Retrieval of Non-rigid 3D Models Based on Approximated Topological Structure and Local Volume

Yiyu Hong[1] and Jongweon Kim[2]
[1] Department of Copyright Protection, Sangmyung University
Seoul, 03015 – South Korea
[e-mail: hongyiyu@cclabs.kr]
[2] Department of Electronics Engineering, Sangmyung University
Seoul, 03015 – South Korea
[e-mail: jwkim@smu.ac.kr]
*Corresponding author: Jongweon Kim

Received January 22, 2017; revised March 19, 2017; accepted April 9, 2017; published August 31, 2017

Abstract

With the increasing popularity of 3D technology such as 3D printing, 3D modeling, etc., there is a growing need to search for similar models on the internet. Matching non-rigid shapes has become an active research field in computer graphics. In this paper, we present an efficient and effective non-rigid model retrieval method based on topological structure and local volume. The integral geodesic distances are first calculated for each vertex on a mesh to construct the topological structure. Next, each node on the topological structure is assigned a local volume that is calculated using the shape diameter function (SDF). Finally, we utilize the Hungarian algorithm to measure similarity between two non-rigid models. Experimental results on the latest benchmark (SHREC' 15 Non-rigid 3D Shape Retrieval) demonstrate that our method works well compared to the state-of-the-art.

Keywords: Non-rigid model, Integral geodesic distance, Shape Diameter Function, The Hungarian algorithm

1. Introduction

The rapid development of 3D technology (3D printing, 3D scanning, 3D modeling, etc.) and computer networks have led to 3D models being widely used in many fields. Considering that designing and creating a 3D model is not simple, retrieving 3D models accurately and quickly from a huge database is becoming more desirable. There are also issues [1-2] concerning copyright protection of 3D models. High Accuracy 3D model retrieving technology would benefit the interest of copyright holders by intercepting illegally distributed 3D models.

In the beginning of 3D shape retrieval, most efforts were focused on retrieval methods for rigid 3D models. However, recently retrieval methods for non-rigid 3D models that may require more shape analysis, have been an active area of computer graphics research. As shown in **Fig. 1**, non-rigid 3D models indicate that, with different poses or articulations, the human and hand models in each row are in the same category.

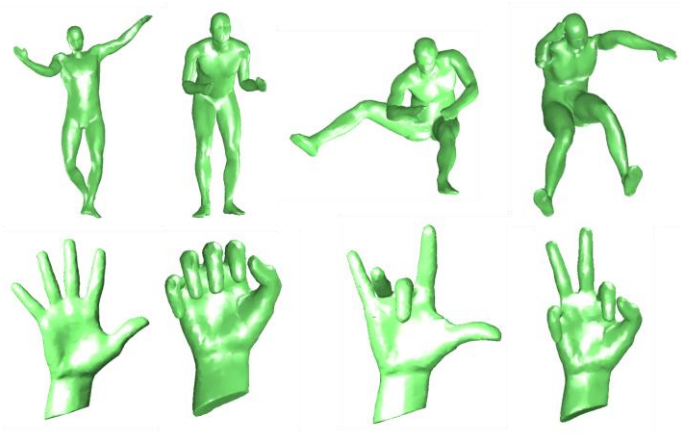

Fig. 1. Non-rigid models

To appropriately compare two non-rigid models, shape descriptors must be invariant to non-rigid bending and articulations. In this paper, we utilize two characteristics of non-rigid models to measure dissimilarity between the models. The first characteristic is geodesic distance and path that means the shortest distance and path between two vertices on the mesh surface. **Fig. 2 (a)** shows that the distance and path on the mesh between two pose-deformed models are nearly unchanged. The second characteristic is local volume on the corresponding

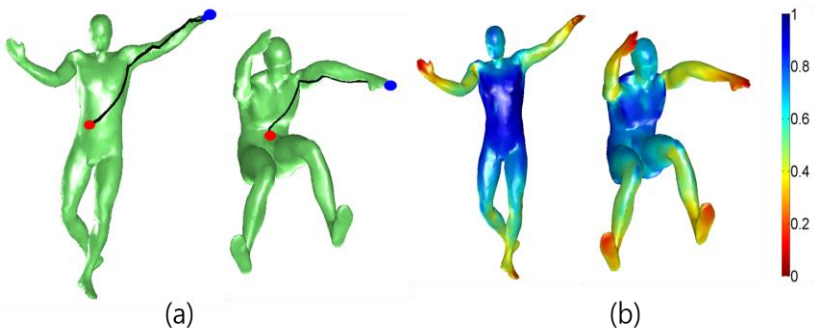

Fig. 2. (a) Geodesic path (b) Local volume

position between two non-rigid models. In **Fig. 2 (b)**, the color on the models indicates local volume calculated by SDF [3]. We can see that the local volume in the corresponding positions is almost similar.

2. Related Work

During the past few years, several algorithms [4-17, 30, 31] have been proposed for 3D shape retrieval. Existing methods can be divided into two main types: retrieval methods for rigid and non-rigid 3D models. For rigid model retrieval, there are algorithms based on 2D views [5-6, 9], spectral transformation [7], topology [4], statistics [8], etc. For more details about these algorithms, we refer readers to [22-23]. Retrieval approaches for non-rigid models are extensions of algorithms for rigid model retrieval. The extension requires extracted features from models to be isometry-invariant. For example, Lian et al. in [9] extended a 2D-view based rigid model retrieval method [10] that they had proposed before, to use for non-rigid model retrieval by first utilizing multidimensional scaling on the 3D model to obtain its bending invariant representation. Readers can refer to [24-26] for a good comparison of methods for non-rigid 3D shape retrieval.

