10-2004

Deformable object model matching by topological and geometric similarity

Kwok-Leung TAN

Rynson W. H. LAU

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Deformable Object Model Matching by Topological and Geometric Similarity

Kwok-Leung Tam [1]   Rynson W.H. Lau [1,2]   Chong-Wah Ngo [1]

[1] Department of Computer Science, City University of Hong Kong, Hong Kong
[2] Department of CEIT, City University of Hong Kong, Hong Kong

Abstract

In this paper, we present a novel method for efficient 3D model comparison. The method is designed to match highly deformed models through capturing two types of information. First, we propose a feature point extraction algorithm, which is based on "Level Set Diagram", to reliably capture the topological points of a general 3D model. These topological points represent the skeletal structure of the model. Second, we also capture both spatial and curvature information, which describes the global surface of a 3D model. This is different from traditional topological 3D matching methods that use only low-dimension local features. Our method can accurately distinguish different types of 3D models even if they have similar topology. By applying the bipartite graph matching technique, our method can achieve a high precision of 0.54 even at a recall rate of 1.0 as demonstrated in our experimental results.

1. Introduction

Due to the popularity of 3D graphics and the high cost of geometry model creation, there is an increasing demand for model sharing. This motivates research on geometry model matching and retrieval and the development of 3D search engines [3]. Existing 3D model matching methods can be categorized into three major domains: geometry-based, frequency-based and topology-based. Among these three approaches, only topology-based matching methods can handle highly deformable models, i.e., models representing the same object but in different postures. However, topology-based methods [5,16] generally suffer from high computation cost. Our method, which combines both topology and geometry features for matching, is designed to handle highly deformable models like [5], but in a novel and more efficient way. Given a 3D model, our feature point extraction method first analyzes the model to obtain two sets of topological points. Instead of using a reeb graph [5] or an explicit skeleton [16], our method uses topological points to represent the skeletal information. Since these features points only depend on model topology and are independent of model tessellation, the number of feature points is thus limited. Apart from the feature point sets, we also capture spatial information and the

surface curvature distribution with respect to each feature point as geometry data. Our experimental results show that the proposed method produces a very good matching result.

The main contributions of this paper can be summarized as follows:

- Our method analyzes the skeletal structure of a 3D model by topological points. This is relatively new among existing methods that analyze models based on reeb graph [5] and explicit skeleton [16].
- Apart from skeleton matching, we also use geometry information to distinguish global surface curvature of models with similar skeleton. This area has not been studied in previous topology methods.
- To capture topological points reliably, we propose a feature point extraction method, which is based on the "Level Set Diagram" [10]. Although the new method is used to extract feature points here, it can be extended to drive skeleton extraction [10] and model compression [15] of general 3D models.

The rest of the paper is organized as follows. Section 2 gives a brief review on previous work. Section 3 presents our method in details. Section 4 discusses our similarity measure. Section 5 presents and evaluates some of the experimental results. Finally, section 6 briefly concludes our work.

2. Related Work

Recently, many methods have been proposed for matching 3D models. They can be roughly categorized into three approaches: geometry-based, frequency-based and topology-based.

Geometry-based methods make use of the geometric properties for model comparison. The geometric properties can be further grouped into three types: physical properties, surface sampling and statistical analysis. Physical properties includes: surface area, volume-area ratio and eigenvalues [20]. In [7], the non-overlapping volumes (MIV and MSV) are proposed to build a hierarchical search tree for indexing. Methods that consider surface properties as matching features include: surface normal vectors [6], cord angles [14], spin images [1] and harmonic shape images [21]. Methods that use statistical features include: D2 (point to point length distribution) [13], linear combination of lengths, curvature and volume

1

distribution [17]. Although all these methods are generally very efficient, they cannot be used for exact matching of similar models and are mainly used for model categorization.

Frequency-based methods convert spatial data into sinusoids of frequencies and amplitudes [9,18]. Though they allows multi-resolution analysis of 3D models, the requirement for pose-normalization accounts for their major limitation because pose-normalization methods (e.g., PCA) may not be stable if there exists more than one major axis in the model. To address this problem, spherical harmonic [3] is proposed to extract features from a set of concentric spheres, and results are shown to be promising among rigid models.

