

3D Model Comparison through Kernel Density Matching

Yiming Wang^{1,3}, Tong Lu^{*1,3}, Rongjun Gao^{1,3}, Wenyin Liu^{1,2}

¹State Key Lab. for Novel Software Technology, Nanjing University, Nanjing, 210093, China

²Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China

³JiangYin High Technology and Information Institute, Nanjing University, China

{freewym@gmail.com, lutong@nju.edu.cn, csluwy@cityu.edu.hk}

Abstract

A novel 3D shape matching method is proposed in this paper. We first extract angular and distance feature pairs from pre-processed 3D models, then estimate their kernel densities after quantifying the feature pairs into a fixed number of bins. During 3D matching, we adopt the KL-divergence as a distance of 3D comparison. Experimental results show that our method is effective to match similar 3D shapes, and robust to model deformations or rotation transformations.

1. Introduction

3D systems have been widely used in entertainment and engineering industries in recent years, creating a strong commercial demand for automatic 3D retrieval systems to reuse previous designs.

Most existing methods yield matching results by simplifying a 3D mesh model to either topological or statistic features. Topology-based 3D matching methods first convert models to different types of topological graphs, such as skeletons [1], multi-resolution Reeb-graph [2] and attribute-graph [3], to describe the topological constraints of model faces and edges. Graph-matching algorithms are then used to search for the maximum common subgraphs to calculate the similarity of two 3D models. Unfortunately, topological structures are not always reliable for 3D matching since two visually similar models may have different internal representations. Moreover, topological graphs may be too complex for exact or inexact matching algorithms. Statistics-based methods convert 3D models to statistical shape descriptors such as 3D moments [4-5], distance distribution histograms [6-7], and function projections [8-9]. Although moments are invariant to model rotation or scaling, they are unable to provide detailed descriptions about 3D shapes. Distribution histogram such as D2 [6] is also an invariant descriptor, but

limited in that 3D shapes cannot always be well distinguished only by distance constraints. Therefore, different groups of 3D models may share similar or even the same distribution histograms [10]. Function projection is effective to describe 3D shapes; however, it still faces difficulties to match more complex models. For statistics-based methods, one of the difficulties lies in the fact that they cannot yield good results since topological or geometrical constraints may be lost during the feature extraction process.

We propose a novel shape matching method to calculate similarity of two 3D shapes. It consists of two components: feature representation and 3D matching. During feature representation, a model is gradually converted to a continuous joint density in the following four steps: 1) *model normalization*: normalize the models into a uniform size centered at the origin of the coordinate system by translation and scale transformations; 2) *feature-pair extraction*: extract angle and distance feature pairs over model surfaces; 3) *feature-pair quantization*: quantize the extracted feature pairs into a fixed number of bins; and 4) *joint density fitting*: fit the feature pairs from different bins into a continuous kernel density. Finally, in 3D matching step, we calculate the similarity of two 3D models by adopting the Kullback-Leibler divergence (KL-divergence) as a distance of their estimated distribution densities.

Our method offers the following novelties over other 3D shape matching methods. First, the continuous estimation makes this method not sensitive to model deformations and model defects. Second, as a kind of non-parametric method, our method has no prior assumptions on model shapes. Third, comparing with other statistical algorithms, our method considers both angular and distance constraints over a 3D shape through a joint distribution estimation, making the similarity calculation more accurate and robust.

2. Feature extraction and representation

2.1. Model normalization

During this step, we adopt the following operators to pre-process and normalize each 3D model: mesh subdivision, model translation and model scaling. The purpose is to make our method robust to multiple model resolutions and insensitive to translation or scaling.

Mesh subdivision checks triangular meshes in a 3D model and subdivides large meshes into smaller ones to obtain uniform grids. As we can see, the meshes on the horse body in Fig. 1(a) are much larger than other parts, potentially making 3D matching unreliable. In our method, model meshes are subdivided as follows:

1. Set a subdivision threshold th_{area} for each 3D model (th_{area} can be the arithmetic mean area or median area of all the model meshes);
2. Traverse each mesh, and subdivide it if its area is larger than th_{area} : add a new edge from the midpoint of the longest edge (see $edge_l$ in Fig. 2) to its opposite vertex (see $vertex_A$ in Fig. 2) and subdivide the triangular mesh into two new ones;
3. Repeat 2 until each mesh area is less than or equal to th_{area} .

Fig. 1(b) shows the result after mesh subdivision.

