

遗传退火算法的模型相似性计算方法

高雪瑶¹, 谭涛¹, 张春祥²

(1. 哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080; 2. 哈尔滨理工大学 软件与微电子学院, 黑龙江 哈尔滨 150080)

摘 要:为了检索最相似的 CAD 模型, 本文结合遗传算法的全局搜索能力和模拟退火算法的局部搜索能力, 提出了基于遗传退火算法的模型相似性度量方法。利用面的边数差异来计算源模型面与目标模型面之间的形状相似性。结合面的形状相似性和面的邻接关系来计算面的结构相似性。以面的形状相似性和结构相似性为基础, 构造 2 个模型的整体相似度矩阵。利用遗传退火算法对该矩阵进行搜索, 得到 2 个模型之间的最优面匹配序列。以最优面匹配序列为基础, 计算 2 个模型的相似性。实验结果表明: 相对于模拟退火算法, 本文所提出方法使 13.33% 的模型的排序效果有所改善。该方法能够更准确地度量 2 个模型之间的差异。

关键词:遗传算法; 模拟退火算法; 遗传退火算法; 形状相似性; 邻接关系; 结构相似性; 整体相似度矩阵; 面匹配序列

DOI: 10.11990/jheu.201901093

网络出版地址: <http://www.cnki.net/kcms/detail/23.1390.u.20200518.1802.005.html>

中图分类号: TP391.7 文献标志码: A 文章编号: 1006-7043(2020)07-1073-07

Method of calculating model similarity based on genetic annealing algorithm

GAO Xueyao¹, TAN Tao¹, ZHANG Chunxiang²

(1. School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China; 2. School of Software and Microelectronics, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: To retrieve the most identical CAD model, this study proposes a model similarity calculation method based on a genetic annealing algorithm, which combines the global searching ability of a genetic algorithm and the local searching ability of a simulated annealing algorithm. The difference of the faces' edge numbers is applied to compute the shape similarity between source and target faces. The shape similarity and adjacency relation of the faces are combined to calculate the structural similarity. Based on the shape and structural similarities of faces, a global similarity matrix of two models is constructed. The genetic annealing algorithm is used to find an optimal matching sequence of faces between the two models. Based on this optimal matching sequence of faces, the similarity of two models is calculated. Experimental results show that compared with simulated annealing algorithm, the proposed method improves the ranking effect of 13.33% of models, which proves that it can measure the difference of two models accurately.

Keywords: genetic algorithm; simulated annealing algorithm; genetic annealing algorithm; shape similarity; adjacency relation; structure similarity; global similarity matrix; matching sequence of faces

在 CAD 领域中, 80% 的设计是对原有方案的重用和修改。为了从规模庞大的模型库中获取所需要

的模型, 许多研究者使用了基于三维视图的模型检索方法^[1], 如樊亚春等^[2]、范菁等^[3]采用局部视图信息对模型进行组合描述。刘志等^[4]获取了三维模型较优视点集, 并从各个视点混合轮廓线视图中抽取边缘响应特征。利用自然图像与特征线的边缘相似性来检索三维模型。但是视图的投影方向和数目会对模型特征的精度产生影响, 而且此方法对于拓扑结构复杂的三维模型, 效果不稳定。利用模型

收稿日期: 2019-01-31.

网络出版日期: 2020-05-19.

基金项目: 国家自然科学基金项目(61502124, 60903082); 中国博士后科学基金项目(2014M560249); 黑龙江省普通高校基本科研业务费专项资金项目(LGYC2018JC014); 黑龙江省自然科学基金项目(F2015041, F201420).

作者简介: 高雪瑶, 女, 教授;
张春祥, 男, 教授.

通信作者: 张春祥, E-mail: zxc6x666@163.com.

本体的结构信息^[5]结合三维模型面的几何属性^[6],是行之有效的办法。徐静等^[7]根据外部边闭合规范、待定边转换规范和表面区域合并规范,将三维零件模型分割为若干个区域。对分割区域进行编码,利用区域结构编码来检索零件。此外,多属性融合的特征描述子^[8-9]能更好地描述模型之间的差异。王新颖^[10]从多个不同的方向上利用张量来抽取特征,给出了一种加权多线性主成分分析方法,并将其应用于三维模型的特征提取过程之中。Furuya等^[11]将部分整体关系嵌入网络算法应用于三维模型检索过程中,能够有效地检测部分整体包含关系,并将其融入特征空间之中。Wang等^[12]比较三维模型面,对比中立点的形状特征来计算2个三维模型之间的相似性。Tao^[13]利用顶点之间的兼容性矩阵和边缘兼容性矩阵表示模型,采用归一化方法来构造最优顶点映射矩阵,使用分级分配算法来检索CAD模型。Li等^[14]使用多个分类器来预测给定的三维模型,使用不精确推理理论来融合所有的预测结果,提高了高维空间的搜索效率。Chen等^[15]以样本模型作为对齐目标,将所有三维模型依次排列,提出了一种基于样本对齐的模型检索方法。