Unlike geometry-based nor frequency-based methods, topology-based methods can be used to match deformable models. These methods consider the model skeletons for matching. In [16], an explicit skeleton matching method is presented. The skeleton is extracted by applying a 3D thinning process on a voxelized model. The requirement of voxelization, however, means that the method is subject to quantization error and high computational and memory costs. In addition, this method may easily mismatch models which represent different objects but have similar skeletons, because it only considers skeletal information for matching.

In [5], geodesic distance is used to construct a multi-resolution reeb graph for matching deformable models. It does not require the voxelization of object. However, it may still be easy to mismatch different models with similar skeletons because it only considers area and length in node matching. To tackle this problem, [19] proposes a hybrid approach to combine topology and geometry information for matching. In order to handle slight topological change, it shoots rays from the center of mass of the model to the surface to form a surface penetration map. However, this method cannot handle highly deformable models because the center of mass of a deformable model may change significantly and even move outside of the model.

Our shape comparison method, which is also based on a hybrid approach, analyzes 3D models in both topology and geometry domains. However, unlike [19], our method emphases on topology analysis and hence, it can handle highly deformable models.

3. Feature Extraction

In this section, we describe in detail our method for features extraction. It involves two main steps. Section 3.1 discusses how we extract topological points from a 3D model, while section 3.2 discusses how we extract

additional geometric information for describing each of the topological points for matching.

3.1 Topological Point Extraction

In [10], Lazarus et al. presented a "Level Set Diagram" (LSD) algorithm to construct a skeleton of a 3D model. The algorithm actually simulates the marching of a waterfront. The word "waterfront" is used from the graphical explanation in [4]. A critical point arise when the topology of a waterfront changes. There are three types of critical points: *minima*, *maxima* and *saddles*. If a waterfront opens or closes, maxima are registered. On the other hand, if waterfronts meet leading to a split or merge, saddles are registered. All waterfronts grow from a single minimum. By analyzing these critical points, the skeleton can be built. We would refer readers to [10,15] for details where [15] is the modified version of [10] to handle genus topology.

Though LSD runs reliably on smooth manifolds, it performs poorly in identifying critical points on general 3D models, which may have arbitrary curvature and probably noise. The waterfronts maintained by LSD can easily be corrupted, leading to incorrect identification of these points. Here, we propose a novel approach to tackle this deficiency.

From our observation, we find that many of these incorrect points are wrongly identified during the registration of saddles. To avoid this problem, we propose to count the number of unvisited vertices adjacent to a waterfront. This can be carried out by running a depth-first search on the waterfront before registering a saddle. We consider three cases below:
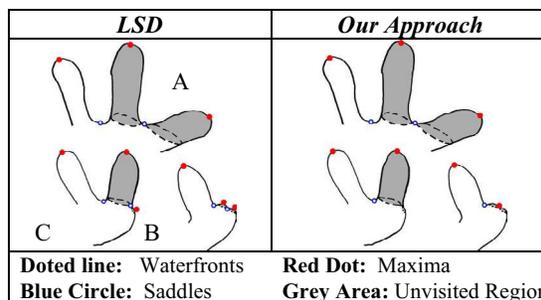

LSD	*Our Approach*
Doted line: Waterfronts	**Red Dot:** Maxima
Blue Circle: Saddles	**Grey Area:** Unvisited Region

Figure 1. Selection of saddles in our method.

- *A Good saddle:* Waterfronts can be uniquely identified and more than one unvisited vertex (the grey region in case A of figure 1) is found adjacent to each waterfront. Since this is considered as a normal case in [10], we register the point as a normal saddle.

2

- *A Bad saddle with good maximum:* No unvisited vertices (the grey region in case B of figure 1) are found adjacent to waterfronts. As such, there is no significant protrusion. We register the point as a maximum and disallow all vertices on the waterfronts to become saddles or maxima.
- *A Bad saddle:* Two or more waterfronts are identified but only one waterfront has at least one adjacent unvisited vertex. This is a degenerate case (case C of figure 1). Our feature extraction method does not register the point as a saddle.