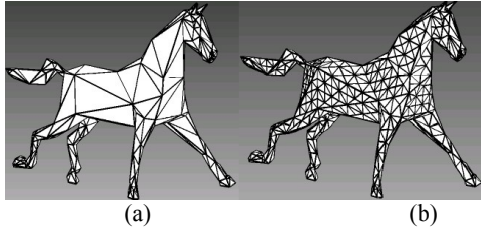


Fig. 1 A mesh subdivision example. (a) an original horse model with larger meshes on its body; (b) mesh subdivision result.

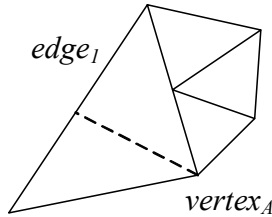


Fig. 2 Mesh subdivision.

Model translation and scaling are adopted to make our method not sensitive to translation or scaling. For this purpose, we first translate the centroid of the model to the origin of its model coordinate system and then calculate the scaling factor by normalizing the variance of the distances from the mesh vertices to the centroid:

$$factor_{scaling} = 1 / \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} \|v_i p_{centroid}\|^2} \quad (1)$$

where v_i is a mesh vertex, n is vertex number of a 3D model, and $p_{centroid}$ is its centroid.

As a result, each model is normalized into a unit coordinate system with uniformly distributed mesh triangles after this step.

2.2. Feature-pair extraction

We propose a new descriptor taking advantage of both statistical and structural information from 3D models. We notice that the geometrical constraints between every pair of mesh triangles play a key role to describe a 3D shape. In another word, given the geometrical constraints of any mesh triangular to all others, a 3D model can be uniquely determined.

Suppose a normalized 3D model $P = \{t_0, t_1, \dots, t_{n-1}\}$, where t_i is a triangular mesh on P and $t_i = \{v_{i1}, v_{i2}, v_{i3}\}$. It can be concluded that different angular and distance distributions of t_0, t_1, \dots, t_{n-1} result in different 3D shapes. Without losing generality, we arbitrarily select two triangular meshes, t_i and t_j from $\{t_0, t_1, \dots, t_{n-1}\}$, as an example to explain the constraints. As shown in Fig. 3, \vec{n}_i and \vec{n}_j are the normal vector of t_i and t_j , respectively. The constraint between t_i and t_j is defined as $\langle \theta_{i,j}, L_{i,j} \rangle$, where

$$\theta_{i,j} = \frac{\vec{n}_i \cdot \vec{n}_j}{\|\vec{n}_i\| \cdot \|\vec{n}_j\|} \quad (2)$$

$$L_{i,j} = \min \sum_{k=0}^{m-1} l_{x_k, x_{k+1}} \quad (3)$$

where $x_0 = i, x_m = j, t_{x_{k+1}} = next(t_{x_k})$, $(t_{x_0}, t_{x_1}, \dots, t_{x_{m-1}}, t_{x_m})$ is any path from t_i to t_j over the surface of P , and

$$l_{x,y} = \begin{cases} d_x + d_y, & (v_{x1}, v_{x2} \in t_y) \vee (v_{x2}, v_{x3} \in t_y) \vee (v_{x3}, v_{x1} \in t_y) \\ \infty, & else \end{cases} \quad (4)$$

where d_x is the distance from the centroid of t_x to the shared edge with t_y (see d_i and d_{kl} when calculating $l_{i,k1}$ in Fig. 3). We use the Floyd algorithm [11] to minimize (3).

The above pair $\langle \theta_{i,j}, L_{i,j} \rangle$ between any two meshes well describes the structural constraints of a 3D shape. However, for a model P with n triangles, there are altogether $n(n-1)/2$ constraints pairs as follows, with a very large computational cost:

$$S_{constraint} = \{ \langle \theta_{i,j}, L_{i,j} \rangle \mid i \in [0, n-1], j \in [i+1, n] \} \quad (5)$$

Therefore, we need to resample $S_{constraint}$ to reduce the computational cost before estimating the density from the constraints.

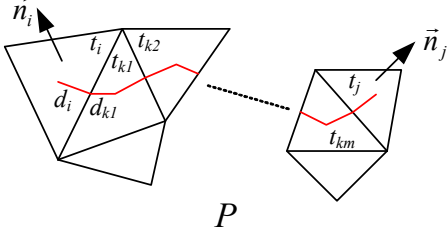


Fig. 3 Feature pair extraction from a 3D model.

