩放矩阵会改变模型的大小,因此在计算过程中不考虑缩放矩阵的影响,具体步骤为:

(1) 将旋转角度转换为弧度:

$$\begin{cases} RX = DX/180 \times \pi \\ RY = DY/180 \times \pi \\ RZ = DZ/180 \times \pi \end{cases}$$

将模型移到坐标原点的平移矩阵为 T_1 ,遍历场景中模型所有的顶点坐标,得到在 X 轴、 Y 轴和 Z 轴方向上的最大值和最小值,假设分别为 Max_x 、 Min_x 、 Max_y 、 Min_y 、 Max_z 和 Min_z ,并求出模型的中心点坐标

(X_c, Y_c, Z_c) 。满足以下关系式:

$$X_c = (\text{Max}_x + \text{Min}_x) / 2$$

$$Y_c = (\text{Max}_y + \text{Min}_y) / 2$$

$$Z_c = (\text{Max}_z + \text{Min}_z) / 2$$

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & -X_c \\ 0 & 1 & 0 & -Y_c \\ 0 & 0 & 1 & -Z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2) 模型绕 X 轴、 Y 轴和 Z 轴的旋转矩阵分别为

R_x 、 R_y 和 R_z :

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos RX & -\sin RX & 0 \\ 0 & \sin RX & \cos RX & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos RY & 0 & -\sin RY & 0 \\ 0 & 1 & 0 & 0 \\ \sin RY & 0 & \cos RY & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos RZ & -\sin RZ & 0 & 0 \\ \sin RZ & \cos RZ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3) 将模型移到物体的中心的平移矩阵为 T_2 :

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & X_c \\ 0 & 1 & 0 & Y_c \\ 0 & 0 & 1 & Z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(4) 模型变换矩阵 M_v 为:

$$M_v = T_2 \times R_z \times R_y \times R_x \times T_1$$

通过模型变换矩阵和实物模型做相乘运算可以确定三维模型最终的大小、形状和位置,从而可以计算在当前旋转角度下的模型顶面投影体积。

1.2.2 模型顶面投影体积计算

根据计算出来的模型变换矩阵,可以得到模型在经过变换之后的坐标,从而可以计算出模型在 X 轴、 Y 轴和 Z 轴方向上的最大值和最小值,假设分别为 X_{\max} 、 X_{\min} 、 Y_{\max} 、 Y_{\min} 、 Z_{\max} 和 Z_{\min} 。

在 OpenGL 中提供了正射投影和透视投影这两种不同的投影类型。正射投影的投影线互相平行,投影的结果与原来的物体大小相等,符合文中的需求,因此将投影方式设置为正射投影。根据正射投影可以得到打印模型所有的顶面区域,读取系统计算的深度缓存信息得到视口范围内的深度值。

将每一个像素的深度值与初始深度值进行比较,依据深度缓存原理,如果该像素深度值小于初始深度

值,则说明该像素点是物体所在的像素点。然后利用 gluUnProject 函数将所有物体所在的像素点实现屏幕坐标向世界坐标的转换,这样每一个物体所在的像素点都对一个世界坐标。

通过相邻像素点的坐标值可以求出单个像素的边长,计算其面积 S ,将打印模型顶面区域所在的像素点的 Z 坐标值减去 Z_{\min} ,累加值记为 Z_{sum} ,则物体在 Z 轴正射投影下的体积 V_p 可表示为:

$$V_p = S \times Z_{\text{sum}}$$

2 模型打印时间计算

文中提出了一种打印时间的计算方法。假设打印模型所需的时间为 h_t ,模型的总层数为 R ,每一层的厚度为 d mm,每层打印所花的时间为 h ,存在这样的关系:

$$h_t = R \times h$$

$$R = (Z_{\max} - Z_{\min}) / d$$

设每一层的切片长(设定为沿 X 轴方向)为 L mm,宽(设定为沿 Y 轴方向)为 W mm,根据 1.2.1 节有:

$$L = X_{\max} - X_{\min}$$

$$W = Y_{\max} - Y_{\min}$$

喷头长度为 k mm,打印字车在 X 轴的运行速度为 V_x m/s,打印字车在 Y 轴运行速度为 V_y m/s,打印机头走过同一个图像区域的次数为 m ,打印字车从打印结束点回到打印起始点所用耗时为 S (S 一般为常量),则完成每层打印所需的步径数 E (向上取整)为:

$$E = W / (k/m) + 3$$

每层打印耗时为:

$$h = E \times \left(\frac{L}{V_x \times 1000} \right) + \frac{W + k/m \times 3}{V_y \times 1000} + s$$