It should be noted that our depth-first search procedure is completely different from the original contour split procedure proposed in [10]. The contour split procedure is applied on the dual graph of the model to analyze its skeleton. Our depth-first search procedure, however, is directly applied on the waterfronts, not the dual graph, and is used to filter noise and solve the problem of incorrect topological point identification.
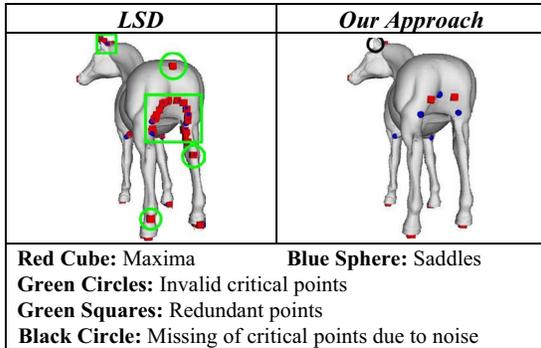

LSD	*Our Approach*

Red Cube: Maxima **Blue Sphere:** Saddles
Green Circles: Invalid critical points
Green Squares: Redundant points
Black Circle: Missing of critical points due to noise

Figure 2. Comparison of LSD and our method.

Figure 2 compares the results of the LSD method with our method. We can see that our method can extract more reliable topological points and is more noise resistant for general 3D models. The deficiency of the LSD method (left diagram of figure 2) is due to the corruption of waterfronts during mesh transversal. Our method (right diagram of figure 2), on the other hand, extracts critical points more accurately and reliably without such a deficiency.

Though our method can reduce most redundant and incorrect points, it sometimes misses very small features. In the black circle (right diagram of figure 2), our method fails to find any points on one of the two ears. The failure is caused by our noise filtering step that two feature points are considered as noise if they are only one vertex away. However, missing of feature points only occurs infrequently, and the error

percentage is far lower than that of the LSD method as will be shown in table 3. By using bipartite matching (detail in section 4.2), the matching error becomes insignificant.

Although the LSD method considers three types of critical points, namely, maxima, minima and saddles, to simplify our feature matching process (which will be shown in section 4), we have grouped maxima and minima into the same set and called them *Local Maximum Points*. Hence, for the rest of the paper, we only consider two types of topological points: *local maximum points* and *saddle points*.

3.2 Geometric Information Extraction

While considering topological points as the skeletal representation (topology information), we use the sum of geodesic distance and the curvature distribution as geometry information.

3.2.1 Sum of Geodesic Distance

The *Sum of Geodesic Distance* was firstly introduced in [5]. Let $g(p,q)$ be the geodesic distance between points $p, q \in S$, where S is the surface of the model. The *Sum of Geodesic Distance*, G, is defined as:

$$G(v) = \int_{p \in S} g(v, p) \partial S$$

To describe the spatial location of a topological point relative to the surface center in a scale-invariant manner, we use the normalized geodesic sum, G_{norm}, as a distance measure.

$$G_{norm}(v) = \frac{G(v) - Min_{q \in S}G(q)}{Max_{q \in S}G(q) - Min_{q \in S}G(q)}$$

where $v \in S$ and $0 \le G_{norm} \le 1$.

It should be noted that, in [5], the *Sum of Geodesic Distance* is used as an index for the sampling interval. However, in our method, it is used as a feature to describe the spatial location of a topological point directly. The normalization formula used is also different. To do this, we find the minimum and the maximum values of G. In [5], these values are approximated by calculating the geodesic sum on a large set of vertex regions. Since most deformable models are high quality meshes, it is common to have vertices count above 10000. The calculation of geodesic sum becomes a very slow process in practice. Thus, it would be advantages if we can limit the number of geodesic sum calculation. We notice that the vertices associated with the maximum and the minimum values of G, in most cases, locate on or near a path between two furthest points. The path has already been found during the preparation stage of LSD [10]. By applying binary search constrained by

3

gradient descent detection, we can easily find an approximation of $Max_{q \in S} G(q)$ and $Min_{q \in S} G(q)$. Among all the models in our database, we find that such an approximation only derivates from the actual value by no more than 3.76% and the number of geodesic sum computations is less than 20.