本文提取模型的几何拓扑属性^[16],综合考虑几何信息在模型相似性计算中的作用^[17]。利用边数差异来计算面的形状相似性,进而构建2个模型之间的面相似度矩阵。以模型面的邻接关系和面的形状相似性为基础来度量2个模型面的结构相似性。综合面的形状相似性和结构相似性来建立2个模型的整体相似度矩阵。融合遗传算法和模拟退火算法来搜索最优面匹配序列,并计算这2个模型之间的相似性。

1 模型之间的相似性计算

模型是由多个面组成的。模型面的形状差异,造成了模型之间的千差万别。为了计算2个模型之间的相似性,需要度量这2个模型面之间的形状差异。面是由边构成的,2个边数不同的面,其形状一定有差别。本文利用边数差异来度量2个面之间的形状相似性。使用面相似性来度量2个模型面的形状差异。如果2个模型面的组成边数差异越小,那么这2个面的形状就越相似。如果2个模型面之间的组成边数差异越大,那么这2个面之间的形状差异就越大。本文以源模型 U 和目标模型 V 为例来说明模型之间的形状相似性计算过程。

带有槽的三棱柱如图1所示。图2也是一个带有槽的三棱柱,只是面的编号有所不同。

此处,使用 u_s 表示源模型 U 的面; v_t 表示目标模型 V 的面。源模型面 u_s 和目标模型面 v_t 之间的

面相似性计算过程为:

$$S_f(u_s, v_t) = 1 - \frac{|N(u_s) - N(v_t)|}{\max(N(u_s), N(v_t))} \quad (1)$$

式中: $N(f)$ 表示面 f 的组成边数; $\max(x, y)$ 表示 x 与 y 之间的最大值。

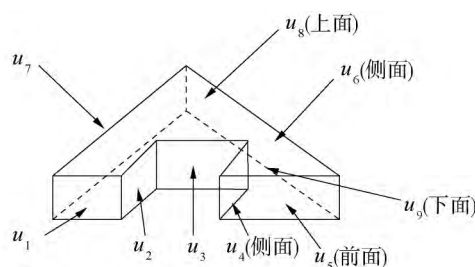


图1 带有槽的三棱柱

Fig. 1 Triangular prism with slot

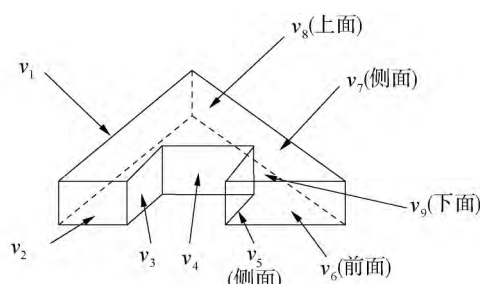


图2 不同面编号的带有槽的三棱柱

Fig. 2 Triangular prism with slot under different face number

如果面 u_s 和 v_t 所含边数差别越小,那么 u_s 与 v_t 之间的相似性就越高, $S_f(u_s, v_t)$ 的值就越大;相反,如果面 u_s 和 v_t 所含边数差别越大,那么 u_s 与 v_t 之间的相似性就越低, $S_f(u_s, v_t)$ 的值也就越小。利用式(1)计算源模型与目标模型各个面之间的形状相似性,并构造面相似度矩阵。此处,使用 m 表示源模型面数,利用 n 表示目标模型面数。搜索面相似度矩阵,能够获得源模型与目标模型之间的最优面匹配序列。为了便于求解,假设 $m \leq n$,即面相似度矩阵的行数要小于等于列数。如果 $m > n$,则将源模型与目标模型互换来进行求解。

源模型与目标模型之间的相似性,不仅取决于各个组成面之间的形状相似程度,而且还取决于各个组成面之间的结构相似程度。在计算源模型面 u_s 与目标模型面 v_t 之间的结构相似性时,需要考虑 u_s 的邻接面和 v_t 的邻接面之间形状相似程度,还需要考虑 u_s 及其邻接面与 v_t 及其邻接面之间邻接顺序对应情况。在模型中,使用邻接矩阵来表示组成面之间的邻接关系:

$$w_{ab} = \begin{cases} 1, & a \text{ 与 } b \text{ 相邻} \\ 0, & a \text{ 与 } b \text{ 不相邻} \end{cases} \quad (2)$$