There are several algorithms (ShapeDNA [13], Heat Kernel Signature [14], Wave Kernel Signature [15], etc.) based on eigenvalues and associated eigenfunctions of the Laplace-Beltrami operator to construct spectral shape descriptors that depend on mesh surface. The Laplace-Beltrami operator [13] provides good properties of translation invariance, rotation invariance, isometric invariance and optionally scaling invariance. A comparative survey of these types of algorithms can be found in [16].

One intuitive approach for non-rigid 3D model retrieval is comparison of topological structure and the corresponding geometric features between two non-rigid 3D models. Hilaga et al. [4] presented multi-resolution reeb graph that is a topology construction method based on geodesic distance and reeb graph theory. The topology matching used coarse-to-fine strategy to search the node pairs that have maximum similarity. However, these types of topology construction and similarity measurement algorithms must satisfy many conditions and therefore cannot achieve good performance. Sfikas et al. [11] proposed a conformal factor guided topological structure construction algorithm. Conformal factor is primarily based on curvature that can easily be affected by geometric noise.

Gal [17] proposed a 2D histogram based pose-oblivious shape signature that combines two scalar functions defined on the surface of a 3D model. The first function is called a local-diameter function that can measure local volume of a 3D model. In the following study [3], they modified this function and renamed it SDF that is used consistently in mesh partitioning and skeletonization. The second function is a centricity function that measures the integral geodesic distances for the entire 3D model.

We propose an efficient and effective approach for non-rigid 3D model retrieval that is largely based on two pose invariant features: geodesic distance and local volume. Our contributions in this paper can be summarized as follows.

- Based on integral geodesic, we propose a simple and effective method for constructing topological structure of 3D models.
- The SDF algorithm is modified to works more effectively and efficiently on our non-rigid 3D models retrieval method.
- We modified the penalizing scheme of the 3D model matching algorithm that used in [11] to make the matching algorithm more reasonable

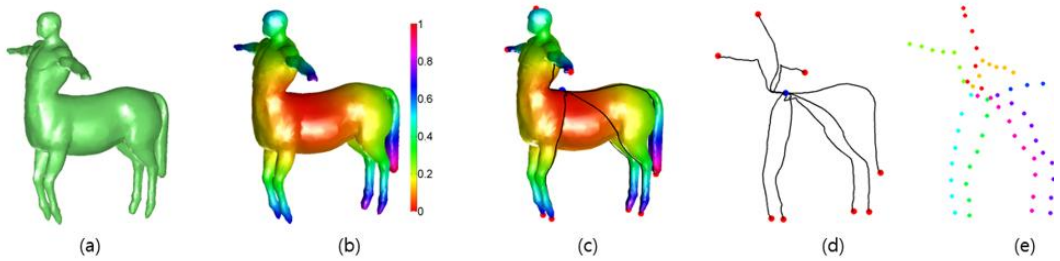

Fig. 3. Overall topology construction process (a) Original model (b) Color-coding of integral geodesic distances of the model (c) Shortest paths from each protrusion tip to surface center (d) Topological structure (e) Selected topological nodes.

3. Method description

3.1 Construction of Topological Structure

Our algorithm for construction of topological structure requires four steps. First, integral geodesic distances are calculated for every vertex on the mesh. Second, we extract vertices that reside on tips of protrusions and vertexes on the center of surfaces using integral geodesic distances. Third, the protrusion tips and vertex on the surface center are connected by finding the shortest geodesic paths. Finally, we sample points on the geodesic paths to extract topological nodes. **Fig. 3** illustrates the overall topology construction process, and the process is discussed in detail below.

Integral geodesic distances were first proposed by Hilaga et al. [4]. The discrete case can be defined as follows:

$$IGD(p) = \sum_{q \in S} g(p, q) \tag{1}$$

where $g(p, q)$ denotes the shortest geodesic distance between vertex p and q. Therefore, $IGD(p)$ is the integral of all geodesic distances from p to all vertices q on a surface S. In our approach, all geodesic distances and paths are computed by the fast marching method [19-20]. **Fig. 3 (b)** shows color-coding of integral geodesic distances of the model. Generally, the vertex that has the minimum integral geodesic distance will reside on the center of a surface, and the vertices that are farther from the center of the surface will have a larger scalar value of integral geodesic distance. Using this property of integral geodesic distance, we can extract vertices on the tips of protrusions by measuring whether the scalar value of integral geodesic distance of a vertex is the local maxima within a radius of a geodesic neighborhood [4, 18]. In this paper, the radius of a geodesic neighborhood is called R_g. A smaller R_g will extract tip-vertex on small parts of the 3D model, such as each finger of a human model. A larger R_g will more likely extract tip-vertex from a large protrusion of a 3D model, such as an arm or leg of a human model that is more robust to geometric noise. In **Fig. 3 (c)**, the blue point and the red points represent the surface center and extracted protrusion tips in the case where R_g is set as $\sqrt{0.005 * area(S)} * 3$. The term $\sqrt{0.005 * area(S)}$ was introduced in [4] and is used as a threshold to divide mesh faces into groups, so we multiply a scalar value to change the scale of R_g.