可以看出上式中的 L 、 W 和 R 三个变量与模型的摆放有关。

3 基于 Nelder-Mead 的最优函数求解

通过计算物体在任意放置角度下的打印支撑耗材体积 f_v 和打印时间 h_t ,构造一个目标函数 F , $F = a \times f_v + b \times h_t$,其中 a 和 b 为各自的权重系数, f_v 和 h_t 的大小与模型的摆放有关,利用单纯形算法求出 F 的最小值,得到模型绕 X 轴、 Y 轴和 Z 轴旋转的最优放置角度。

3.1 Nelder-Mead 算法

Nelder-Mead 算法是一种求解多维函数极值的算法,由于该算法运用了单纯形的概念,因此又称单纯形算法。一个 n 维单纯形是指包含 $n+1$ 个节点的凸多面体,例如,一维单纯形是直线上的线段,二维单纯形

是平面中的三角形,三维单纯形是三维空间中的四面体。

3.2 算法思路和流程

文中研究的是 3D 打印模型的最优化放置功能,最终需要解算出待打印绕 X 轴、Y 轴和 Z 轴所旋转的最佳角度,构造的目标函数有 3 个角度变量,为三维单纯形,在变换过程中只改变顶点的位置和尺寸,几何形状和维度保持不变,始终保持一个 4 个点的点集,即为三维空间中的四面体。

具体的算法如下:
输入:模型绕 X 轴、Y 轴和 Z 轴旋转的角度 DX、DY 和 DZ 以及打印支撑耗材体积和打印时间所占目标函数的权重系数 α 和 β
输出:在该权重系数下使得目标函数值最小的 DX、DY 和 DZ

Step1:在进行每次迭代之前,分别计算 4 个点对应的函数值,并对各自的函数值进行排序,使其满足:
 $f(x_1) \leq f(x_2) \leq f(x_3)$
Step2:找出目标函数最大值 f_h ,次大值 f_s 以及最小值 f_l ,各自对应的点分别为 x_h 、 x_s 和 x_l ,其中 $h=4$, $s=3$, $l=1$ 。

Step3:计算重心点 c , c 为包含最好的 3 个点的多面体的中心,即除点 x_h 外其他点的平均值:

$$c = \frac{1}{3} \sum_{k=1}^3 x_k, k \in (1,2,3)$$

Step4:对单纯形进行变换操作,通过使用反射、扩张或收缩产生的新的点来尝试替代最差的点 x_h ,直到满足收敛条件,得到新的单纯形,从而得到最好的点。

图 3 为具体的计算过程。

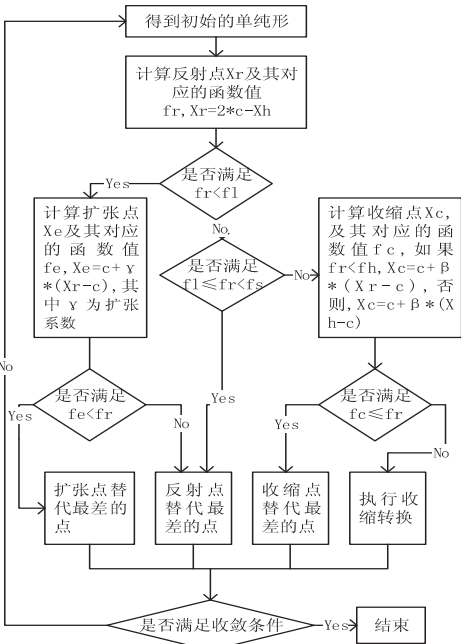


图3 单纯形计算流程

4 实验结果

利用 VS2013 开发工具和 OpenGL 函数库,通过 VC++实现了所提出的优化算法,其中用户界面基于 Qt 进行开发。实验中每层切片的厚度 d 为 0.028 mm,喷头长度 k 为 54 mm,采用 4pass 打印,即 m 为 4,打印字车在 X 轴的运行速度 V_x 为 0.6, V_y 为 0.1, s 为 1.5 s。图 4(a)为模型在初始状态下的放置图,在权重系数框中输入 0 到 1 之间的实数,构造出关于打印模型支撑结构耗材体积和打印时间的带有权重的目标函数。图 4(b)为打印时间权重为 0、耗材权重为 1 得到的打印支撑耗材最优放置图;图 4(c)为打印时间权重为 1、耗材权重为 0 得到的打印时间最优的放置图。

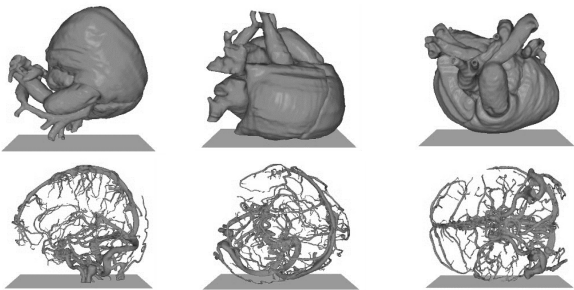


图 4 各种状态下的模型放置图

如表 1 和表 2 所示,在不改变模型内部结构的前提下,使用文中算法可以明显减少打印支撑耗材体积,提高打印速度,从而达到降低成本的目的。

表 1 心脏在各种摆放状态下的打印成本

指标	三角形 总数	打印 时间/s	支撑耗材 体积/mm ³
初始位置	1 228 404	10 347	115 864
打印时间最优	1 228 404	7 736	94 091
支撑耗材最优	1 228 404	8 179	76 186

表 2 血管在各种摆放状态下的打印成本

指标	三角形 总数	打印 时间/s	支撑耗材 体积/mm ³
初始位置	320 496	36 411	612 135
打印时间最优	320 496	32 102	584 117
支撑耗材最优	320 496	43 207	527 359

5 结束语

针对目前 3D 打印材料费用昂贵、打印成本高的问题,提出了一种基于 Nelder-Mead 的优化算法。该算法计算出了模型在任意摆放角度下的支撑结构耗材体积和打印时间,通过不断地迭代实现了对模型的最优放置,从而减少了打印支撑结构耗材、缩短了打印时间,达到了降低 3D 打印成本的目的。

参考文献:

[1] CATTELL R. Scalable SQL and NoSQL data stores[J]. ACM SIGMOD Record,2011,39(4):12-27.

[2] BARABASI A L. Linked: how everything is connected to everything else and what it means for business, science, and everyday life[M]. New York: Plume Book, 2009.

[3] ROBINSON I, WEBBER J, EIFREM E. Graph databases[J]. Oreilly Media, 2013, 7(4): 15-31.

[4] WOOD P T. Query languages for graph databases[J]. ACM SIGMOD Record, 2012, 41(1): 50-60.

[5] 王余蓝. 图形数据库 NEO4J 与关系型数据库的比较研究[J]. 现代电子技术, 2012, 35(20): 77-79.

[6] 王青松, 李 爽, 马瑞萍, 等. 基于模糊聚类分析的数据库模糊查询的研究[J]. 小型微型计算机系统, 2015, 36(6): 1199-1203.

[7] 王 慧. 关系型数据库的模糊查询方法研究与实现[D]. 南京: 南京信息工程大学, 2009.

[8] 陈逸菲, 叶小岭, 张颖超. 基于语言变量的关系数据库模糊查询[J]. 计算机工程, 2009, 35(9): 28-30.

[9] 樊新华. 基于关系数据库的模糊查询技术[J]. 计算机与数字工程, 2009, 37(10): 149-152.

[10] 郑知卉. 关系数据库模糊聚合查询方法研究[D]. 沈阳: 东北大学, 2012.