3.2.2 Curvature Distribution

To describe the global surface curvature change with respect to a topological point t, we use a feature vector $V(t)$ of dimension n to store the curvature information. In order to handle topological change, the curvature information is partitioned according to the geodesic distance as shown in figure 3.
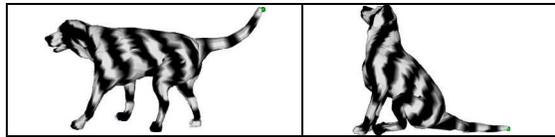

Figure 3. Vertices of two surface meshes are partitioned into black and white bands $P_i(t)$, according to their geodesic distances from the topological point t at the dog's tail (the green dot).

We define $c(t)$ to be the maximum geodesic distance obtained with respect to t:
$$c(t) = Max_{q \in S} \ g(t, q)$$
and let P be a partition of the model surface S with respect to t such that $P_1(t) \cup P_2(t) \cup ... \cup P_n(t) = S$:

$$P_i(t) = \{ q \in S \mid \frac{c(t)}{n}(i-1) \le g(t,q) < \frac{c(t)}{n}(i) \}$$

where $i=1,...,n$. $t, q \in S$ are vertices on the surface and n is the number of partitions.

To approximate the curvature of the model surface, we apply the "Gauss Bonnet" (angle deficit) method [12]. Since the Gaussian curvature $K_G(q)$ at vertex q is only a discrete local approximation, it may be affected by noise or become very large at corners. We normalize the curvature to the range of [0,1] by the following function:

$$\textbf{Normalized } K_G(q) = 1 - \frac{1}{(1+ \mid K_G(q) \mid)}$$

We can then define our feature vector $V(t)$ with respect to t as the sum of Normalized Gaussian curvature in individual partition $P_i(t)$.

$$V_i(t) = \sum\nolimits_{q \in P_i(t)} \textbf{Normalized } K_G(q)$$

where $i=1,...n$. t is a topological point such that $t \in S$. In our prototype system, we set n (the dimension of $V(t)$) to 20 for all the models in our database.

4. Similarity Measure

In this section, we detail the steps for computing the similarity of two objects. This involves two measures. Section 4.1 presents how we compute the similarity of two topological points, while section 4.2 presents how we compute the similarity of two models, which is based on the point set similarity measure.

4.1 Similarity Measure for Topological points

To determine the similarity value of two topological points t_1 and t_2, we use the following formula to combine the normalized geodesic sums G_{norm} and the curvature features V with ratio W_t:

$$Dist_{tp}(t_1, t_2) = W_t \times \mid G_{norm}(t_1) - G_{norm}(t_2) \mid$$
$$+ (1 - W_t) \times L_{2,norm}(V(t_1), V(t_2))$$
$$Sim_{tp}(t_1, t_2) = 1 - Dist_{tp}(t_1, t_2), \quad Sim_{tp} \in [0,1]$$

4.2 Similarity Measure for Models

Since we now have a set of topological points and a similarity value for each point, matching two models can be formulated as a bipartite graph matching problem.

Let $G = (V, E)$ be a *bipartite graph* such that there is a partition $V = A \cup B$ and every edge in E has one endpoint in A and the other in B. Let $M \subseteq E$ be a *matching* such that no vertices are incident to more than one edge in M. We also define the cardinality $|M|$ to be the number of edges in M and let $c : E \to R$ be a cost function on the edge of G. The cost (weight) of a matching is the sum of the cost of its edges, i.e.,

$$c(M) = \sum\nolimits_{e \in M} c(e)$$

The "Maximum Weight Maximum Cardinality Bipartite Matching" (**MWMCB** Matching) problem is thus to find the matching on graph G such that $|M|$ and $c(M)$ are maximized [11], where $c(M)$ is the MWMCB weight.

To allow maximum number of matched topological points and maximum similarity weight between two models, we construct a bipartite graph $G = (A \cup B, E)$, where A and B are the two topological point sets for comparison and Sim_{tp} is the edge weight c. By applying a standard **MWMCB** matching algorithm on G, we can measure the similarity of two point sets. Let n_1 and n_2 be the numbers of vertices in A and B, respectively. The number of matched edges should then be equal to n_1, where $n_1 \le n_2$. Hence, the maximum bound of the MWMCB weight, $c(M)$, is shown as follows:

$$c(M) \le \underset{t_1 \in A, t_2 \in B}{Max} (Sim_{tp}(t_1, t_2)) \times n_1 \le n_1$$

