3D CAD Model Similarity Computation Based on Particle Swarm Optimization Algorithm (Postprint)

Authors: Gao Xueyao, Chen Yunan, Zhang Chunxiang

Date: 2018-05-18T00:00:00+00:00

Abstract

To more accurately measure the shape differences between two models, we propose a model similarity calculation method based on particle swarm optimization. By utilizing the number of edges comprising each face to construct a face similarity matrix, the particle swarm algorithm is employed to search this matrix and obtain an optimal face matching sequence between the two models. Corresponding face similarity values are then extracted from the face similarity matrix based on this optimal sequence. The overall similarity between models is computed through the accumulation of similarities between individual faces, which serves as the foundation for measuring inter-model differences. Experimental results demonstrate that the proposed method can accurately measure the similarity degree between two models.

Full Text

Preamble

Title: Similarity Calculation of 3D CAD Model Based on Particle Swarm Algorithm

Authors: Gao Xueyao[1], Chen Yunan[1], Zhang Chunxiang[2]
[1]School of Computer Science & Technology, [2]School of Software, Harbin University of Science & Technology, Harbin 150080, China

Abstract: To accurately measure shape differences between two models, this paper proposes a model similarity calculation method based on particle swarm optimization. The method constructs a face similarity matrix using the number of edges comprising each face, then searches this matrix using a particle swarm algorithm to obtain an optimal face matching sequence between the two models.

Based on this optimal sequence, corresponding face similarity values are extracted from the matrix, and the overall similarity between models is computed by accumulating similarities between individual faces, thereby providing a metric for shape differences. Experimental results demonstrate that the proposed method can accurately evaluate the similarity between two models.

Keywords: shape difference; model similarity; particle swarm algorithm; matching sequence
Classification: TP391.7
DOI: 10.3969/j.issn.1001-3695.2017.09.0961

0 Introduction

Model similarity calculation plays a crucial role in 3D CAD model retrieval, directly affecting both efficiency and reliability. Liu Zhi obtained optimal viewpoint sets for 3D models to render hybrid silhouette views, extracting Gabor edge response features to build a feature library from which similar models were retrieved using a bag-of-visual-words approach [?]. Fan Jing employed multi-level local views for compositional description of tree models, using semantic information associated with each view to transform view features and enable coarse-to-fine multi-stage retrieval [?]. Zhuang Ting constructed a distance-cosine two-dimensional grid, building a distance-cosine joint shape distribution matrix for 3D models by counting random points in the grid, using L2 distance between distribution matrices to represent model similarity and a binary particle swarm optimization algorithm to compute inter-model similarity [?]. Zhang Kaixing converted the similarity evaluation problem between two freeform surfaces into one between their local features, extracting SIFT-like operators from curvature cloud maps generated in the parametric domain to assess freeform surface similarity by comparing local feature similarity [?]. Bai Jing unified various nonlinear features as attribute graphs, employing a nonlinear agglomerative hierarchical clustering algorithm for attribute graph clustering while introducing an incremental dynamic classification method to effectively cluster reusable regions in 3D CAD models [?]. Qin Feiwei utilized hierarchical feature ontologies and ontology mapping to generate semantic descriptors, performing ontology-based reasoning to achieve better retrieval performance [?]. Kim Hyungki presented a freehand sketch-based modeling approach, using shape distribution to compare multi-resolution models and simple shape query models to improve retrieval accuracy [?]. Chen Qiang proposed three combined feature descriptors and a class-based feature descriptor for model retrieval [?]. Tao Songqiao segmented face adjacency graphs into convex, concave, and planar regions, representing face regions through region attribute encoding and measuring model similarity by comparing these encodings [?]. Jin Yao derived a 3D version of stretch deformation energy, extending surface parameterization to volume parameterization with fixed boundary conditions to achieve stretch minimization [?]. Huangfu Zhongmin extracted geometric and topological attributes from B-Rep representations, using attribute adjacency graphs to represent 3D CAD models and graph spectra to describe local features, implementing model retrieval through a two-layer search mechanism [?]. Chen Long utilized topological information between elements to generate multiple zero-genus quadrilateral subdomains, achieving subdomain parameterization through boundary interpolation and interior point optimization algorithms [?]. Pan Wanbin propagated shape and positioning information from simple local regions to reusable regions based on correspondence between regions and three-dimensional dimension constraint graphs [?].

This paper constructs a face similarity matrix between source and target models based on differences in the number of edges comprising each face. Using this matrix, a particle swarm algorithm searches for the optimal face matching sequence between the source and target models, which then serves as the basis for calculating inter-model similarity.