2.3. Feature-pair quantization

Given the model $P = \{t_0, t_1, \dots, t_{n-1}\}$, feature-pair quantization maps its constraints set $S_{constraint}$ to n bins to reduce the complexity from $O(n^2)$ to $O(n)$. Feature-pair quantization has the following steps:

1. Sort $S_{constraint}$ in ascending order of $\theta_{i,j}$;
2. Divide $S_{constraint}$ into $\lfloor \sqrt{n} \rfloor + 1$ bins on the condition that each bin has the same number of feature pairs;
3. Traverse the divided bin group. For each bin bin_k ($k=1, 2, \dots, \lfloor \sqrt{n} \rfloor + 1$):
 - a) sort the feature pairs in bin_k in ascending order of $L_{i,j}$;
 - b) divide bin_k into $\lfloor \sqrt{n} \rfloor + 1$ sub-bins on the condition that each sub-bin has the same number of feature pairs;
 - c) replace the bin group in Step 2 with the sub-bins obtained in this step;
4. Traverse the new bin group and represent bin_k ($k=1..n$) with the following feature pair $\langle \hat{\theta}_k, \hat{L}_k \rangle$, where

$$\hat{\theta}_k = \sum_{\langle \theta_{i,j}, L_{i,j} \rangle \in bin_k} \theta_{i,j} / |bin_k|, \hat{L}_k = \sum_{\langle \theta_{i,j}, L_{i,j} \rangle \in bin_k} L_{i,j} / |bin_k| \quad (6)$$

Feature-pair quantization reduces the number of feature pairs to $O(n)$, reducing the potential computational cost in density fitting or 3D matching. Our next target is to fitting the feature pairs with joint densities.

2.4. Joint density estimation

Finally, we use kernel density to estimate the distributions in the feature pair space. Kernel density has the following advantages: 1) it needs no explicit model assumptions and directly uses the observed data to determine densities, and 2) it can estimate a continuous probability density. Considering the feature pairs we obtained, the kernel density estimated from bivariate samples $\mathbf{X}_1, \dots, \mathbf{X}_n$ can be defined as:

$$\hat{g}(\mathbf{x}; \mathbf{H}) = n^{-1} \sum_{i=1}^n K_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i) \quad (7)$$

where $\mathbf{x} = (x_1, x_2)^T$, $\mathbf{X}_i = (X_{i2}, X_{i2})$, $K_{\mathbf{H}}(\mathbf{x})$ is the bivariate kernel (often adopts a standard Gaussian

function) and \mathbf{H} is the bandwidth matrix. \mathbf{H} is an important factor in determining the performance of \hat{g} , and mean integrated squared error (MISE) [12] can be adopted as the criterion to evaluate \mathbf{H} . In our experiment, we adopt the method in [13] for bandwidth selection.

For simplicity, we assume the orientation of the kernel is along the coordinate axes and the bandwidth matrix is diagonal. Therefore, (7) can be rewritten as:

$$\hat{g}(x, y) = n^{-1} \sum_{i=1}^n K\left(\frac{x - x_i}{h_x}\right) K\left(\frac{y - y_i}{h_y}\right) \quad (8)$$

where (x_i, y_i) represents a bivariate sample, h_x and h_y are the elements on the diagonal of \mathbf{H} . The fast Gauss transform [16] can be adopted to further improve the efficiency in calculating the sum of Gaussian functions in (8).

3. KL divergence calculation

KL divergence is a non-symmetric measure of the difference between two probability distributions. Let $p^{(i)}(\mathbf{x})$ be the density of the i th retrieved model and $q(\mathbf{x})$ the density of the query model, the KL divergence from q to $p^{(i)}$ is as follows:

$$KL(q; p^{(i)}) = \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p^{(i)}(\mathbf{x})} d\mathbf{x} \\ = -E_q \{ \log p^{(i)}(\mathbf{x}) \} + \int q(\mathbf{x}) \log q(\mathbf{x}) d\mathbf{x} \quad (9)$$

Our target is to find i that minimizes (9). Since $\int q(\mathbf{x}) \log q(\mathbf{x}) d\mathbf{x}$ does not depend on the retrieved models, the problem can be formulated as follows:

$$f(X) = \arg \max_i \frac{1}{m} \sum_{j=1}^m \log p^{(i)}(t_j) \quad (10)$$

where X is the query with feature pairs t_1, t_2, \dots, t_m .

4. Experiments

In the first experiment, we test our algorithm on 10 selected group models from the Princeton database [14]. Fig. 4 gives the retrieval results comparing with some of the known 3D comparison algorithms (SFHM is an improved method of [15]).

The next experiment is to compare a group of 3D models with their transformed ones (with translation, scaling, rotation, and their combinations). Table 1 shows some comparison results, where KL divergences are listed below the transformed 3D models. It can be seen that affine transformations only have tiny affects in our method.

Finally we match 3D models with 10%, 20% and 30% randomly reduced meshes. Table 2 also shows a

tiny reduced KL divergence value, indicating the robustness of our method.