4

The point set similarity is then normalized as follows:

$$Sim_{\text{point set}}(A,B) = \frac{c(M)}{n_2}$$

To increase the matching accuracy, we only allow bipartite matching among the same type of topological points in the two models. That is, we calculate the similarity for the set of *local maximum points,* ($A_{\max}$, $B_{\max}$), and for the set of *saddle points,* (A_{saddle}, B_{saddle}), separately. We then combine them together with ratio W_s as the similarity measure for two models:

$$Sim_{\text{model}}(A,B) = W_S \times Sim_{\text{point set}}(A_{\text{saddle}}, B_{\text{saddle}})$$
$$+(1 - W_S) \times Sim_{\text{point set}}(A_{\max}, B_{\max})$$

5. Experimental Results

We have implemented the proposed method in a prototype system to test its performance. To demonstrate its ability to handle deformable models, we have chosen 10 base models from a popular industrial modeling tool named "Poser". For each of these 10 base models, we generate 3 more different poses as deformed models. Incorporating some models downloaded from the Internet, we have created a database of 70 different models. For each of these models, we generate an addition of 7 models by rotating them along the *xy* axis, *yz* axis, non-uniform scaling along the *xy* axis, *yz* axis, and reducing their vertex counts by 25%, 50% and 75%. To create the reference database, we manually categorize the 560 models into 25 groups. Some of the sample models are shown in figure 5.

5.1 Performance Evaluation

Figure 4 shows the Precision-Recall graph. We use each of the 560 models as a query to the database (W_t=0.5 and W_s=0.5) and average the matching results. From the plot, we can see that our method can achieve a high precision value of 0.54 even at high recall value of 1.0. The features captured are found to be invariant to rotation, non-uniform scaling, multi-resolution and pose deformation. As a comparison, we have also plotted in figure 4 the precision and recall of a geometry-based method, D2 [13] and a frequency-based method, Fourier [18]. They are two of the best methods that we have tested in our earlier work [9]. The results indicate that our method performs much better than [13,18], which cannot handle deformable models. We can see that the precision of the geometry-based and the frequency-based methods drops dramatically when the recall is above 0.1.

Since our method depends only on model topology but not on model tessellation, the number of topological points captured is relatively small (minimum 2, mean 18, maximum 60 topological points for our model database), and the number of geodesic sum computations is thus limited. On the contrary, the voxelization process in [16] involves high computational and memory costs, while [5] uses interval sampling which requires a large number of geodesic sum computations.
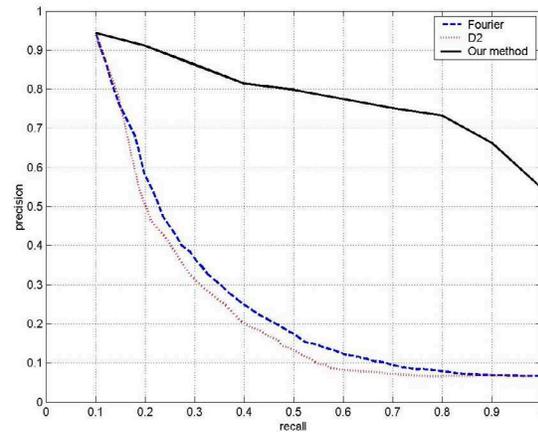

Figure 4. Plot of Precision and Recall.

5.2 Matching Accuracy

5.2.1 Matching of Models with Similar Skeletons

Table 1 shows the similarity ratings of our method on models with similar skeletons. We can see that our method can distinguish models with very similar skeletons. This can be explained by the use of additional spatial and curvature features. For the case of cats and dogs, though the numbers of topological points captured are the same, i.e., they have similar skeletons, they have relatively different similarity rating. The difference is mainly contributed by the spatial information and surface curvature distribution features. Our observation here is that dogs have longer limbs than cats. The spatial locations of topological points for dogs located at limbs are thus further away from the surface center than those of the cats. In addition, since dogs have a larger body than cats, the curvature distributions are also different.