1 Face Similarity Calculation Between Source and Target Models

Models are composed of multiple faces, and shape differences between these constituent faces lead to tremendous variation in overall model shape. When calculating similarity between source model A and target model B, we accumulate similarities between their individual faces. If the difference in edge counts between corresponding faces is small, the models exhibit high similarity; if the difference is large, similarity is low. The source model A and target model B are illustrated in [Figure 1: see original paper].

The similarity $S(u_i, v_j)$ between source model face u_i and target model face v_j is calculated using:

$$S(u_i, v_j) = 1 - \frac{|num(u_i) - num(v_j)|}{\max(num(u_i), num(v_j))}$$

where $num(f)$ denotes the number of edges comprising face f, and $\max(x, y)$ represents the maximum of x and y. As shown in the formula, smaller differences in edge counts between two faces indicate greater shape similarity. Using $S(u_i, v_j)$, we construct the face similarity matrix S_{AB} between source model A and target model B, where m represents the number of faces in the source model and n represents the number of faces in the target model. If $m > n$, we swap the source and target models to ensure the matrix rows are always less than or equal to the columns.

The face similarity matrix is structured as follows, where source model face indices $u_1, u_2, \dots, u_m$ serve as row labels and target model face indices $v_1, v_2, \dots, v_n$ serve as column labels. The element at row u_i and column v_j in matrix S_{AB} represents the similarity $S(u_i, v_j)$ between source model face u_i and target model face v_j.

Source model A comprises seven faces: $u_1, u_2, u_3, u_4, u_5, u_6$, and u_7. Face u_1 is adjacent to faces u_2, u_5, u_6, and u_7; face u_2 is adjacent to u_1, u_3, u_6, and u_7; face u_3 is adjacent to u_2, u_4, u_6, and u_7; face u_4 is adjacent to u_3, u_5, u_6, and u_7; face u_5 is adjacent to u_1, u_4, u_6, and u_7; face u_6 is adjacent to u_1, u_2, u_3, u_4, and u_5; and face u_7 is adjacent to u_1, u_2, u_3, u_4, and u_5. Since face u_1 has four adjacent faces, it has four edges. Similarly, faces u_2, u_3, u_4, and u_5 each have four edges, while faces u_6 and u_7 each have five edges.

Target model B also comprises seven faces: $v_1, v_2, v_3, v_4, v_5, v_6$, and v_7. Faces v_1 through v_5 are quadrilaterals with four edges each, while faces v_6 and v_7 are pentagons with five edges each.

The face similarity matrix S_{AB} between source model A and target model B is:

$$S_{AB} = \begin{bmatrix} S(u_1, v_1) & S(u_1, v_2) & \cdots & S(u_1, v_n) \\ S(u_2, v_1) & S(u_2, v_2) & \cdots & S(u_2, v_n) \\ \vdots & \vdots & \ddots & \vdots \\ S(u_m, v_1) & S(u_m, v_2) & \cdots & S(u_m, v_n) \end{bmatrix}$$

In the specific case of source model A and target model B, the matrix becomes:

$$\begin{bmatrix} 1 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 \\ 0.8 & 0.8 & 1 & 0.8 & 0.8 & 0.8 & 0.8 \\ 0.8 & 0.8 & 0.8 & 1 & 0.8 & 0.8 & 0.8 \\ 0.8 & 0.8 & 0.8 & 0.8 & 1 & 0.8 & 0.8 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0.8 \\ 1 & 1 & 1 & 1 & 1 & 0.8 & 1 \end{bmatrix}$$

For example, in source model A, face u_1 is adjacent to faces u_2, u_5, u_6, and u_7, while in target model B, face v_1 is adjacent to faces v_2, v_5, v_6, and v_7. According to formula (1), the value at row u_1, column v_1 in the similarity matrix is 1. Similarly, since face v_7 in target model B is adjacent to faces v_1, v_2, v_3, v_4, and v_5, the value at row u_1, column v_7 is 0.8.

2 Model Face Matching Based on Particle Swarm Algorithm

Greater similarity between constituent faces of two models yields higher overall model similarity. For any pair of models, multiple correspondence schemes exist between their faces. This paper employs a particle swarm algorithm to search the face similarity matrix for the optimal face matching sequence, which is then used to calculate inter-model similarity.

During the particle swarm search process, we define a d-dimensional search space with population size p. Let x_t denote the position of the t-th particle—a vector of column indices found in the similarity matrix—and v_t denote its velocity. In the k-th iteration, $P^k_{best,t}$ represents the individual best position of particle t with fitness value $f_{Pbest,t}$, while G^k_{best} represents the global best position with fitness value f_{Gbest}.

