A Shape Distribution for Comparing 3D Models

Levi C. Monteverde[1], Conrado R. Ruiz Jr.[2], and Zhiyong Huang[3,4]

[1] Citibank, International Technology Organization, 1 Temasek Avenue #26-00 Millenia Tower Singapore 039192, Singapore
Levijones.ait.monteverde@citigroup.com
[2] De La Salle University, College of Computer Studies, Taft. Avenue, Manila, Philippines
Ruizc@dlsu.edu.ph
[3] School of Computing, National University of Singapore
[4] Institute for Infocomm Research (I^2R), Singapore
Huangzy@comp.nus.edu.sg

Abstract. This study developed a new shape-based 3D model descriptor based on the D2 shape descriptor developed by Osada, et al of Princeton University. Shape descriptors can be used to measure dissimilarity between two 3D models. In this work, we advance it by proposing a novel descriptor D2a. In our method, N pairs of faces are randomly chosen from a 3D model, with probability proportional to the area of the face. The ratio of the smaller area over the larger area is computed and its frequency stored, generating a frequency distribution of N ratios which is stored as the second dimension of a 2D array, while the first dimension contains the frequency distribution of distances of randomly generated point pairs (the D2 distribution). The resulting descriptor, D2a, is a two-dimensional histogram that incorporates two shape features: the ratio of face areas and the distance between two random points.

1 Introduction

An important research area is the efficient storage and retrieval (shape-matching) of desired 3D models from an unorganized database of models (e.g. the World Wide Web). For example, if one needs a 3D model of an airplane, searching a database using "airplane" as keyword may not yield satisfactory results, since the filenames may not be descriptive (e.g. 001.wrl), may be in a foreign language, or might be misspelled. A better way is to combine keyword searching with an actual 3D model as query.

The shape dissimilarity between two 3D models is measured by applying a distance measure (such as Euclidian distance) to the shape descriptors of the two models being compared. The top k closest matches of the query model are those models with the smallest dissimilarity value compared to the query.

A shape descriptor that is used to compare 3D models based on shape similarity must be both accurate and efficient. In 2001, Osada [1] et al proposed the use of "shape distributions" as shape descriptors. These are frequency distributions

T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4351, Part I, pp. 54 – 63, 2007.

of certain shape features like angles, areas and distances randomly sampled from a 3D model. The most effective distribution was D2, which was the frequency distribution of distances of randomly selected point pairs on the surface of a 3D model. D2 was represented as a 1D array of integers, a histogram of frequencies.

D2 has several advantages. It is both rotation and translation invariant, is computationally cheap (both for generating the descriptor and comparing two descriptors), and describes the overall shape of an object, which means that it is not easily affected by minor shape distortions. For example, when a 3D model of a car is compared to a version of itself that has 5% less polygons, the overall shapes (and D2 values) of the two models remain very close [1].

In 2003, Ohbuchi [2], et al, extended Osada's D2 and added a second dimension (to the D2's one-dimensional histogram). The second dimension records the frequencies of the angles between the normal vectors of the two surfaces containing the two random points of a D2 sample. Instead of a 64-bin histogram, they used a 64x8 histogram for the AD (for "angle") descriptor, and 64x4 histogram for the AAD (for "absolute angle") descriptor. The AD and AAD enhanced descriptors outperformed D2 by 19% and 28% respectively. They used a database of 215 VRML models and implemented D2 in order to compare results with AD and AAD.

Prior to the Princeton Shape Benchmark (PSB) of Shilane, et al [3], researchers had to assemble their own test and training databases of 3D models from various sources. In order to compare the results of a shape matching algorithm, it was necessary to implement all shape descriptors that were to be compared. In 2003, the PSB was made available publicly for researchers in 3D model classification and retrieval. The PSB consists of a database of 1,814 3D models in the Object File Format (.OFF) and utility programs to measure the performance of any shape-matching algorithm that uses the PSB database. This allows researchers to directly compare a shape descriptor's 3D shape-matching performance with the results other studies that also used the PSB.

In 2004, Shilane [3] used the PSB to compare the performance of twelve shape descriptors in measuring shape dissimilarity, including Osada's D2. This study extends the D2 shape descriptor by extracting another shape feature – the ratio of areas between randomly chosen faces – and combining this with the original D2 histogram. A more detailed discussion of this method is presented in chapter 2. The PSB and the results of Shilane are used to gauge the relative effectiveness of the D2a shape descriptor in 3D shape matching.