5.2.2 Matching of Models with Dissimilar Skeletons

Table 2 shows the similarity ratings of our method on models with dissimilar skeletons. We can see that the similarity values differ significantly. For the case of dogs and humans, the similarity rating is much smaller than the case of cats and dogs. This can be accounted for by two factors, the topological and geometric effects. From our observation, there are many fingers

5

and toes in human models while there are none in dogs. Hence, the skeletal (topological) features are different. For geometric effects, we observe that humans have very long arms and legs, but have relatively shorter body than those of the dogs. Such differences not only affect the spatial location (geodesic sum) of the topological points, but also the curvature distribution features. All these dissimilarities are reflected in the larger difference in the similarity table.

Table 1. Similarity of models with similar skeletons.

Mean Similarity	Cats	Dogs	Horses	Dolphins
Cats	<u>0.91</u>	0.85	0.81	0.78
Dogs	0.85	<u>0.95</u>	0.83	0.81
Horses	0.82	0.83	<u>0.92</u>	0.71
Dolphins	0.78	0.81	0.71	<u>0.93</u>

Table 2. Similarity of models with dissimilar skeletons.

Mean Similarity	Frogs	Humans	Dogs
Frogs	<u>0.95</u>	0.62	0.42
Humans	0.63	<u>0.88</u>	0.31
Dogs	0.42	0.31	<u>0.95</u>

(Note that in tables 1 and 2, all the models within each group are a series of deformable models.)

5.3 LSD and Our Feature Extraction Algorithm

To study the reliability of our feature point extraction algorithm for extracting topological points, we pick some models from our model database for comparison. In table 3, alternating columns compare the topological points captured by the LSD method and our feature point extraction algorithm. We can see that the LSD method captures a large number of redundant and incorrect topological points. Since these invalid points spread across a large region, it may be difficult to solve the problem through clustering (e.g., the first horse on the 3rd row). On the other hand, our method captures topological points precisely and most of them agree with human inspection. Most saddle points are located at the articulate joints while the local maximum points are located at the tips of a protrusion. For the example of noisy sphere (i.e., the right epcot on the 3th row), our method is able to correctly return the local maximum points at the two tips, which match the topology of a sphere. Overall, our method works better than LSD in capturing topological points on general 3D models.

5.4 Complexity

Apart from having a high matching accuracy, our method also has a relatively low computational cost. In

our topological point extraction algorithm, we modified the LSD method by introducing a depth-first search in the pre-registration of saddle points. The LSD algorithm [10] (without contour split) requires $O(n\log n + e)$, where n is the total number of vertices and e is the number of edges. To apply depth-first search for checking valid saddles or maxima, it requires $O(\lambda_S(n_v + n_e))$, where n_v and n_e are the numbers of vertices and edges, respectively, explored during the depth-first search. λ_S is the total number of saddle points found. Hence, the overall complexity is $O(n\log n + e + \lambda_S(n_v + n_e))$.

Compared with [5], although our method also needs to compute the expensive geodesic sum, we only apply it on topologically important feature points. The total complexity of computing the geodesic sum for the whole model is $O((\mu_{max} + \mu_{saddle} + \beta)(n\log n + e))$, where μ_{max} and μ_{saddle} are the numbers of *local maximum points* and *saddle points*, respectively. β is the number of binary search iterations (usually, less than 20) required to find $Min_{q \in S}G(q)$. It should be noted that $\mu_{max} + \mu_{saddle} + \beta$ is much smaller than the total number of base vertices required by [5]. For example, we consider the most complicated human model where $\mu_{max} + \mu_{saddle} = 60$ and $\beta = 20$, making a total of 80. Whereas in [5], about 150 base vertices and their geodesic sums are computed. These geodesic sums are then interpolated to become those of the other model vertices. Hence, the interpolated values associated with the other model vertices are only an approximation from those of the base vertices. Our method, on the other hand, computes the geodesic sum directly on all topological points and is thus believed to be more accurate. Unfortunately, we do not have their program to verify this.