式中: a 和 b 分别表示模型的2个面。在源模型中,使用如下规则来表示面与面之间的邻接顺序关系:

当 $i \leq s$ 时,表示源模型面 u_i 和 u_s 之间的顺序为正序。当 $i > s$ 时,表示源模型面 u_i 和 u_s 之间的顺序为反序。当 $j \leq t$ 时,表示目标模型面 v_j 和 v_t 之间的顺序为正序。当 $j > t$ 时,表示目标模型面 v_j 和 v_t 之间的顺序为反序。此处,令 $w = w_{u_i u_s} + w_{v_j v_t}$ 。源模型面 u_i 及 u_s 与目标模型面 v_j 及 v_t 之间的邻接顺序对应值 $W(i, s, j, t)$ 的计算过程为:

$$W(i, s, j, t) = \begin{cases} 0.5, & w=1, \text{且}(i < s, j < t \text{ 或 } i > s, j > t) \\ 0.25, & w=2, \text{且}(i > s, j < t \text{ 或 } i < s, j > t) \\ 1, & w=2, \text{且}(i < s, j < t \text{ 或 } i > s, j > t) \\ 0, & \text{其他} \end{cases} \quad (3)$$

源模型面 u_i 与目标模型面 v_j 之间的结构相似性计算过程如下:

$$\text{Sim}_{st}(u_i, v_j) = \sum_{i=1}^n \sum_{j=1}^m S_f(u_i, v_j) \cdot W(i, s, j, t) / (mn) \quad (4)$$

在计算 u_i 与 v_j 的结构相似性时,不仅考虑了 u_i 的所有邻接面与 v_j 的所有邻接面之间的形状对应程度,而且考虑了 u_i 及其邻接面与 v_j 及其邻接面的邻接顺序对应关系。因此,式(4)能够度量源模型面与目标模型面之间的结构相似性。

为了准确地计算源模型面 u_i 与目标模型面 v_j 之间的相似性,需要考虑二者之间的形状和结构相似程度。源模型面 u_i 与目标模型面 v_j 之间的相似性 $S(u_i, v_j)$ 的计算过程为:

$$S(u_i, v_j) = S_f(u_i, v_j) \cdot S_{st}(u_i, v_j) \quad (5)$$

利用式(5)计算源模型与目标模型各个面之间的相似性,并构造源模型与目标模型之间的整体相似度矩阵 S 。

2 基于遗传退火算法的面匹配

遗传算法对离散空间具有较强的全局搜索能力,模拟退火算法具有较好的局部搜索能力。为了提高面匹配的准确性,文本融合这2种算法,提出了基于遗传退火算法的面匹配方法。

1) 染色体编码。

使用二元序对 $(i, j(i))$ 表示源模型面 u_i 与目标模型面 $v_{j(i)}$ 相匹配。对于每个源模型面而言,都可以获得一个二元序对。按照源模型面序号递增的顺序,利用所有的二元序对来构造染色体 $X = ((1, j(1)), (2, j(2)), \dots, (m, j(m)))$ 。对于源模型面 u_i ,有多种面匹配方案。因此,可以获得多条不同的染色体。此处,选取 s 条染色体 $X_t = ((1, j_t(1)), (2, j_t(2)), \dots, (m, j_t(m)))$ ($t=1, 2, \dots, s$)。设定染色体的数量 s 为20。源模型与目标模型的面匹配过程将转化为求最优染色体编码的问题。染色体 X 的适应度值 $F(X)$ 的计算过程为:

$$F(X) = \sum_{i=1}^m S(i, j(i)) \quad (6)$$

2) 选择操作。

在选择最优面匹配序列时,采用轮盘赌比例选择操作,实现方式如下:

利用式(6)分别计算出 $F(X_1)$ 、 $F(X_2)$ 、 \dots 、 $F(X_s)$ 的适应度值。染色体 X_t 的选择概率为:

$$P(X_t) = F(X_t) / \sum_{i=1}^s F(X_i), \quad t = 1, 2, \dots, s \quad (7)$$

产生一个 $[0, 1]$ 的随机数 r 。若 $\sum_{i=0}^{t-1} P(X_i) \leq r <$

$\sum_{i=0}^t P(X_i)$, 则选择染色体 X_t 遗传到下一代。其中 $P(X_0) = 0$ 。

3) 交叉操作。

采用顺序交叉方法,对面匹配序列进行交叉,实现方法如下:

在相邻2个染色体之间,随机选取一段用于产生待交叉序列(待交叉序列的长度为序列长度的 $1/3 \sim 2/3$, 并采用四舍五入方法进行取整),以概率 P_c 产生交叉。交叉操作是使种群多样化的主要因素。