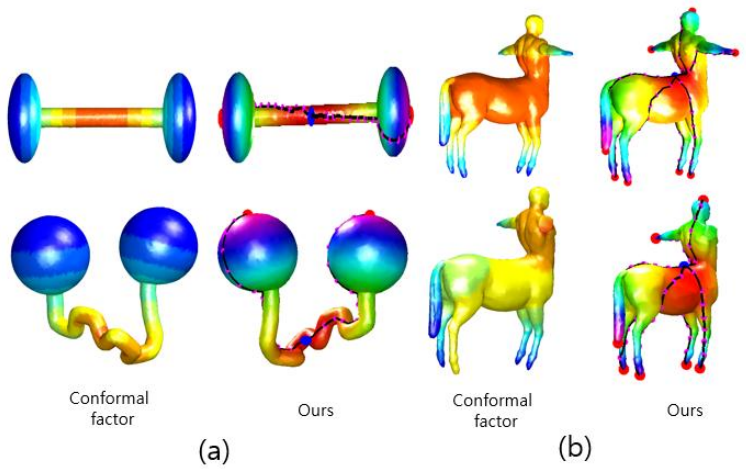

Fig. 4. Comparison between conformal factor and our topological structure

To construct a topological structure simply and effectively, we found that connecting protrusion tips and the surface center on the mesh surface can approximate the topology of a model without a complex process (**Fig. 3 (c)**). The connection can be made by finding the shortest paths from each protrusion tip to the surface center using the fast marching method. In **Fig. 3 (c)**, the black lines represent the shortest paths. For better presentation, we show the topological structure alone in **Fig. 3 (d)**. Every path from a protrusion tip to the surface center is called a *topological path* in this paper.

After constructing the topological structure, we select topological nodes that can represent corresponding sub-parts of the 3D model. On every topological path, we choose points from the protrusion tip to the surface center using a geodesic distance interval T_n, and consider them as topological nodes. In **Fig. 3 (e)**, the points with the same color represent selected topological nodes on the same topological path when $T_n = \sqrt{0.005 * area(S)} * 0.75$. We define the topological nodes on the same topological path in order from the protrusion tip to the surface center as a *topological string*.

In **Fig. 4**, we compare conformal factor [11-12] with our topological structure. The conformal factor is produced by the source code [29] on the internet. As in [11], we quantized the conformal factor into eight levels. From **Fig. 4 (a)**, we can see that the conformal factor is not consistent between the two two-ball models at the distorted location while the integral geodesic shows the consistency which leads to stability of our topological structure. In **Fig. 4(b)**, we cut off an arm of the centaur model; from the picture, we can see that the conformal factor completely lost consistency in contrast to the integral geodesic.

3.2 SDF Values Assignment

Shapira et al. in [3] introduced the SDF that is a scalar function defined on the mesh surface to measure the local shape volume of a 3D mesh. For a given face of a mesh, the SDF sends cone-shaped rays (**Fig. 5(a)**) from the centroid of a face to its normal-opposite side (inward direction on the mesh). Then, the length of a ray can be calculated by checking the ray-mesh intersections, and the scalar value of the SDF for the face calculated as the weighted average of all ray lengths.

In our implementation, SDF is computed using a cone of angle 120° with 30 rays. We do not calculate SDF values for every face on the mesh, we only care about faces that are near the

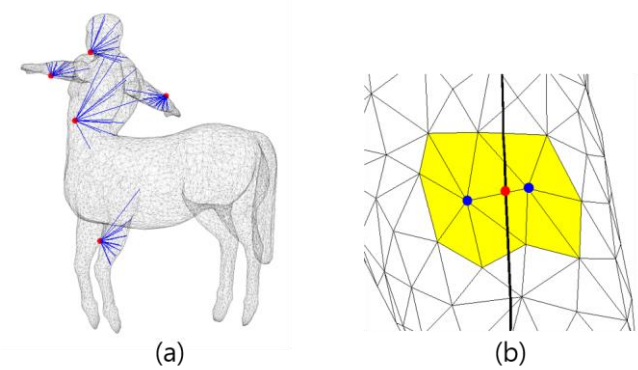

Fig. 5. (a) Cone-shaped rays sent to the inside of the mesh (b) Faces related to SDF values assignment.

topological nodes. As shown in **Fig. 5(b)**, assume that the red point is a topological node that we selected on a topological path (thick black line), and then we find the one vertex-ring face (yellow face) of the two blue points that construct the edge where the topological node resides. The topological node is assigned the average SDF value of these faces.

After calculating all SDF values for topological nodes, to be compatible with 3D meshes in different scales and resolution, the SDF values are normalized as follow:

$$nsdf(TN) = \frac{sdf(TN)}{\sqrt{area(S)}} \tag{2}$$

Where $sdf(TN)$ and $nsdf(TN)$ denote the original SDF value and the normalized SDF value for topological node s, and S denotes the surface of the mesh. Instead of using a logarithmized version [3], we normalize the SDF value by dividing it with the root area of the mesh. In reference [3], the reason for a normalized SDF value in log-space is that the author wants to enhance the importance of the small parts that have small SDF values, like fingers of a human model, to perform good segmentation of a 3D mesh. In our approach, our goal is not to do 3D segmentation. We did test with two normalizing functions, the result in section 4 shows that our method works better.

3.3 Matching Approach

In the matching approach, we first calculate all dissimilarity distances among topological strings with node-by-node SDF values of two 3D models. Then, the Hungarian algorithm is used to find "minimum matching". The Hungarian algorithm [21] is a combinatorial optimization algorithm that solves the assignment problem. Our matching approach is similar to [11] but different in the penalizing method.