2 Overview

In this section, we provide the theoretical framework behind the D2a shape descriptor, as well as a detailed description of D2a.

2.1 General Approach to the Shape Comparison of 3D Models

The most successful and popular approach so far in comparing the shape of two 3D models has been to apply two steps [4]:

Step 1. Apply some function on the shape feature(s) of a given 3D model to extract a "shape descriptor" for the 3D model. Shape features include areas, distances, angles, 2D projections, etc.

Step 2. Apply a distance formula to compare the shape descriptors of 2 models. Examples of distance formulas are the Manhattan, Euclidian, and Earth Mover's distance formulas. Figure 1 demonstrates this two-step process.

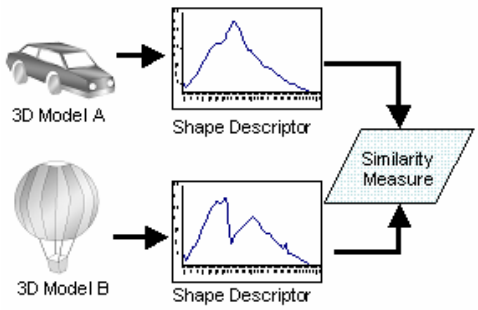

Fig. 1. The general approach to shape-based comparison of 3D models

2.2 Invariance to Transformation

The challenge in developing an ideal shape descriptor is twofold:

1. To develop the best shape descriptor that can represent 3D models, and
2. To remain unaffected by all transformations (scale, rotation and translation).

There are two ways to address these challenges:

1. Develop a *transformation-invariant descriptor* so that all rotations, scaling and translations of a model result in the same descriptor.
2. *Normalization*. 3D models can be normalized by finding a suitable transformation for each one. Unfortunately for this approach, there is no robust way to normalize rotation transformations [5], unlike with scale or translation. It is also possible to normalize the shape descriptor itself instead of the 3D model.

Figure 2 illustrates how 3D Shape Comparison is made when the descriptors are transformation invariant. In cases where the descriptor is not transformation invariant, normalization is applied to either the 3D model before the descriptor is extracted or on the descriptor itself.

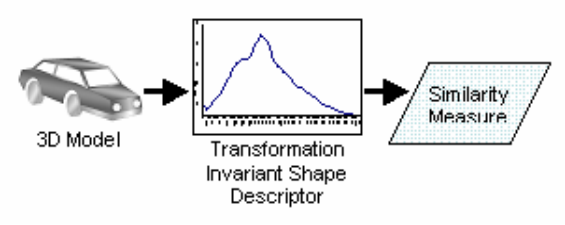

Fig. 2. Extracting a rotation, scale and translation invariant shape descriptor from a 3D model

2.3 The D2 Shape Descriptor

A 3D model is made up of a finite number of vertices and faces. Theoretically, on those faces lie an infinite number of points.

The distances between all pairs of points on the surface of the 3D model have a probability distribution. **This probability distribution is D2**. D2 is also called a shape distribution, because it is based on a feature of the model's shape, i.e. distances between all pairs of points. Osada noted that the D2 shape distribution is distinctive for each 3D model [1], and therefore represents the model's overall shape, i.e. it can be used as a shape descriptor.

However, since it is impossible to find the probability distribution of an infinite set (i.e. the set of all points on a 3D model), the actual implementation of D2 approximates the distribution by *randomly sampling* a sufficient number of points and recording the frequency of each range of distances. For example, the D2 distribution can be approximated by 1024 samples points, resulting in 1024*1024/2 + 1024 = 524,800 sample point-pair distances, since $|P_i P_j|$ is the same as $|P_j P_i|$, and is counted only once.

D2 is invariant to translation and rotation. Intuitively, no matter how the 3D model is rotated or translated, all of its vertices, faces and surface points move along with it, resulting in the same point-pair distances. However, it requires normalization for scale transformations. There several ways to normalize a D2 distribution. The two simplest and most effective are aligning by mean and aligning by maximum distance [2].

2.4 The D2a Shape Descriptor

The intuition behind D2a is that objects made up of faces with more varied sizes (i.e. has very big, very small and in-between sized surfaces) should look different from objects made up of faces with more uniform sizes (i.e. has mostly big, mostly small or mostly average-sized surfaces). For example, a 3D model of a car can have relatively large surfaces (making up the roof and windows), very small surfaces (making up the nuts and bolts), and many sizes in between (e.g. rear-view mirror) due to the discrete nature of mesh presentation for free form surfaces. A simple cube on the other hand, is made up of six equally-sized faces.