对于面序对 $(i, j(i))$, 定义操作 $c(i, j(i)) = j(i)$, 即取面序对的第2分量。对于染色体 $X = ((1, j(1)), (2, j(2)), \dots, (m, j(m)))$, 定义操作 $C(X) = (c(1, j(1)), c(2, j(2)), \dots, c(m, j(m)))$ 。 $C(X)$ 为从源模型与目标模型的面匹配序列中依次抽取每个面序对的第2分量所形成的向量。对于父代染色体 $X_p = ((1, j_p(1)), (2, j_p(2)), \dots, (m, j_p(m)))$ 和 $X_q = ((1, j_q(1)), (2, j_q(2)), \dots, (m, j_q(m)))$, 可以获得 $C(X_p)$ 和 $C(X_q)$ 。构造子代染色体 $X_{sp} = ((1, j_{sp}(1)), (2, j_{sp}(2)), \dots, (m, j_{sp}(m)))$ 和 $X_{sq} = ((1, j_{sq}(1)), (2, j_{sq}(2)), \dots, (m, j_{sq}(m)))$ 。

所取出的交叉分量为 $X'_{sp} = ((\text{round}(m/3), j_{sp}(\text{round}(m/3))), \dots, (\text{round}(2m/3), j_{sp}(\text{round}(2m/3))))$ 和 $X'_{sq} = ((\text{round}(m/3), j_{sq}(\text{round}(m/3))), \dots, (\text{round}(2m/3), j_{sq}(\text{round}(2m/3))))$ 。互换 X'_{sp} 与 X'_{sq} 所对应面序对的第2分量。 $C(X'_{sp}) = (c(\text{round}(m/3), j_{sp}(\text{round}(m/3))), \dots, c(\text{round}(2m/3), j_{sp}(\text{round}(2m/3))))$ 和 $C(X'_{sq}) = (c(\text{round}(m/3), j_{sq}(\text{round}(m/3))), \dots, c(\text{round}(2m/3), j_{sq}(\text{round}(2m/3))))$ 。 $\text{setdiff}(A, B) = A - B$ 为序列差操作。 $I_{sp} = \text{setdiff}(C(X_p), C(X'_{sq}))$, $I_{sq} = \text{setdiff}(C(X_q), C(X'_{sp}))$ 。将 I_{sp} 中的元素依次插入 X_{sp} 的第 $1 \sim \text{round}(m/3)-1$ 和第 $\text{round}(2m/3)+1 \sim m$ 的面序对的第2分量。将 I_{sq} 中的元素依次插入 X'_{sq} 的第 $1 \sim \text{round}(m/3)-1$ 和第 $\text{round}(2m/3)+1 \sim m$ 的面序对的第2分量。从而获得子代染色体 X_{sp} 和 X_{sq} 。

4) 变异操作。

为了避免陷入局部最优,同时保证解空间的多样性,采用逆转变异方法对面匹配序列进行处理。随机选择染色体 $X_i = ((1, j_i(1)), (2, j_i(2)), \dots, (b, j_i(b)), \dots, (d, j_i(d)), \dots, (m, j_i(m)))$ 。随机选取2个交换位置点 b 和 d , 取出位置点 b 和 d 的面序对 $(b, j_i(b))$ 和 $(d, j_i(d))$ 。将 $(b, j_i(b))$ 和 $(d, j_i(d))$ 的第二分量进行互换。变异后的面匹配序列为 $X_{ii} = ((1, j_i(1)), (2, j_i(2)), \dots, (b, j_i(d)), \dots, (d, j_i(b)), \dots, (m, j_i(m)))$ 。变异操作是保证种群多样性的辅助手段。为了避免损失优良的个体,变异概率 P_d 设定为 0.01。

5) 模拟退火操作。

通常,模拟退火算法在当前解的邻域范围内更新解。本文采用如下方法来产生新解:

在染色体 X_i 中,对基因 $j_i(k)$ 使用 rand 函数产生一个 $[0, 1]$ 之间的随机数 r 。利用下式决定基因 $j_i(k)$ 的移动方向,设定步长为 1:

$$j_i^{t+1}(k) = \begin{cases} j_i^t(k) + 1, & r > \frac{2}{3} \\ j_i^t(k), & \text{其他} \\ j_i^t(k) - 1, & r < \frac{1}{3} \end{cases} \quad (8)$$