In calculating dissimilarity between two topological strings, if two topological strings have the same number of topological nodes, the dissimilarity can be calculated by averaging the differences between the corresponding SDF scalar values. If two topological strings have different lengths, we first append the shorter topological string with its last topological node to have the same length as the longer one. Then, we penalize these appended values by weighting. Let p and q be two topological strings, and let $p[i].sdf$ denote the SDF value of the ith topological node starting from the protrusion tip on p. Assuming that p has more topological nodes than q, the dissimilarity between two topological strings is defined as:

$$Dis(p,q)=\frac{\begin{pmatrix}\sum_{k=1}^{len(q)}|p[k].sdf-q[k].sdf|+\\ \sum_{l=len(q)+1}^{len(p)}\left|\begin{matrix}p[l].sdf-\\ q[len(q)].sdf\end{matrix}\right|\\ \times w_{l-len(q)}\end{pmatrix}}{len(q)} \tag{3}$$

$$w_t=1+t\times\alpha,\quad t=1,\dots,len(p)-len(q) \tag{4}$$

where len denotes the number of topological nodes in a topological string, and w denotes the penalizing weights. **Fig. 6** illustrates the comparison between two topological strings. In our experiments, $\alpha=0.2$ yields a good retrieval result.

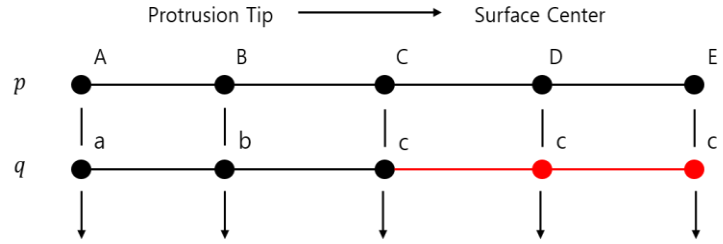

Fig. 6. Comparison between two topological strings (p, q) with SDF values, where $len(p)>len(q)$

Let M and N be two 3D models. Assume that they have m and n topological strings, respectively. After comparing each topological string in M with each in N, we get a $m\times n$ matrix with the dissimilarity values calculated using equation (3). To apply the Hungarian algorithm, the string dissimilarity matrix is required to be a square matrix. In the case of $m=n$, the Hungarian algorithm can be directly applied. If $m\neq n$, we pad the rows (or columns) of the string dissimilarity matrix with the mean of existing values of the columns (or rows). We also tested padding with max and min values; the result in section 4 shows that padding with mean values works best. Assuming that $m>n$, we can define an $m\times m$ string dissimilarity matrix as $Dism(i,j)$, where $1\leq i\leq m,\ 1\leq j\leq n$ have the dissimilarity values of topological strings. The padding procedure is mathematically formulated as follow:

$$Dism(i,j)=\sum_{u=1}^{n}Dism(i,u)/n, \tag{5}$$
$$1\leq i\leq m, n<j<m, m>n$$

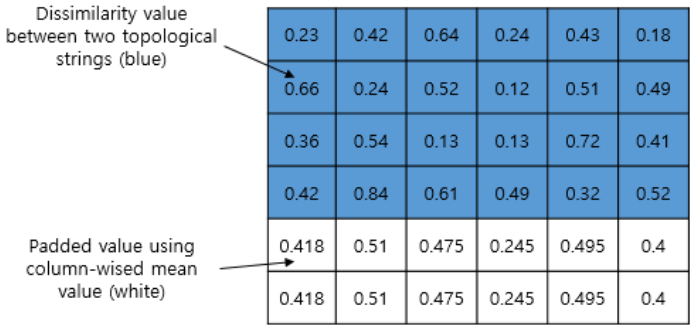

Fig. 7. Example of the string dissimilarity matrix when two models have 4 and 6 topological strings respectively.

For example, if there are two 3D models that have 4 and 6 topological strings, respectively, **Fig. 7** shows the corresponding string dissimilarity matrix.

Applying the Hungarian algorithm will return the "minimum matching" indexes. The final dissimilarity value between two models is the average of the indexed value of the dissimilarity matrix.

4. Experimental Results

In this section, we first test how the parameters affect the retrieval result of the proposed algorithm. Then, we compare our algorithm with other state-of-the-art methods by using the parameters that give the best retrieval performance. In the following experiments, if there is no notice of parameters: R_g and T_n is set as $\sqrt{0.005 * area(S)} * 3$ and $\sqrt{0.005 * area(S)} * 0.75$. Then, as the calculation of geodesic distances is computationally expensive, we first use QSlim [27] to simplify the mesh to 1500 faces. Retrieval accuracy is evaluated by following five quantitative measures and the precision-recall curve [22]:

- Nearest Neighbor (NN): The percentage of best matches that belong to the query's class.
- First Tier (FT) and Second Tier (ST): The percentage of models belonging to the query's class that appear within the top $(K-1)$ and $2(K-1)$ matches where the number of models in the query's class is *K*.
- E-measure: A composite measure of the precision and recall for a fixed number (32) of retrieved models.
- Discounted Cumulative Gain (DCG): A statistic that weights correct results near the front of the list more than correct results later in the ranked list.
- Precision-Recall curve (P-R curve): Precision is the ratio of retrieved models that are relevant to a given query, while recall is the ratio of relevant models to a given query that have been retrieved from the total number of relevant models. Thus, a higher P-R curve indicates better retrieval performance.

The five quantitative measures are in the range [0,1] and higher values indicate better retrieval performance. For more details about the metrics we refer readers to [22].

We carry out experiments on the datasets of SHREC' 15 Non-rigid 3D Shape Retrieval to test our parameters. The datasets of the SHREC' 15 Non-rigid 3D Shape Retrieval [26] contain 1200 non-rigid models that are classified into 50 classes, each with 24 models.