To compare two models, we apply the **MCWMB** matching algorithm. This algorithm has a complexity of $O(N(M + N\log N))$, where $N=|A| + |B|$, $M=|A| \times |B|$, and $|A|$, $|B|$ are the numbers of vertices in the two topological point sets A and B, as discussed in section 4.2. For [5], the complexity of comparing two models is $O(N_R(M_R + N_R))$, where N_R, M_R are the R-nodes counts of the two models. Although the complexity of our method is higher than [5] here, it should be noted that the number of matching points (i.e., the number of topological points) is less than that of [5]. In our method, feature points are dependent on model topology only, not on tessellation, and thus the number of matching points is limited. For comparison, our method has on average 18 topological points for each model of our database, while the method in [5] has

6

about 300 r-nodes for each model. To give some statistics on matching time, we have maximum, minimum and average matching times of 110ms, <0.5ms and 0.71ms, respectively, on a PIII-500MHz machine. By reducing the number of geodesic sum computations (which consume 90% of the cost of the feature extraction process) and the number of nodes in the matching process, our method can support faster content-base retrieval and allow users to submit their query model on the web environment.

We would like to point out that when a 3D model contains more and more topological and/or geometric details, the reeb graph based matching methods performs less accurate [2]. The multi-resolution version of reeb graph [5], which follows topological alignment of a model, is not a skeleton. It stores no explicit topological information and not even the shape of the model. For geometric information, area and length are the only features used. These features are relatively local, and have no specific information to describe the model surface. To better tackle such limitation, we derive our topological points from LSD. LSD not only represents the actual skeleton, but also gives intrinsic topological properties, like genus number from its structure. Further, by grouping feature points into *local maximum points* (protrusion tips) and *saddle points* (branches or handles (genus)), our method can better represent the model. To distinguish similar models of different shapes or poses, features that can describe the overall surface must be used. In our method, the additional curvature feature vectors are especially reserved for such purpose. By capturing additional topological and geometric features from both domains, we believe that our method can out-perform [5].

6. Conclusion

This paper proposes a novel 3D model matching method through analyzing the models in both topological and geometric domains. Unlike existing topology methods [5,16], we use topological points to represent the skeletal information. As demonstrated in our experimental results, the new method has a very high accuracy in matching highly deformable models, invariant to rotation, non-uniform scaling and multi-resolution. The precision rate reaches as high as 0.54 even at a recall rate of 1.0.

Acknowledgements

References

[1] P. de Alarcon et al., "Spin Images and Neural Networks for Efficient Content-Based Retrieval in 3D Object Databases," *Proc. CIVR*, pp. 225-234, 2002.

[2] D. Bespalov et al., "Scale-Space Representation of 3D Models and Topological Matching" *Proc. ACM Symp. on Solid Modeling and Apps.*, pp.208-15, June 2003.

[3] T. Funkhouser et al., "A Search Engine for 3D Models," *ACM Trans. on Graphics*, **22**(1):83-105, Jan 2003

[4] J. Hart, "Computational Topology for Shape Modeling," *Proc. Shape Modeling*, pp. 36-45, 1999.

[5] M. Hilaga et al., "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," *Proc. ACM SIGGRAPH*, pp. 203-212, Aug. 2001.

[6] B. Horn, "Extended Gaussian images," *Proc. IEEE*, 72(12):1671-1686, Dec. 1984.

[7] D. Keim, "Efficient Geometry-based Similarity Search of 3D Spatial Databases", *Proc. ACM SIGMOD*, pp. 419-430, 1999.

[9] R.W.H. Lau and B. Wong, "Web-Based 3D Geometry Model Retrieval," *World Wide Web Journal*, Kluwer Academic Publishers, **5**(3):193-206, 2002.

[10] F. Lazarus and A. Verroust, "Level Set Diagrams of Polyhedral Objects," *Proc. ACM Solid Modeling and Applications*, pp.130-140, 1999.

[11] K. Mehlhorn and S. Naher, *LEDA: A Platform for Combinatorial and Geometric Computing*, Cambridge University Press, 1999.

[12] M. Meyer et al., "Discrete Differential-Geometry Operators for Triangulated 2-Manifolds," *Proc. VisMath*, May 2002.

[13] R. Osada et al., "Matching 3D Models with Shape Distributions," *Proc. Shape Modeling*, May 2001.

[14] E. Paquet and M. Rioux, "Content-based Access of VRML Libraries," *Proc. IAPR, Springer, LNCS 1464*, pp. 20-32, Aug. 1998.

[15] D. Steiner and A. Fischer, "Cutting 3D Freeform Objects with Genus-n into Single Boundary Surfaces using Topological Graphs," *Proc. ACM Solid Modeling and Apps.*, 2002.