式中: $j_i^t(k)$ 为位置 k 上的基因在第 t 次迭代时的数值。经过变换之后,产生的新解为 $j_i^{t+1}(k)$ 。若 $r > 2/3$, 则 $j_i^t(k)$ 的值加 1。若 $r < 1/3$, 则 $j_i^t(k)$ 的值减 1。若 r 为其他值时,则 $j_i^t(k)$ 的值保持不变。当前最优解序列为 $C(X_i) = (j_i(1), \dots, j_i(2), \dots, j_i(b), \dots, j_i(d), \dots, j_i(m))$ 。经过模拟退火操作之后,产生新解 $C(X_z) = (j_z(1), \dots, j_z(2), \dots, j_z(m))$ 。产生新解的次数为 Markov 链的长度 L , 设置 L 的数值为 20。在每次产生新解时,使用 Metropolis 准则来判断是否接受新解 X_{new} , 还是保留原始解 X_{old} 。 X_{old} 为上一次所搜索到的面匹配序列, X_{new} 为当前搜索到的面匹配序列, 即:

$$\begin{cases} X_{\text{old}} = ((1, j_{\text{old}}(1)), (2, j_{\text{old}}(2)), \dots, (m, j_{\text{old}}(m))) \\ X_{\text{new}} = ((1, j_{\text{new}}(1)), (2, j_{\text{new}}(2)), \dots, (m, j_{\text{new}}(m))) \end{cases}$$

将搜索到的面匹配序列 X_{old} 和 X_{new} 代入式(6)计算出对应的相似度数值 $F(X_{\text{old}})$ 和 $F(X_{\text{new}})$, 使用 Metropolis 准则进行判断。

Metropolis 接受准则为:

$$P = \begin{cases} 1, & F(X_{\text{old}}) < F(X_{\text{new}}) \\ \exp\left(-\frac{F(X_{\text{old}}) - F(X_{\text{new}})}{T}\right), & F(X_{\text{old}}) \geq F(X_{\text{new}}) \end{cases} \quad (9)$$

式中: $F(X_{\text{old}})$ 为上一次得到的面匹配序列的适应度值; $F(X_{\text{new}})$ 为新的面匹配序列的适应度值。退火过程采用下式来完成:

$$T_k = QT_{k-1} \quad (10)$$

式中: Q 为退火参数, 设定 Q 的值为 0.95。

在产生新解的过程中, 基因的移动是随机的, 而且是相互独立的。因此, 在所产生的新解中, 会出现 0 元素、越界元素和重复元素。为了解决这一问题, 进行了如下处理:

```

I1 = setdiff(randperm(n), I);
i = 1;
for k = 1: m
    if(I(k) = 0 || I(k) > n)
        I(k) = I1(i);
        i = i + 1;
    end
end
[B, N] = unique(I);
R = setdiff(1: numel(I), N);
I2 = setdiff(randperm(n), unique(I));
for r = 1: length(R)
    for p = 1: m
        if(I(R(r)) = I(p))
            I(p) = I2(r);
            break;
        end
    end
end
end

```

此处, 集合 $I = C(X)$, 即 I 为面匹配序列的第二分量的集合。randperm 函数返回 $1 \sim n$ 的一个随机序列; setdiff(A, B) 为序列差操作, 返回集合 A 与 B 的差集; [B, N] = unique(I) 为去重函数, B 为 I 中不重复元素组成的集合, N 为 B 中元素在 I 中的位置序号的集合; numel(I) 为 I 中元素的个数; length(R) 为集合 R 的长度。

基于遗传退火算法的面匹配:

1) 根据式(5)计算源模型 U 的第 i 个面和目标模型 V 的第 j 个面之间的相似性, 并构造整体相似性矩阵 S 。

2) 初始化温度 $T = 100$, 降温参数 $Q = 0.95$ 、种群规模 $s = 20$ 、交叉概率 $p_c = 0.7$ 、变异概率 $p_d = 0.01$, Markov 链的长度 $L = 20$ 和最大迭代次数 $\text{Maxk} = 100$, 置迭代次数的初始值 $i = 1$ 。

3) 产生初始种群 $G = (X_1, X_2, \dots, X_{20})$, 染色体 $X_t = ((1, j_t(1)), (2, j_t(2)), \dots, (m, j_t(m)))$ ($t = 1, 2, \dots, 20$)。利用式(6)计算染色体 X_t 的适应度值 $F(X_t)$, 并进行选择操作。

4)对种群 G 中的染色体随机配对并执行交叉操作,得到交叉后的种群 G_c 。

5)对交叉后的种群 G_c 中的染色体执行变异操作,得到变异后的种群 G_m 。

6)对种群 G_m 中的染色体执行模拟退火操作,采用 Metropolis 准则判断更新解序列,产生种群 $G_g=(X_1^*, X_2^*, \cdots, X_{20}^*), G=G_g$ 。