Table 1. Retrieval results from different $R_g = \sqrt{0.005 * area(S)} * \alpha$ on SHREC' 15 Non-rigid 3D Shape Retrieval

α \ Metrics	NN	FT	ST	E-measure	DCG
2	0.9325	0.6838	0.7901	0.6196	0.8913
3	0.9725	**0.8019**	**0.8878**	**0.7103**	**0.9462**
4	**0.9808**	0.7946	0.8811	0.7032	0.9455
5	0.9733	0.7757	0.8666	0.6909	0.9363
6	0.9708	0.7453	0.8470	0.6685	0.9268

Table 1 shows the retrieval results of our algorithm with various $R_g = \sqrt{0.005 * area(S)} * \alpha$. We can see that when $\alpha = 3$, all metrics yields best result except NN that is slightly lower than when $\alpha = 4, 5$. And when $\alpha = 2$, the results drop significantly, approximately 10% on each metric. When α is greater than 3, the results decrease as α increases.

Table 2 shows the retrieval results of our algorithm with various $T_n = \sqrt{0.005 * area(S)} * \beta$. When $\beta = 0.75$, all metrics yield the best results except NN that is slightly lower than when $\beta = 0.5, 1.25$. When β is greater or smaller than 0.75, the results decrease as β increases or decreases.

Table 2. Retrieval results with different $T_n = \sqrt{0.005 * area(S)} * \beta$ on SHREC' 15 Non-rigid 3D Shape Retrieval

β \ Metrics	NN	FT	ST	E-measure	DCG
0.25	0.9625	0.7709	0.8697	0.6904	0.9333
0.5	**0.9775**	0.7969	0.8874	0.7075	0.9448
0.75	0.9725	**0.8019**	**0.8878**	**0.7103**	**0.9462**
1	0.9725	0.7993	0.8872	0.7080	0.9450
1.25	0.9742	0.7924	0.8846	0.7057	0.9438

As described in Section 3.3, to apply the Hungarian algorithm, the string dissimilarity matrix must be a square matrix. When the topological string number of two models is not equal, we tested padding the matrix with min, mean and max value. From **Table 3**, we see that padding with the mean value yields the best retrieval result.

Table 3. Retrieval results by padding the string dissimilarity matrix with min, mean and max values for SHREC' 15 Non-rigid 3D Shape Retrieval

Padded \ Metrics	NN	FT	ST	E-measure	DCG
Min	0.9700	0.7752	0.8700	0.6918	0.9362
Mean	**0.9725**	**0.8019**	**0.8878**	**0.7103**	**0.9462**
Max	0.9683	0.7789	0.8741	0.6958	0.9376

The results in the **Table 4** show that the root-area SDF normalization method is more appropriate for our algorithm than the log-scale normalization method [3].

Table 4. Comparison between the log-scale SDF normalization method and the root-area SDF normalization method for SHREC' 15 Non-rigid 3D Shape Retrieval

Normalization \ Metrics	NN	FT	ST	E-measure	DCG
Log-space	0.9408	0.7118	0.8126	0.6425	0.9015
Root-area	**0.9725**	**0.8019**	**0.8878**	**0.7103**	**0.9462**

Because most of our algorithm is based on reference [11] (denoted as ConTopo++), we compare retrieval performance between two algorithms on the SHREC'11 Non-rigid 3D Watertight Meshes dataset [25]. In **Table 5**, Proposed_1500 and Proposed_6000 denote our algorithm performance when mesh faces are reduced to 1500 and 6000. From the table, we can see that higher resolution that requires a longer time to calculate integral geodesic, has better retrieval performance with our algorithm, and both perform better than ConTopo++ using five quantitative measures. There are two main reasons for better performance of our algorithm. First, as we mentioned in section 3.1, on some models that have significantly distorted places or unbalanced structure, the conformal factor is not as stable as geodesic distance. Second, when padding the string dissimilarity matrix, ConTopo++ used the difference between the numbers of strings in each model divided by their sum, while we padded with the mean of the already filled column-wised or row-wised string dissimilarity values.

Table 5. Comparison between our algorithm and ConTopo++ on SHREC'11 Non-rigid 3D Watertight Meshes

Methods \ Metrics	NN	FT	ST	E-measure	DCG
Proposed_1500	0.9933	0.9078	0.9746	0.7171	0.9823
Proposed_6000	**0.9983**	**0.9408**	**0.9913**	**0.7297**	**0.9902**
Contopo++	0.9930	0.8850	0.9520	0.6950	0.9810

In **Fig. 8** and **Fig. 10**, we illustrate the P-R curve of the proposed method against the published results of the SHREC'11 Non-rigid 3D Watertight Meshes and SHREC' 15 Non-rigid 3D Shape Retrieval, respectively. The corresponding five quantitative measures are illustrated in **Fig. 9** and **Fig. 11** respectively using column charts. In **Fig. 8** and **Fig. 9**, we can see that our Proposed_6000 and Proposed_1500 ranked second and third among the options for the SHREC'11 Non-rigid 3D Watertight Meshes. **Fig. 10** and **Fig. 11** illustrate that the retrieval performance of our methods ranked fourth and seventh among the options for SHREC' 15 Non-rigid 3D Shape Retrieval.

We implemented the proposed algorithm in Matlab on a personal computer with a 3.60 GHz i7-4790 CPU, 8GB DDR3 memory. There are only two contestants on SHREC' 15 Non-rigid 3D Shape Retrieval wrote about their running time, Giachetti's HAPT algorithm needs 3 min on average to extract a feature map of the tested dataset, and Limberger's algorithm requires 18 seconds to compute three local descriptors on a model. In our method, most of this time is spent on calculating the integral geodesic. When we reduce the mesh to 1500 faces and 6000 faces, it takes approximately 2 seconds and 25 seconds, respectively, to calculate integral geodesic by using parallel computation with 4 cores. For mesh matching that uses the Hungarian algorithm, it takes approximately 2 milliseconds for comparing between two meshes.