The **area ratio** *ar* of a face pair (F_i,F_j) of an object *O* is defined as the area of the smaller face over the area of the larger face:

$$ar(F_i, F_j) = \frac{\min(area(F_i), area(F_j))}{\max(area(F_i), area(F_j))}, \quad (1)$$

Allying the equation (1) to every face pair, we can derive the distribution of area ratio of the object.

A practical way to compute the area variability (or uniformity) of an object's faces is by sampling the **area ratios** of these faces in the following procedure:

```
ComputeAr(O)
    // Input: a 3D object O of M faces (F1..FM)
    // Output: histogram of area ratio of the faces
    choose N faces with probability of being chosen proportional to the area of
    each face
    for each face pair (Fi, Fj) of the selected N faces,
      if area(Fi)>area(Fj)
        ar = area(Fj)/area(Fi)
      else
        ar= area(Fi)/area(Fj)
      index = ⌊ar* numBins ⌋
      Ratio_Histogram[index] := Ratio_Histogram[index] + 1
```

Where `numBins` (=2 in our experiment) defines the granularity of the frequency distribution, and `Ratio_Histogram` contains the area ratio frequency distribution.

Since the car has varied polygon sizes, while the cube has uniform polygon sizes, one can expect the ratios of polygon areas on the car to be more variable than those of a simple cube. In fact, it can be easily observed that the only possible ratio between areas of faces in the cube is 1.0, since all faces have the same area. For the car, ratios of areas should tend to be lower than 1.0 and closer to 0.0 since one random face is likely to be much bigger or smaller than another random face. Thus, one can expect that in the frequency distribution of area ratios, a car's graph should have higher frequencies for lower ratios (i.e. the graph should skew higher to the left) due to the variances in area ratios.

It can also be conjectured that for 3D models which are made up mostly of same-sized polygons, the frequency distribution of area ratios should be lie near the value 1.0 (i.e. the graph should skew higher to the right), since one random face should not be much bigger or smaller than another random face. This conjecture can be verified by the graphs on Figure 3.

It can be seen that for objects with varied face areas (Figure 3a), the probability of two random faces having different areas is greater (therefore most ratios are below 1.0), while for objects with uniform face areas (Figure 3b), the probability of two random faces having the same area is greater (therefore most ratios are exactly 1.0).

The ratio of areas shape feature is stored in the second dimension of a 2D array whose first dimension contains the D2 distribution. The second dimension only has two bins for our experiment: the first to store the frequency of ratios that are < 1.0, and the second to store the frequency of ratios that are exactly 1.0.

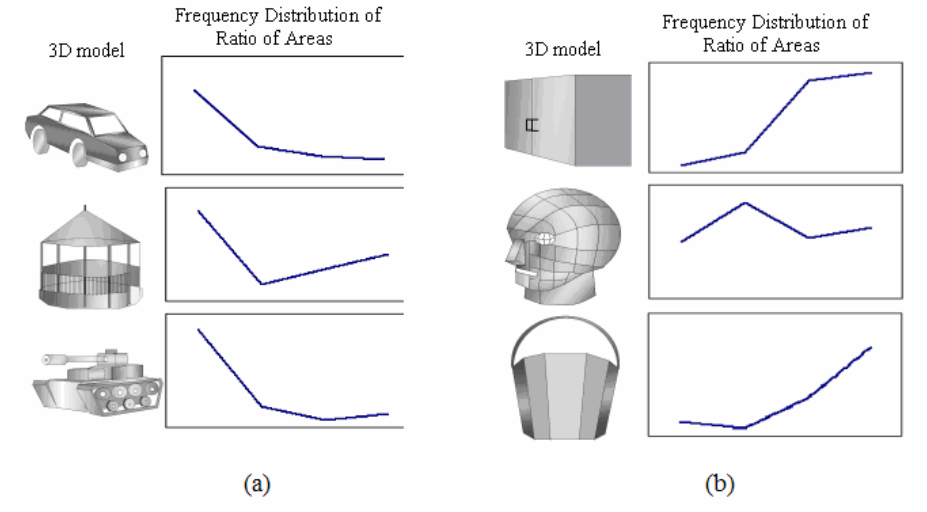

Fig. 3. Shown are 3D objects: (a) with **varied** face areas, (b) with relatively **uniform** face areas; and the graphs of their respective area ratio frequency distributions. The x-axis represents the ratio of areas (0..1), while the y-axis is the frequency. For the objects are made up of varied face areas, the ratios between these areas (small over big) tend to be small. Thus the graph shows higher frequency for small ratios. For the objects are made up of face areas of roughly similar size, the ratios between these areas (small over big) tend to be near 1.0. Thus the graph shows higher frequency for ratios near 1.0.