Velocity and position updates follow equations (2) and (3):

$$v_t^{k+1} = w \cdot v_t^k + c_1 \cdot r_1 \cdot (P^k_{best,t} - x_t^k) + c_2 \cdot r_2 \cdot (G^k_{best} - x_t^k)$$

$$x_t^{k+1} = x_t^k + v_t^{k+1}$$

where k is the current iteration number, w is the inertia weight, c_1 and c_2 are learning factors, and r_1 and r_2 are random numbers in $[0, 1]$.

To guide particles toward optimal positions, each particle is assigned a fitness value. The position vector found by particle t in matrix S_{AB} is $x_t = (j(1), j(2), \dots, j(m))$, where $j(i)$ indicates the target face matched with the i-th face of source model A ($i = 1, 2, \dots, m$). The fitness function for particle t is calculated using equation (4):

$$f(t) = \sum_{i=1}^{m} S[i, j(i)]$$

where $S[i, j(i)]$ is the element at row i, column $j(i)$ in matrix S_{AB}.

The particle swarm-based face matching process proceeds as follows:

a) Construct the face similarity matrix S_{AB} between source model A and target model B using equation (1).

b) Initialize population size p, inertia weight w, iteration count s, and learning factors c_1, c_2. Initialize particle positions x_t and velocities v_t ($t = 1, 2, \dots, n$). Set current iteration $k = 0$.

c) Calculate fitness values $f(t)$ for each particle using equation (4).

d) If $k > s$, terminate the algorithm and output G_{best}; otherwise, proceed to step e).

e) If $f(t) < f_{Pbest,t}$, update $f_{Pbest,t} = f(t)$ and $P^k_{best,t} = x_t^k$.

f) If $f_{Pbest,t} < f_{Gbest}$, update $f_{Gbest} = f_{Pbest,t}$ and $G^k_{best} = P^k_{best,t}$.

g) Update velocity v_t^{k+1} and position x_t^{k+1} for each particle using equations (2) and (3).

h) Increment $k = k + 1$ and return to step d).

The particle swarm algorithm searches the model face similarity matrix S_{AB} for the optimal matching sequence, yielding the optimal position solution vector $(j(1), j(2), \dots, j(m))$. This vector indicates that the optimal face matching sequence between source model A and target model B is $(1, j(1)), (2, j(2)), \dots, (m, j(m))$.

Based on this optimal sequence, corresponding elements are extracted from S_{AB}, and the overall similarity $S_{mod}(A, B)$ between models A and B is obtained by accumulating face-level similarities, as calculated using equation (5):

$$S_{mod}(A,B) = \frac{\sum_{i=1}^{m} S[i, j(i)]}{\min(m,n)}$$

where $(i, j(i))$ represents the i-th pair in the optimal face matching sequence, and $\min(m, n)$ is the smaller of m and n.

3 Experiments

The particle swarm algorithm was implemented in MATLAB on a Windows 7 system with an i5 processor. Three CAD models were selected for the experiments, with target and source models shown in [Figure 2: see original paper]. The target model is a quadrilateral pyramid; source model A is a quadrilateral pyramid; source model B is a triangular prism; and source model C is a truncated cone with through holes.

The target model comprises five faces: v_1, v_2, v_3, v_4, and v_5. Face v_1 is adjacent to faces v_2, v_4, and v_5; face v_2 is adjacent to v_1, v_3, and v_5; face v_3 is adjacent to v_2, v_4, and v_5; face v_4 is adjacent to v_1, v_3, and v_5; and face v_5 is adjacent to v_1, v_2, v_3, and v_4. With three adjacent faces, face v_1 has three edges. Similarly, faces v_2, v_3, and v_4 each have three edges, while face v_5 has four edges.

Using source model A as an example, the particle swarm algorithm searches for the optimal face matching sequence between source model A and the target model. Source model A consists of five faces: u_1, u_2, u_3, u_4, and u_5, where faces u_1 through u_4 are triangles with three edges each, and face u_5 is a quadrilateral with four edges.

Using equation (1), we construct the face similarity matrix between source model A and the target model:

$$\begin{bmatrix} 1 & 0.75 & 0.75 & 0.75 & 0.75 \\ 0.75 & 1 & 0.75 & 0.75 & 0.75 \\ 0.75 & 0.75 & 1 & 0.75 & 0.75 \\ 0.75 & 0.75 & 0.75 & 1 & 0.75 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

For instance, in source model A, face u_1 is adjacent to faces u_2, u_4, and u_5, while in the target model, face v_1 is adjacent to faces v_2, v_4, and v_5. According to equation (1), the value at row u_1, column v_1 is 1. In the target model, face v_5 is adjacent to faces v_1, v_2, v_3, and v_4, making the value at row u_1, column v_5 equal to 0.75.