[16] H. Sundar et al., "Skeleton Based Shape Matching and Retrieval," *Proc. Shape Modeling and Apps*, 2003.

[17] J. Vandeborre et al., "A Practical Approach for 3D Model Indexing by combining Local and Global Invariants," *Proc. Int'l Symp. on 3D Data Processing Visualization and Transmission*, Jun 2002

[18] D. Vranic and D. Saupe, "3D Shape Descriptor based on 3D Fourier Transform," *Proc. EURASIP Conf. on Digital Signal Processing for Multimedia Communications and Services*, pp. 271–274, 2001.

[19] M. Yu et al., "3D Model Retrieval with Morphing-based Geometric and Topological Feature Maps," *Proc. IEEE CVPR*, 2003.

[20] C. Zhang and T. Chen, "Efficient Feature Extraction for 2D/3D Objects in Mesh Representation," *Proc. IEEE ICIP*, Oct. 2001.

[21] D. Zhang et al., "Harmonic Maps and Their Applications in Surface Matching," *Proc. IEEE CVPR*, Vol. 2, 1999.
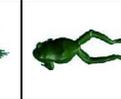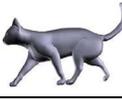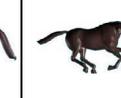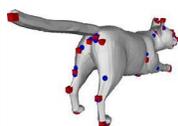
human							
	boys			girls			
	Query: 1.0	0.947	0.945	0.908	0.89	0.88	0.82	0.79
dog				frog			
	Query: 1.0	0.97	0.929	0.928	Query: 1.0	0.977	0.904	0.899
cat				hand			
	Query: 1.0	0.94	0.93	0.82	Query: 1.0	0.92	0.90	0.77
Raptor				Horse			
	Query: 1.0	0.96	0.95	0.90	Query: 1.0	0.95	0.92	0.83

Figure 5. Some example model groups and similarity values.

Table 3. Comparison between LSD and our feature point extraction algorithm.

LSD	Topological points*	*Our Approach*	Topological points*	LSD	Topological points*	*Our Approach*	Topological points*
	Total: 16 C: 76 I: 16 M: 0 R: 44		Total: 16 C: 16 I: 0 M: 0 R: 0		Total: 16 C: 30 I: 3 M: 0 R: 11		Total: 16 C: 16 I: 0 M: 0 R: 0
	Total: 16 C: 46 I: 7 M: 0 R: 23		Total: 16 C: 16 I: 0 M: 0 R: 0		Total: 15 C: 58 I: 9 M: 0 R: 34		Total: 15 C: 14 I: 0 M: 1 R: 0
	Total: 16 C: 84 I: 6 M: 0 R: 62		Total: 16 C: 14 I: 0 M: 2 R: 0		Total: 2 C: 6 I: 0 M: 0 R: 4		Total: 2 C: 2 I: 0 M: 0 R: 0
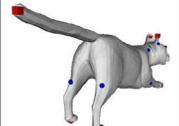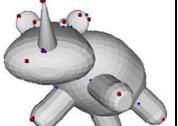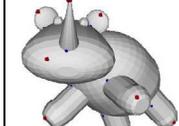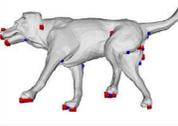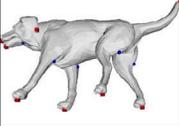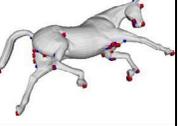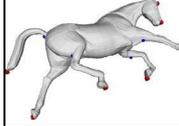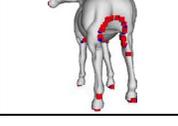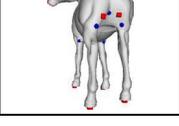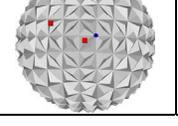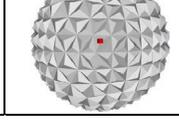
Blue Sphere: Saddle points   **Red Cube:** Local Maximum

Captured (C): Number of topological points captured by algorithm

Incorrect (I): Number of incorrect points far away from topological location (e.g., articulate joints, tips of protrusion)

Missing (M): Number of topological points that algorithm fail to capture

Redundant (R): Redundant points locate near topological location captured by the algorithm

(*) Topological Points include both local maximum and saddle points

8