[11] CASTELLTORT A, LAURENT A. Fuzzy queries over NoSQL graph databases: perspectives for extending the cypher language[C]//15th international conference on information

processing and management of uncertainty in knowledge-based systems. [s. l.]: [s. n.], 2014: 384-395.

[12] CASTELLTORT A, MARTIN T. Handling scalable approximate queries over NoSQL graph databases: cypherf and the Fuzzy4S framework [J]. Fuzzy Sets & Systems, 2017, 5(40): 1-29.

[13] CASTELLTORT A, LAURENT A. Rogue behavior detection in NoSQL graph databases[J]. Journal of Innovation in Digital Ecosystems, 2016, 3(2): 70-82.

[14] CASTELLTORT A, LAURENT A. Fuzzy historical graph pattern matching a NoSQL graph database approach for fraud ring resolution[C]//IFIP international conference on artificial intelligence applications and innovations. [s. l.]: [s. n.], 2015: 151-167.

[15] SOUGUI I B, HIDRI M S, GRISSA-TOUZI A. No-FSQL: a graph-based fuzzy NoSQL querying model[J]. International Journal of Fuzzy Systems Applications, 2016, 5(2): 54-63.

[16] PIVERT O, SLAMA O, SMITS G, et al. SUGAR: a graph database fuzzy querying system[C]//IEEE tenth international conference on research challenges in information science. Grenoble, France: IEEE, 2016: 1-2.

[17] PIVERT O, SMITS G, THION V. Expression and efficient processing of fuzzy queries in a graph database context [C]//IEEE international conference on fuzzy systems. Istanbul, Turkey: IEEE, 2015: 1-8.

(上接第 15 页)

该优化算法主要应用于软材质实心打印材料,用于医学手术模拟训练和手动切割,因此,没有考虑模型的内部结构优化。未来需要考虑内部支撑结构的优化,从而适应更多的打印需求。

参考文献:

[1] 卢秉恒,李涤尘. 增材制造(3D 打印)技术发展[J]. 机械制造与自动化, 2013, 42(4): 1-4.

[2] 刘利刚,徐文鹏,王伟明,等. 3D 打印中的几何计算研究进展[J]. 计算机学报, 2015, 38(6): 1243-1267.

[3] 胡瑞珍,黄 惠. 3D 打印启发下的模型实例化优化研究综述[J]. 计算机辅助设计与图形学学报, 2015, 27(6): 961-967.

[4] WANG Weiming, WANG Tuanfeng, YANG Zhouwang, et al. Cost-effective printing of 3D objects with skin-frame structures[J]. ACM Transactions on Graphics, 2013, 32(6): 177.

[5] KINDINGER J. Lightweight structural cores, specifying structural core[J]. ASM Handbook, 2001, 21: 180-183.

[6] LU Lin, SHARF A, ZHAO Haisen, et al. Build-to-last;

strength to weight 3D printed objects[J]. ACM Transactions on Graphics, 2014, 33(4): 97.

[7] CHEN Y. 3D texture mapping for rapid manufacturing[J]. Computer-Aided Design and Applications, 2007, 4(1): 761-771.

[8] 傅驰林,刘 斌. 3D 打印模型节材优化[J]. 计算机辅助设计与图形学学报, 2017, 29(4): 742-750.

[9] 沈振宏. 三维打印支撑结构的研究与实现[D]. 南京: 南京航空航天大学, 2016.

[10] 陈 岩,王士玮,杨周旺,等. FDM 三维打印的支撑结构的设计算法[J]. 中国科学: 信息科学, 2015, 45(2): 259-269.

[11] 欧立松. 面向三维打印的几何模型后处理技术研究[D]. 南京: 南京航空航天大学, 2015.

[12] 吴芬芬. 3D 打印物体的稳定平衡优化[D]. 合肥: 中国科学技术大学, 2016.

[13] 徐文鹏. 3D 打印中的结构优化问题研究[D]. 合肥: 中国科学技术大学, 2016.

[14] 李小丽,马剑雄,李 萍,等. 3D 打印技术及应用趋势[J]. 自动化仪表, 2014, 35(1): 1-5.

[15] 王运赣,王 宣. 三维打印技术[M]. 武汉: 华中科技大学出版社, 2013.