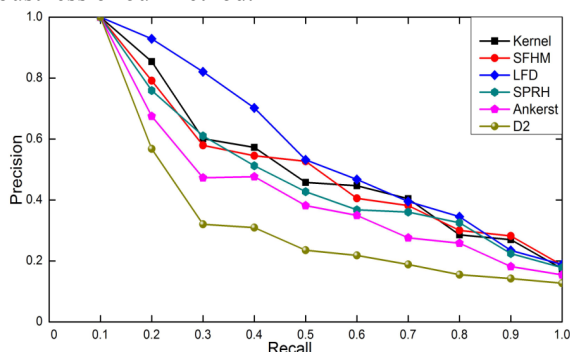


Fig. 4 Comparison of retrieval results.

Table. 1 KL divergence between models with affine transformations.








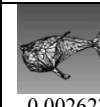
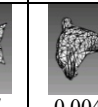
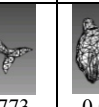


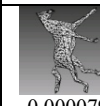

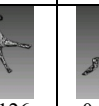

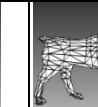
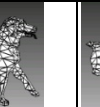


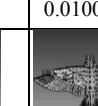



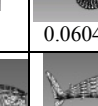
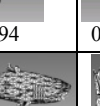
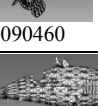
	Trans.	Scaling	Rotation	Combined
	 0.000037	 0.000039	 0.000062	 0.000312
	 0.004921	 0.002627	 0.004773	 0.004833
	 0.000142	 0.000079	 0.000126	 0.000011

Table. 2 KL divergence for defected models.

	Reduce 10%	Reduce 20%	Reduce 30%
	 0.010066	 0.020901	 0.033383
	 0.060494	 0.090460	 0.185409
	 0.010610	 0.046834	 0.090779

5. Conclusions

This paper proposes a novel 3D shape matching method based on kernel density estimation and KL-divergence calculation. Experimental results show that our method is effective to match 3D shapes and robust to model deformations. Further research includes involving 3D details comparison and content-based matching.

Acknowledgement

The paper was supported by the Natural Science Foundation of China under Grant No. 60723003 and 60703080, the 973 Program of China under Grant No. 2010CB327903, and the Natural Science Foundation of JiangSu under Grant No. BK2009082 and BK2008263.

References

- [1] H. Sundar, D. Silver, N. Gagvani, S. Dickinson. Skeleton based shape matching and retrieval. In: *Proceedings of Shape Modeling and Applications*, 130-139, 2003.
- [2] M. Hilaga, Y. Shinagawa, et. al. Topology matching for fully automatic similarity estimation of 3D shapes. *ACM SIGGRAPH*, 203-212, 2001.
- [3] M.El-Mehalawi, R.A. Miller. A database system of mechanical components based on geometric and topologic similarity, Part I: representation. *Computer-Aided Design*, 35(1):83-94, 2003.
- [4] F.A. Sadjadi, E.L. Hall. Three-dimensional moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(2):127-136, 1980.
- [5] M. Novotni, R. Klein. 3D Zernike descriptors for content based shape retrieval. In: *Proceedings of ACM Symposium on Solid Modeling and Applications*, Washington, USA, 216-225, 2003.
- [6] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807-832, 2002.
- [7] C.Y. Ip, D. Lapadat, L. Sieger, W.C. Regli. Using shape distributions to compare solid models. In: *Proceedings of ACM Symposium on Solid Modeling and Applications*, 273-280, 2002.
- [8] D. Saupe, D. Vranic. 3D model retrieval with spherical harmonics and moments. In: *Proceedings of Germany DAGM Conference*, Munich, Germany, 392-397, 2001.
- [9] S. Hou, K. Ramani. Classifier combination for sketch-based 3D part retrieval. *Computers & Graphics*, 31(4):1-12, 2007.
- [10] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin. Matching 3D models with shape distributions. In: *Proceedings of Shape Modeling and Applications*, 2001:154-166.
- [11] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6), 345, 1962.
- [12] T Duong, M Hazelton. Plug-in bandwidth matrices for bivariate kernel density estimation. *Journal of Nonparametric Statistics*, 15(1): 17-30, 2003.
- [13] <http://www.mathworks.com/matlabcentral/fileexchange/17204-kernel-density-estimation>.
- [14] P. Shilane, P. Min, et. al. The Princeton shape benchmark. In: *Proceedings of the Shape Modeling International*, 2004, 167-178.
- [15] C. Xu, J. Liu, X. Tang. 2D Shape Matching by Contour Flexibility. *IEEE TPAMI*, 31(1):180-186, Jan. 2009.
- [16] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. In: *proceedings of the International Conference of Computer Vision*, 1: 464, 2003.