7)判断是否满足条件 $i \leq \text{Maxk}$,若满足,则输出最优面匹配序列,否则重复执行步骤 4)~6)。

在源模型 U 和目标模型 V 的整体相似度矩阵 S 中,利用上述算法来搜索二者之间的最优面匹配序列,得到的解向量 $(j(1), j(2), \cdots, j(m))$ 。所对应的最优面匹配序列为 $((1, j(1)), (2, j(2)), \cdots, (m, j(m)))$ 。以最优面匹配序列为基础,从整体相似度矩阵 S 中取出源模型面与目标模型面的相似性。通过累积面的相似性来计算源模型 U 与目标模型 V 之间的相似性,其计算过程为:

$$S_{\text{Model}}(U, V) = \sum_{i=1}^m S(i, j(i)) / n \quad (11)$$

式中: m 表示源模型 U 的面数; n 为目标模型 V 的面数; $(i, j(i))$ 为利用上述算法所获得的最优面匹配序列中的第 i 个序对。

3 实验和结果

本文使用 Matlab 编写基于遗传退火的面匹配算法,对整体相似度矩阵 S 进行搜索。在实验中,目标模型为一个六棱锥,如图 3 所示。同时选取了 15 个 CAD 模型作为源模型,如图 4 所示。

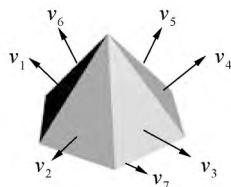


图 3 目标模型
Fig. 3 Target model

使用贪心算法对面相似度矩阵进行搜索,以获得最优面匹配序列。利用蝙蝠算法^[18]、模拟退火算法和本文所提出的方法分别对整体相似度矩阵进行搜索,以获得最优面匹配序列。以最优面匹配序列为基础,使用式(11)计算源模型与目标模型之间的相似性。在 4 组实验中,源模型与目标模型之间的相似性如表 1 所示。

对源模型按照相似性由大到小的顺序进行排序。对贪心算法,源模型排序的结果为 A、L、N、D、J、K、G、B、E、H、C、F、I、M、O。对蝙蝠算法,源模型排序的结果为 A、B、C、D、E、G、J、K、L、H、N、I、F、M、O。对模拟退火算法,源模型排序的结果为 A、B、C、

D、E、G、L、H、K、F、J、I、M、N、O。对本文所提出的方法,源模型排序的结果为 A、B、C、D、E、G、K、J、L、H、I、F、M、N、O。从形状上看,源模型 A 与目标模型相同。贪心算法、蝙蝠算法、模拟退火算法和本文所提出方法都把模型 A 排在了第 1 位。

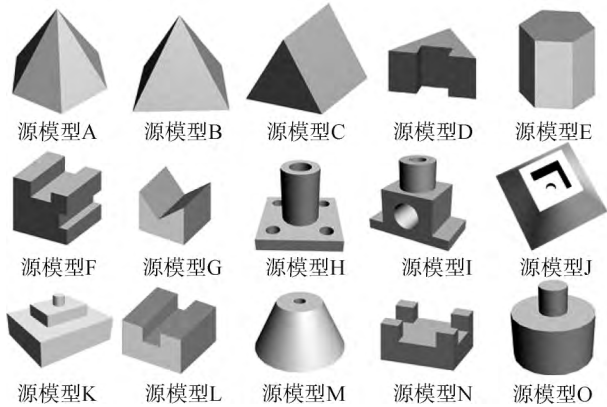


图 4 源模型

Fig. 4 Source model

表 1 源模型与目标模型之间的相似性

Table 1 Similarity between source model and target one

源模型	源模型面数	贪心算法	蝙蝠算法	模拟退火算法	本文提出方法
A	7	1.000 0	0.196 8	0.196 8	0.196 8
B	5	0.678 6	0.155 1	0.155 3	0.155 3
C	5	0.607 1	0.139 4	0.139 4	0.139 4
D	9	0.739 0	0.129 0	0.129 0	0.129 0
E	8	0.678 6	0.117 4	0.119 0	0.118 0
F	18	0.595 7	0.048 6	0.053 5	0.045 2
G	7	0.695 7	0.111 6	0.111 6	0.111 6
H	13	0.667 1	0.073 3	0.070 4	0.071 5
I	15	0.570 0	0.049 4	0.045 1	0.049 7
J	13	0.738 6	0.084 8	0.052 4	0.084 6
K	13	0.738 6	0.083 5	0.053 6	0.086 3
L	10	0.750 0	0.074 4	0.074 4	0.072 9
M	4	0.382 9	0.042 7	0.042 7	0.042 7
N	18	0.750 0	0.052 2	0.037 1	0.038 8
O	5	0.334 3	0.025 6	0.025 6	0.025 6