3 Experiments and Results

D2a was implemented in C++ and compiled using Microsoft Visual Studio.NET 2003. The Princeton Shape Benchmark was used to compare the results with Shilane's [3]. As with Shilane, only the test set of 907 models were used for all the algorithms (the training set was not used). For the D2 component (which is the first dimension of the D2a descriptor), a sample size of 1024 random points were generated and 64 bins were used. The dimensions of the D2a array were 64 x 2 unsigned integers, which took up 512 bytes (4 bytes for each unsigned integer).

The scale was normalized by aligning the mean [2, 3]. The L_1 Manhattan Distance was used to measure the distance between two D2a histograms. [2]

3.1 Performance Metrics

In order to compare the results directly with the results of Shilane, we used the same utility program provided by the PSB, `psbtable.exe`, to compute the same performance metrics: Nearest Neighbor, Tier 1, Tier 2, E-Measure, and Discounted Cumulative Gain (DCG). `psbtable.exe` generates these measurements given a distance matrix binary file (which is the output of our shape matching program) and a classification (.CLA) file. [3]

Nearest Neighbor, Tier 1 and Tier 2 measurements were also used as performance metrics by Osada [1], Kazhdan [5], and Ohbuchi [2] although these studies did not

use the Princeton Shape Benchmark 3D model database. For all performance metrics, higher numbers are better.

1. Nearest Neighbor (NN)

Using each model as query, NN is the percentage of cases where the returned top match is from the query model's class.

2. Tier 1 (T1)

Given a query model with class size N, retrieve the top N-1 matches. Tier 1 is the percentage of the N-1 matches retrieved that belong to the query model's class.

Each model in the class has only one chance to be in the first tier, since the number retrieved = number of possible correct matches. Therefore, only a perfect matching algorithm can return a T1 measure of 100%.

3. Tier 2 (T2)

Given a query model with class size N, retrieve the top 2(N-1) matches. Tier 2 is the percentage of the 2(N-1) matches retrieved that belong to the query model's class.

The number retrieved is 2 times the number of possible correct matches. Therefore, the highest possible score for T2 is 50%. However, a score of 50% doesn't mean that the matching algorithm is perfect, only that it is good enough to return all relevant matches if the retrieval size is big enough (twice the class size of the query -1).

4. E-Measure

The E-Measure is the precision and recall values combined into one value. The intuition is that a user of a search engine is more interested in the first page of query results than in later pages. So, this measure considers only the first 32 retrieved models for every query and calculates the precision and recall over those results.

The E-Measure is defined as:

$$E = 2 / (1/\text{Precision} + 1/\text{Recall})$$

5. Discounted Cumulative Gain (DCG)

DCG gives a sense of how well the overall retrieval would be viewed by a human. Correct shapes near the front of the list are more likely to be seen than correct shapes near the end of the list. With this rationale, discounted cumulative gain is calculated as: 1 + sum 1/lg(i) if the ith shape is in the correct class.

6. Normalized DCG

This metric normalizes the DCG values so that the average DCG becomes 0.0.

$$\text{Normalized } DCG_i = DCG_i / \text{Average DCG} - 1$$

Where:

DCG_i = DCG score of the i^{th} shape descriptor
Average DCG = average of all DCG scores

Normalized DCG shows how much better or worse than the average a shape descriptor's DCG score is. Since it is just another way of presenting DCG scores, it

always results in the same ranking as DCG. Therefore, although we present both DCG and Normalized DCG in the results, we count them as one metric (we use five metrics in all).

4 Results and Conclusions

Table 1 summarizes the results of Shilane [3] for 12 algorithms (first part) and our results for D2a (second part). Table 2 shows a summary of D2a's performance compared to D2. The blue horizontal bars on the first part of the table indicate where the D2a performance falls.