Time complexity: The time complexity of the proposed method can be broken down into three parts: (1) For construscting topological structure, integral geodesic computation requires $O(v \log v)$ where the v is number of the vertex in the mesh, and extracting protrusion tips cost $O(v)$. (2) Computation of SDF value cost $O(TN_f \times v \times R_{sdf})$ where TN_f is number of faces near topological nodes, TN_f is usually around 10, and R_{sdf} is number of rays used to compute SDF, R_{sdf} is set as 30 in our experiments. (3) The computation of similarty between two 3D

models by using the Hungarian Algorithm require $O(MTS^3)$ where MTS is max topological string number between two 3D models, MTS is less than 10 in most cases. Thus, the overall time complexity is $O(v \log v) + O(TN_f \times v \times R_{sdf}) + O(MTS^3)$.

Data storage space: Each topological node is assigned with a double-precision SDF value which occupy 8 bytes, and the average number of topological nodes in a 3D models on SHREC' 15 Non-rigid 3D Shape Retrieval (1200 3D models) with threshold $R_g = \sqrt{0.005 * area(S)} * 3$ and $T_n = \sqrt{0.005 * area(S)} * 0.75$ is 64, so our topological structure and SDF value based descriptor of a 3D model only need $64 \times 8 = 512$ bytes data storage space.

5. Conclusion

In this paper, we developed an efficient and effective method for the retrieval of non-rigid 3D models mainly based on geodesic distance and SDF that are two pose-invariant features on the mesh surface. The experiment on the SHREC'11 Non-rigid 3D Watertight Meshes and SHREC' 15 Non-rigid 3D Shape Retrieval shows that our method has higher retrieval accuracy than many of the state-of-the-art non-rigid 3D model descriptors. Furthermore, our method has the advantage of short running time and small data storage space required for descriptors.

In the future work, considering about the recent deep learning boom, we plan to extract topology and features from 3D models by applying some deep learning method to further improve the performance of the non-rigid 3D model retrieval.

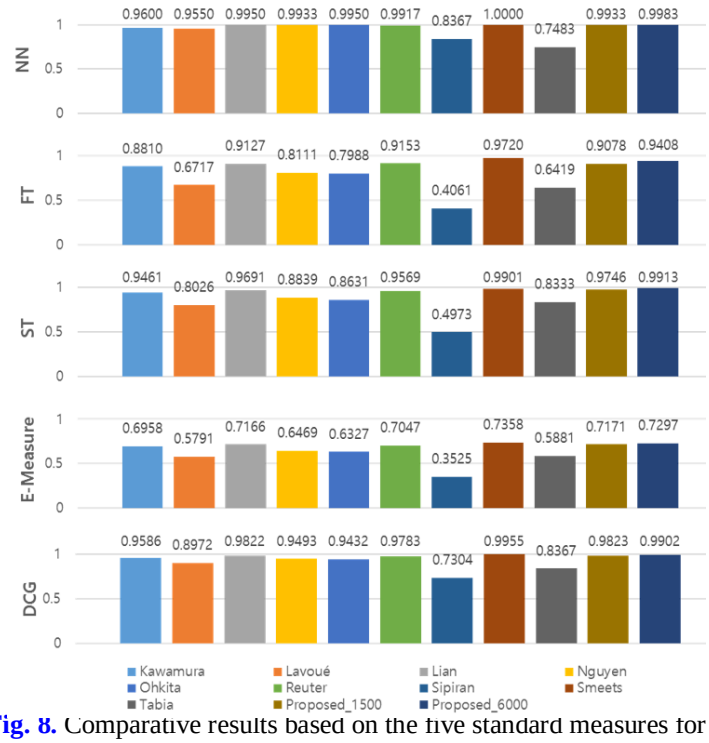

Fig. 8. Comparative results based on the five standard measures for the SHREC'11 Non-rigid 3D Watertight Meshes

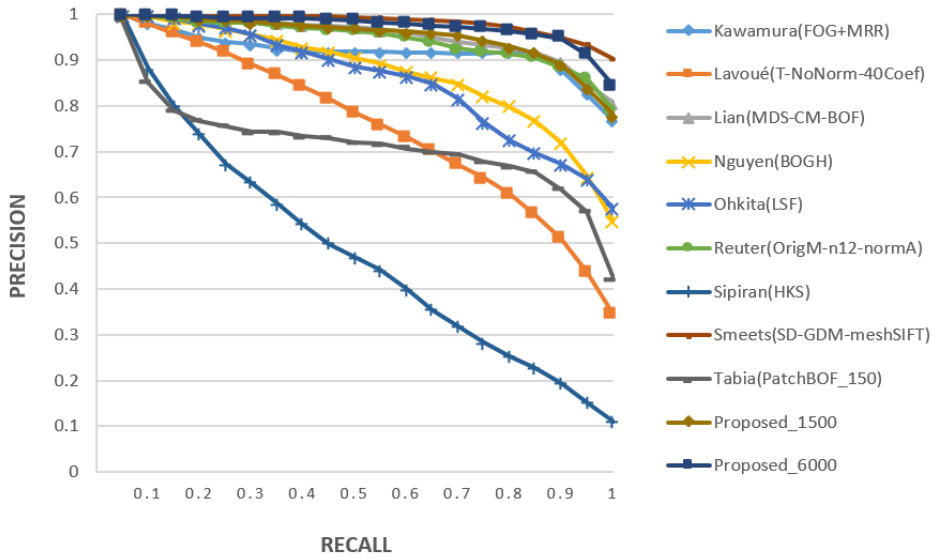

Fig. 9. Comparative results based on the P-R curves for the SHREC'11 Non-rigid 3D Watertight Meshes