从形状上看,源模型 B 和 C 与目标模型的相似度很高。蝙蝠算法、模拟退火算法和本文所提出方法都把源模型 B、C 排在第 2 位和第 3 位,而贪心算法却把与目标模型非常不相似的 L、N 2 个模型排在第 2 位和第 3 位。

从形状上看,源模型 D 和 G 与目标模型的相似程度明显不如 B、C 2 个模型。蝙蝠算法、模拟退火算法和本文所提出方法都把模型 D 和 G 排在模型 B、C 之后,而贪心算法把模型 D 和 G 排在模型 B、C 之前。由此可以看出:在计算 2 个模型的相似性时,蝙蝠算法、模拟退火算法和本文所提出方法的性能要好于贪心算法。其原因是:在计算模型相似性时,

蝙蝠算法、模拟退火算法和本文所提出方法不仅考虑了形状信息,而且考虑了结构信息。贪心算法仅仅考虑了形状信息。

对 H、J、K、L4 个模型而言,K 最接近目标模型,J 与目标模型较为相似,H、L 与目标模型的差异较大。贪心算法把 L 排在了 J、K 之前,K 排在了 J 之后。蝙蝠算法把 K 排在了 J 之后。模拟退火算法将 H、L 排在了 J、K 之前。本文所提出方法将 K、J 排在了 H、L 之前,将 K 排在 J 之前。相对于模拟退火算法而言,所提出方法使 13.33% 的模型的排序效果有所改善。相对于模型 N 而言,I、M 更接近于目标模型。蝙蝠算法将模型 N 排在 I、M 之前,模拟退火算法和本文所提出方法都把模型 N 排在 I、M 之后。由此可以看出:在计算 2 个模型的相似性时,本文所提出方法的性能要优于蝙蝠算法和模拟退火算法。其原因是:本文所提出的方法既兼顾了遗传算法的全局搜索能力,又兼顾了模拟退火算法的局部搜索能力。与模拟退火算法和蝙蝠算法相比,本文所提出方法能够更好地计算 2 个模型之间的相似度。

4 结论

1) 本文利用面之间的边数差异构造 2 个模型之间的面相似度矩阵。以此为基础,结合面的邻接关系来计算源模型面与目标模型面的结构相似性。结合面的形状相似性和结构相似性构造整体相似度矩阵。

2) 利用遗传退火算法来搜索整体相似度矩阵,以获得源模型与目标模型之间的最优面匹配序列。以最优面匹配序列为基础,通过累积源模型面与目标模型面之间的相似性度量 2 个模型之间的差异。

3) 在计算模型相似性时,本文所提出的方法要优于蝙蝠算法、模拟退火算法和贪心算法,能够更准确地计算 2 个模型之间的相似性。

参考文献:

- [1] 牟春倩,唐雁,胡金戈. 基于流形排序的三维模型检索方法[J]. 山东大学学报(工学版), 2017, 47(4): 19-24.
- MOU Chunqian, TANG Yan, HU Jing. A new 3D model retrieval method based on manifold ranking[J]. Journal of Shandong University (engineering science), 2017, 47(4): 19-24.
- [2] 樊亚春,谭小慧,周明全,等. 基于局部多尺度的三维模型草图检索方法[J]. 计算机学报, 2017, 40(11): 2448-2465.
- FAN Yachun, TAN Xiaohui, ZHOU Mingquan, et al. A scale invariant local descriptor for sketch based 3d model