D2a significantly outperformed D2 in three out of five performance metrics. In T1, T2 and E-Measure, the improvements ranged from 15% – 92%. On the other hand, performance decline in the two other metrics was not as substantial: only 5.47% in NN and 3.39% in Normalized DCG.

Table 1. Summary of results for for the D2a algorithm, compared directly with 12 other algorithms. The blue horizontal bars on the first part of the table indicate where the D2a per-formance (using the same test set) would fall.

Shape Descriptor	Storage Size (bytes)	NN (%)	T1 (%)	T2 (%)	E – Meas. (%)	DCG (%)	Norm. DCG (%)
LFD	4,700	65.70	38.00	48.70	28.00	64.30	23.05
REXT	17,416	60.20	32.70	43.20	25.40	60.10	15.02
SHD	2,184	55.60	30.90	41.10	24.10	58.40	11.76
GEDT	32,776	60.30	31.30	40.70	23.70	58.40	11.76
EXT	552	54.90	28.60	37.90	21.90	56.20	7.55
SECSHEL	32,776	54.60	26.70	35.00	20.90	54.50	4.30
VOXEL	32,776	54.00	26.70	35.30	20.70	54.30	3.92
SECTORS	552	50.40	24.90	33.40	19.80	52.90	1.24
CEGI	2,056	42.00	21.10	28.70	17.00	47.90	-8.33
EGI	1,032	37.70	19.70	27.70	16.50	47.20	-9.67
D2	136	31.10	15.80	23.50	13.90	43.40	-16.94
SHELLS	136	22.70	11.10	17.30	10.20	38.60	-26.13
					Avg. DCG	52.45	
D2a (Test)	512	29.40	30.40	30.30	16.00	43.10	-17.52

Table 2. This shows a comparison of D2a's performance with that of D2. Positive numbers indicate an increase in performance, while negative numbers indicate a decline.

D2a Improvement over D2	NN (%)	T1 (%)	T2 (%)	E – Meas. (%)	DCG (%)	Normalized DCG (%)
	-5.47	92.41	28.94	15.11	-0.69	-3.39

4.1 Analysis

From the results obtained (shown on tables 1 and 2), we make the following observations. D2a underperformed in NN and DCG by 5.47% and 3.39%, respectively compared to D2. Both are "closest match" metrics, which indicates that D2 may be slightly better than D2a in placing the correct matches at the top of the retrieval list, even though D2a retrieves more correct matches than D2. D2a outperformed D2 in T1, T2 and E-Measure by 92.41%, 28.94% and 15.11%, respectively. T1, T2 and E-Measure are all related to precision and recall. T1 and T2 are precision values at specific retrieval sizes, while E-Measure is a combination of precision and recall for the top 32 matches.

The results suggest that D2a provides better precision-recall results than D2 when the retrieval size is at least 32 (and at most twice the query model's class size). This means that when a user of a search engine looks for a 3D model and requests at least 32 matches, D2a will likely return more correct matches than D2 (as much as 92% more, from our experiments). However, it is possible that D2 will show more correct matches (around 5% more) in the first page of the results than D2a.

This study shows that combining two shape features – distance between point pairs and ratio of areas of surfaces – into a single shape descriptor can result in better overall classification and retrieval performance. However, two problems may still persist in the D2a shape descriptor. First, the computation is high ($O(n^2)$) as all the ratios need to be calculated. Second, 3D models that are represented by surfaces having the same areas, such as a sphere and a cube, cannot be differentiated by the D2a descriptor, since both have the same ratio distribution. Further studies can be made to address these issues. On the other hand, the storage requirements of D2a are still well below other algorithms with comparable performance.

References

1. Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D.: Matching 3d models with shape distributions. Shape Matching International. (2001) 154-166
2. Ohbuchi, R., Minamitani, T., Takei, T.: Shape-Similarity Search of 3D Models by using Enhanced Shape Functions. Proceedings of the Theory and Practice of Computer Graphics. IEEE Computer Society (2003)

3. Shilane, P., Min, P., Kazhdan, M. and Funkhouser, T.: The Princeton Shape Benchmark. Shape Modeling International, Genova, Italy. (2004)
4. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A. and Dobkin, D.: A search engine for 3d models. ACM Transactions on Graphics. Vol. 22(1), (2002)
5. Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. Eurographics Symposium on Geometry Processing. (2003)