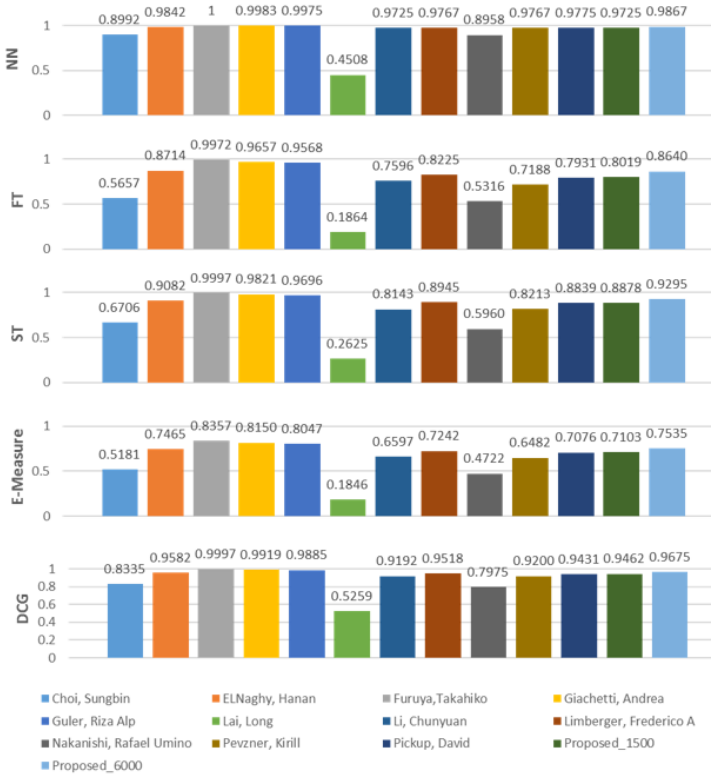

Fig. 10. Comparative results based on the five standard measures for the SHREC'15 Non-rigid 3D Shape Retrieval

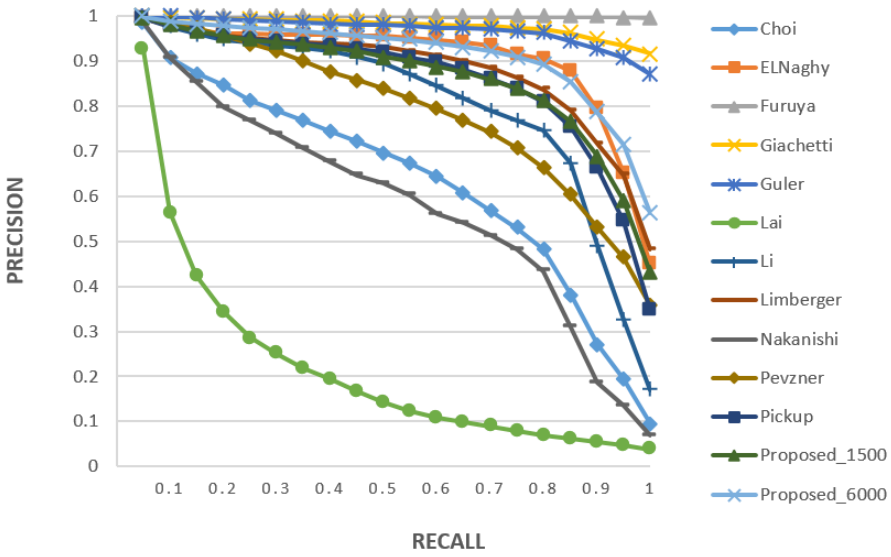

Fig. 11. Comparative results based on the P-R curves for the SHREC'15 Non-rigid 3D Shape Retrieval

Acknowledgements

This research was supported by Institute for Information & communications Technology Promotion (IITP) funded by the Ministry of Science, ICT and Future Planning (No. R0126-15-1024)

References

[1] A. Harris, "The Effects of In-home 3D Printing on Product Liability Law," *Journal of Science Policy and Governance*, vol. 6, issue. 1, February, 2015.

[2] D. Gupta, M. Tarlock, "3D Printing, Copyright Challenges, and the DMCA," *New Matter*, vol. 38, no. 3, 2013.

[3] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent mesh partitioning and skeletonisation using the shape diameter function," *The Visual Computer*, vol. 24, pp. 249-259, 2008. Article (CrossRef Link)

[4] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," in *Proc. of ACM SIGGRAPH '01*, pp. 203-212, 2001. Article (CrossRef Link)

[5] D. Chen, X. Tian, Y. Shen, M. Ouhyoung, "On visual similarity based 3D model retrieval," *Eurographics*, vol. 22, no. 3, pp. 223-232, 2003. Article (CrossRef Link)

[6] G. Passalis, T. Theoharis, I. Kakadiaris, "PTK: A Novel Depth Buffer-Based Shape Descriptor for Three-Dimensional Object Retrieval," *The Visual Computer*, vol. 23, pp. 5-14, 2007. Article (CrossRef Link)

[7] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. of SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 156-164. 2003.