- retrieval[J]. Chinese journal of computers, 2017, 40(11): 2448-2465.
- [3] 范菁,李然,董天阳,等. 基于局部视图的三维树木模型递进检索方法[J]. 计算机辅助设计与图形学学报, 2016, 28(1): 162-171.
- FAN Jing, LI Ran, DONG Tianyang, et al. Progressive retrieval method for 3D tree model based on local view[J]. Journal of computer-aided design & computer graphics, 2016, 8(1): 162-171.
- [4] 刘志,尹世超,潘翔,等. 基于特征线条的三维模型检索方法[J]. 计算机辅助设计与图形学学报, 2016, 28(9): 1512-1520.
- LIU Zhi, YIN Shichao, PAN Xiang, et al. 3D model retrieval method based on feature lines[J]. Journal of computer-aided design & computer graphics, 2016, 28(9): 1512-1520.
- [5] 刘楠楠,王洪涛,郭洪斌,等. 基于多模态信息的三维模型检索算法[J]. 南开大学学报(自然科学版), 2017, 50(6): 59-65.
- LIU Nannan, WANG Hongtao, GUO Hongbin, et al. 3D models retrieval algorithm based on multimodal data[J]. Acta Scientiarum Naturalium Universitatis Nankaiensis (natural science edition), 2017, 50(6): 59-65.
- [6] ZHOU Jilai, ZHOU Mingquan, GENG Guohua, et al. A novel 3D model retrieval method based on shape index distribution[J]. International journal of signal processing, image processing and pattern recognition, 2016, 9(9): 325-332.
- [7] 徐静,董雁. 基于区域分割的零件三维模型检索方法[J]. 计算机辅助设计与图形学学报, 2017, 29(5): 929-938.
- XU Jing, DONG Yan. 3D part model retrieval method based on surface region decomposition[J]. Journal of computer-aided design & computer graphics, 2017, 29(5): 929-938.
- [8] 庄廷,张旭堂,侯珍秀. 多特征结合相似度优化的三维工程模型检索算法[J]. 哈尔滨工程大学学报, 2015, 36(5): 720-724.
- ZHUANG Ting, ZHANG Xutang, HOU Zhenxiu. 3D engineering model retrieval algorithm based on multiple features and similarity calculation optimization[J]. Journal of Harbin Engineering University, 2015, 36(5): 720-724.
- [9] 李海生,孙莉,吴晓群,等. 基于模型内二面角分布直方图的非刚性三维模型检索[J]. 计算机辅助设计与图形学学报, 2017, 29(6): 1128-1134.
- LI Haisheng, SUN Li, WU Xiaoqun, et al. Non-rigid 3D shape retrieval based on inner dihedral angle histogram[J]. Journal of computer-aided design & computer graphics, 2017, 29(6): 1128-1134.
- [10] 王新颖,岳远扬. 基于张量子空间的三维模型特征提取及检索方法[J]. 吉林大学学报, 2016, 34(5): 645

- 650.
- WANG Xinying, YUE Yuanyang. Method of 3D model feature extraction and retrieval based on tensor subspace learning[J]. Journal of Jilin University (information science edition), 2016, 34(5): 645-650.
- [11] FURUYA T, OHBUCHI R. Learning part-in-whole relation of 3D shapes for part-based 3D model retrieval[J]. Computer vision and image understanding, 2018, 166: 102-114.
- [12] WANG Jihua, WANG Huayu. A study of 3D model similarity based on surface bipartite graph matching[J]. Engineering computations, 2017, 34(1): 174-188.
- [13] TAO Songqiao. CAD model retrieval based on graduated assignment algorithm[J]. 3D research, 2015, 6(2): 1-11.
- [14] LI Zongmin, WU Zijian, KUANG Zhenzhong, et al. Evidence-based SVM fusion for 3D model retrieval[J]. Multimedia tools and applications, 2014, 72(2): 1731-1749.
- [15] CHEN Zongyao, LIN Weichao, TSAI C F, et al. 3D model retrieval by sample based alignment[J]. Journal of visual communication and image representation, 2016, 40: 721-731.
- [16] 皇甫中民, 张树生, 闫维恒. 面向三维 CAD 模型检索的模型分割方法[J]. 机械科学与技术, 2017, 36(5): 741-748.
- HUANGFU Zhongmin, ZHANG Shusheng, YAN Luoheng. A method of model segmentation for 3D CAD model retrieval[J]. Mechanical science and technology for aerospace engineering, 2017, 36(5): 741-748.
- [17] 于勇, 顾黎, 印璞, 等. MBD 模型本体建模及检索技术研究与应用[J]. 北京航空航天大学学报, 2017, 43(2): 260-269.
- YU Yong, GU Li, YIN Pu, et al. Research and implementation of ontology modeling and retrieval technology of MBD model[J]. Journal of Beijing University of Aeronautics and Astronautics, 2017, 43(2): 260-269.
- [18] 陈育南. 基于蝙蝠算法的三维 CAD 模型相似性计算[D]. 哈尔滨: 哈尔滨理工大学, 2019: 39-49.
- CHEN Yu'nan. Compute similarity of 3D CAD model based on bat algorithm[D]. Harbin: Harbin University of Science and Technology, 2019: 39-49.

本文引用格式:

高雪瑶, 谭涛, 张春祥. 遗传退火算法的模型相似性计算方法[J]. 哈尔滨工程大学学报, 2020, 41(7): 1073-1079.

GAO Xueyao, TAN Tao, ZHANG Chunxiang. Method of calculating model similarity based on genetic annealing algorithm[J]. Journal of Harbin Engineering University, 2020, 41(7): 1073-1079.