Using this similarity matrix, the particle swarm algorithm searches for the optimal face matching sequence between source model A and the target model, yielding the optimal sequence: $(1, 3), (2, 2), (3, 1), (4, 4), (5, 5)$. The overall similarity between source model A and the target model, calculated using equation (5), is 1.

To validate the proposed method, two comparison experiments were conducted. In these experiments, equation (1) was used to construct face similarity matrices between source and target models. Both greedy and particle swarm algorithms were employed to search for optimal face matching sequences, from which face similarities were extracted and overall model similarity was computed using equation (5).

The similarity results between the target model and source models A, B, and C are presented in .

Table 1 Similarity Between Source Models and Target Model

Source Model	Face Count	Edge Count	Vertex Count	Greedy Algorithm	Particle Swarm Algorithm
Source A	5	16	10	1	1
Source B	5	15	9	0.9	0.9
Source C	6	18	12	0.75	0.8125

Visually, source model A is identical to the target model, yielding a similarity value of 1. Source model B shows considerable similarity with a value of 0.9. Source model C exhibits moderate similarity with a value of 0.8125.

The results indicate that for source model A (identical to the target model) and source model B, both algorithms produce identical results. However, for source model C—a critical case—the particle swarm algorithm achieves a higher similarity value (0.8125) than the greedy algorithm (0.75), representing an 8.33% improvement. This demonstrates that the proposed method more accurately measures similarity between models compared to the greedy approach.

In terms of computational complexity, the greedy algorithm has time complexity $O(mn)$ for searching the face similarity matrix, while the particle swarm algorithm has complexity $O(psmn)$. Although the particle swarm approach is more computationally expensive, it delivers superior performance in measuring model similarity.

4 Conclusion

This paper constructs a face similarity matrix based on differences in face edge counts and employs a particle swarm algorithm to search for optimal face matching sequences between source and target models. By accumulating similarities between matched faces, the method effectively measures overall model similarity. Experimental results demonstrate that the particle swarm algorithm successfully achieves accurate face matching between models and provides precise metrics for shape differences.

References

[1] 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.

[2] Fan Jing, Li Ran, Dong Tianyang. Progressive retrieval method for 3D tree models based on local views [J]. Journal of Computer-Aided Design & Computer Graphics, 2016, 28(1): 162-171.

[3] Zhuang Ting, Zhang Xutang, Hou Zhenxiu. 3D engineering model retrieval algorithm combining multiple features with similarity optimization [J]. Journal of Harbin Engineering University, 2015, 36(5): 720-724.

[4] Zhang Kaixing, Bai Xiaoliang, Zhang Shusheng. Freeform surface similarity evaluation method based on local shape features [J]. Computer Integrated Manufacturing Systems, 2014, 20(3): 530-536.

[5] Bai Jing, Luo Haonan, Qin Feiwei. Clustering of 3D CAD models for nonlinear features [J]. Journal of Computer-Aided Design & Computer Graphics, 2015, 27(8): 1578-1586.

[6] Qin Feiwei, Gao Shuming, Yang Xiaoling, et al. An ontology-based semantic retrieval approach for heterogeneous 3D CAD models [J]. Advanced Engineering Informatics, 2016, 30(4): 751-768.

[7] Kim Hyungki, Cha Moohyun, Mun Duhwan. Shape distribution based retrieval of 3D CAD models at different levels of detail [J]. Multimedia Tools and Applications, 2017, 76(14): 15867-15884.

[8] Chen Qiang, Fang Bin, Yu Yongmei, et al. 3D CAD model retrieval based on the combination of features [J]. Multimedia Tools and Applications, 2015, 74(13): 4907-4925.

[9] Tao Songqiao, Wang Shuting, Chen Anhui. 3D CAD solid model retrieval based on region segmentation [J]. Multimedia Tools and Applications, 2017, 76(1): 103-121.

[10] Jin Yao, Qian Gui Ping, Zhao Jie Yi, et al. Stretch-minimizing volumetric parameterization [J]. Journal of Computer Science and Technology, 2015, 30(3): 553-564.

[11] Huangfu Zhongmin, Zhang Shusheng, Yan Luoheng. 3D CAD model retrieval based on hierarchical feature descriptors [J]. Computer Integrated Manufacturing Systems, 2015, 21(12): 3095-3106.

[12] Chen Long, Fan Xingwang, Wang Meng, et al. Parametric reconstruction of finite element quadrilateral mesh models [J]. Journal of Computer-Aided Design & Computer Graphics, 2017, 29(4): 680-688.

[13] Pan Wanbin, Gao Shuming, Chen Xiang. Local automatic adaptive reuse method for parametric models [J]. Journal of Computer-Aided Design & Computer Graphics, 2016, 28(2): 314-327.

Note: Figure translations are in progress. See original paper for figures.