[8] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, "Shape distributions," *ACM Transactions on Graphics*, vol. 21, pp. 807-832, 2002. Article (CrossRef Link)

[9] Z. Lian, A. Godil, X. Sun, "Visual similarity based 3D shape retrieval using bag-of-features," in *Proc. of SMI '10: Proceedings of the IEEE International Conference on Shape Modeling and Applications*, pp. 25-36, 2010. Article (CrossRef Link)
[10] Z. Lian, A. Godil, X. Sun, H. Zhang, "Non-rigid 3D shape retrieval using multidimensional scaling and bag-of-features," in *Proc. of International Conference on Image Processing*, pp. 3181-3184, 2010. Article (CrossRef Link)
[11] K. Sfikas, T. Theoharis, I. Pratikakis, "Non-rigid 3D object retrieval using topological information guided by conformal factors," *The visual Computer*, vol. 28, pp. 943-955, 2012. Article (CrossRef Link)
[12] M. Ben-Chen, C. Gotsman, "Characterizing shape using conformal factors," in *Proc. of 3DOR '08: Proceedings of the 1st Eurographics conference on 3D Object Retrieval*, pp. 1-8, 2008. Article (CrossRef Link)
[13] M. Reuter, F. Wolter, N. Peinecke, "Laplace-Beltrami spectra as 'Shape-DNA' of surfaces and solids," *Computer-Aided Design*, vol. 38, pp. 342–366, 2006. Article (CrossRef Link)
[14] J. Sun, M. Ovsjanikov, L.J. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, pp. 1383-1392, 2009. Article (CrossRef Link)
[15] M. Aubry, U. Schlickewei, D. Cremers, "The wave kernel signature: a quantum mechanical approach to shape analysis," in *Proc. of computational methods for the innovative design of electrical devices*, pp. 1626-1633, 2011. Article (CrossRef Link)
[16] C. Li, A. Ben Hamza, "Spatially aggregating spectral descriptors for nonrigid 3D shape retrieval: a comparative survey," *Multimedia Systems*, vol. 20, pp. 253-281, 2014. Article (CrossRef Link)
[17] R. Gal, A. Shamir, D. Cohen-Or, "Pose-oblivious shape signature," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 261-271, 2007. Article (CrossRef Link)
[18] A. Agathos, I. Pratikakis, P. Papadakis, S.j. Perantonis, P.N. Azariadis, N.S. Sapidis, "3D articulated object retrieval using a graph-based representation," *The Visual Computer*, vol. 26, pp. 1301-1319, 2010. Article (CrossRef Link)
[19] R. Kimmel, J.A. Sethian, "Computing geodesic paths on manifolds," in *Proc. of the National Academy of Sciences of the United States of America*, vol. 95, pp. 8431-8435, 1998. Article (CrossRef Link)
[20] G. Peyr´e, L.D. Cohen, "Geodesic remeshing using front propagation," *International Journal of Computer Vision*, vol. 69, pp. 145-156, 2006. Article (CrossRef Link)
[21] H.W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, pp. 83-97, 1955. Article (CrossRef Link)
[22] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, "The Princeton Shape Benchmark," in *Proc. of SMI '04: Proceedings of the Shape Modeling International 2004*, pp. 167-178, 2004. Article (CrossRef Link)
[23] J. Tangelder, R. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools and Applications*, vol. 39, pp. 441-471, 2008.Article (CrossRef Link)
[24] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H.V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, D. Vandermeulen, "A comparison of methods for non-rigid 3D shape retrieval," *Pattern Recognition*, vol. 46, pp. 449-461, 2013.Article (CrossRef Link)
[25] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoue, H. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, D. Vandermeulen, "SHREC'11 track: shape retrieval on non-rigid 3D watertight meshes," in *Proc. of 3DOR '11: Proceedings of the 4th Eurographics conference on 3D Object Retrieval*, pp. 79-88. 2011. Article (CrossRef Link)
[26] Z. Lian, J. Zhang, S. Choi, H. ElNaghy, J. El-Sana, T. Furuya, A. Giachetti, R.A. Guler, L. Lai, C. Li, H. Li, F.A. Limberger, R. Martin, R.U. Nakanishi, A.P. Neto, L.G. Nonato, R. Ohbuchi, K. Pevzner, D. Pickup, P. Rosin, A. Sharf, L. Sun, X. Sun, S. Tari, G. Unal, R.C. Wilson, " SHREC'15 Track: Non-rigid 3D Shape Retrieval," in *Proc. of 3DOR Proceedings of the 2015 Eurographics Workshop on 3D Object Retrieval*, pp. 107-120, 2015. Article (CrossRef Link)

[27] M. Garland, Qslim Simplification Software, Available from: http://www.cs.cmu.edu/~./garland/quadrics/qslim.html [retrieved: July, 2016].
[28] M. Garland, P.S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. of SIGGRAPH 1997: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 209-216, 1997. Article (CrossRef Link)
[29] Conformal factor computation, Available from: http://en.pudn.com/downloads637/sourcecode/graph/detail2582972_en.html [retrieved: May, 2016]
[30] Y. Hong, J. Kim, "Non-Rigid 3D Model Retrieval Based on Topological Structure and Shape Diameter Function," in *Proc. of ADVCOMP 2016: The Tenth International Conference on Advanced Engineering Computing and Application in Sciences*, pp. 63-67, 2016.
[31] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, P. Suetens, "Isometric Deformation Modelling for Object Recognition," in *Proc. of* The 13th International Conference on Computer Analysis of Images and Patterns (CAIP'09), pp. 757–765, 2009. Article (CrossRef Link)

Yiyu Hong received his B.S. degree in Computer Science and Technology from Beijing University of Technology, China, in 2014. He is currently pursuing the Ph.D. degree in Copyright Protection, Sangmyung University, Korea. His research interests are digital watermarking, 3D model segmentation, 3D model retrieval, 3D model identification.

JongWeon Kim received the Ph.D. degree from University of Seoul, major in signal processing in 1995. He is currently a professor of Dept. of Electronics Engineering and Director of Creative Content Labs at Sangmyung University in Korea. He has a lot of practical experiences in the digital signal processing and copyright protection technology in the institutional, the industrial, and academic environments. His research interests are in the areas of copyright protection technology, digital rights management, digital watermarking, and digital forensic marking.