2009 Fifth International Joint Conference on INC, IMS and IDC

Combining Shape and Color for Retrieval of 3D Models

Conrado R. Ruiz, Jr.
College of Computer Studies
De La Salle University
Manila, Philippines
e-mail: cons.ruizjr@delasalle.ph

Rafael Cabredo
College of Computer Studies
De La Salle University
Manila, Philippines
e-mail: raffy.cabredo@delasalle.ph

Levi Jones Monteverde
International Technology Organization
Citibank
Singapore
e-mail: Levijones.ait.monteverde@citigroup.com

Zhiyong Huang
Institute for Infocomm Research (I^2R)
A*STAR
Singapore
e-mail: zyhuang@i2r.a-star.edu.sg

Abstract— **Current research on the retrieval systems for 3D models focuses on using the shape of the models to facilitate search and retrieval. This paper explores the possibility of augmenting the existing 3D shape-based similarity measures by combining shape and color. First, a new descriptor was developed based on the D2 shape descriptor. 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). Second, this research introduces the use of the color features of a 3D model in combination with the shape features to determine similarity. The research involves the study and adoption of an existing 2D color-based similarity measure for 3D models. The analysis of the results is based on the precision and recall of both approaches.**

Keywords-shape retrieval; shape database; color retrieval; 3D models; similarity measures

I. Introduction

The rapid expansion of the internet and advancements in computer hardware technology have facilitated the proliferation of 3D models on the web. These 3D models have become widespread in a variety of applications including computer graphics, mechanical CAD, molecular biology, and medicine [13].

The growing usage and demand for these 3D models is brought about by new scanners and interactive tools that help produce 3D models practically and cost efficiently, graphics hardware that is becoming faster and more affordable (at 3 times Moore's Law), and the Internet that contributes to wide and rapid distribution of 3D models [4]. Therefore, there is now a great need for tools that will be able to analyze, classify, index, and store these models into a large database for easy access and retrieval.

An important research area is the efficient storage and retrieval (shape-matching) of desired 3D models from such databases. For example, if one needs a 3D model of an airplane, searching a database using "airplane" as keyword may not be enough, since the filenames may not be descriptive, 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 model, or even hand-drawn 2D sketches of the desired model. Such a system was developed and is available online [4], which can even be extended to even composing 3D models [8].

The similarity 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.

II. Related Work

Several studies have already been conducted to search and retrieve 3D models. One of the problems being addressed is the efficient retrieval of different multimedia. A study conducted by [4] observed that people find the use of text as the simplest approach to facilitate 3D model search and retrieval. However, there are scenarios wherein using the text annotation will fail. For instance, not all objects are annotated (e.g. "cty123.wrl"). On the other hand, some annotated objects would have vague descriptions (e.g. "red.wrl", "big.wrl", searching for "faces" – i.e., human and not polygonal).

This method of text annotation also reduces the amount of information that can be represented, especially if VRML is used to represent the 3D model. VRML is optimized for visualization of the model but lacks the semantic definition and structure [3]. This results in a "polygon soup" representation of models with no useful descriptive element. Thus, 3D similarity algorithms have to be prepared to consider models, which are likely to be incomplete.

Due to these problems, researches are now focusing more on retrieving 3D models based on similarity of content. Different approaches have been studied in other fields

978-0-7695-3769-6/09 $26.00 © 2009 IEEE
DOI 10.1109/NCM.2009.140

including computer vision, computational geometry, computer aided design, and molecular biology [7]. In multimedia, most of the work done focuses on 2D images; only a few researchers work on 3D models.

One of the problems that need to be addressed is finding a similarity measure for 3D models that is invariant to transformations. A common approach used to solve this problem is to provide the 3D model with a shape descriptor in a form of a voxel grid [2]. The objective of this grid is to normalize the model into a standard pose (i.e. same rotation, scale, and translation) so that descriptors can be consistent.

Aside from the voxel-based normalization, there are other approaches. These approaches can be generally classified into two groups. First are studies that use 3D matching algorithms (i.e. Reflective Symmetry Descriptors [5] and Spherical Extent Functions [11]) which transform the model into a canonical form. The other group includes algorithms that construct descriptors or shape signatures that are invariant to any kind of transformation. Examples for these are Shape Histograms [2], Shape Distributions [10][13], and Spherical Harmonics [6].

The studies mentioned above focus on shape based content to find a match with a given 3D model. The studies show satisfactory results using the algorithms. However, similarity is not only limited to comparing the shapes of two objects. Other attributes can be used as well, such as color and texture. Thus, in this paper we also explore the possibility of using a multi-feature similarity measure considering both the shape and the color of the 3D model.

III. OVERVIEW

In this section, we provide the theoretical framework behind a shape descriptor for comparing 3D models and color-based similarity measures for 3D models. We discuss the general approach to shape comparisons of 3D models, the specific shape distribution we are extending and general color similarity measures considering perceptually similar colors.

A. General Approach for Shape Comparison of 3D Models

A typical approach in comparing the shape of two 3D models is to apply two steps [4]:

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.

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. Fig. 1 demonstrates this two-step process.

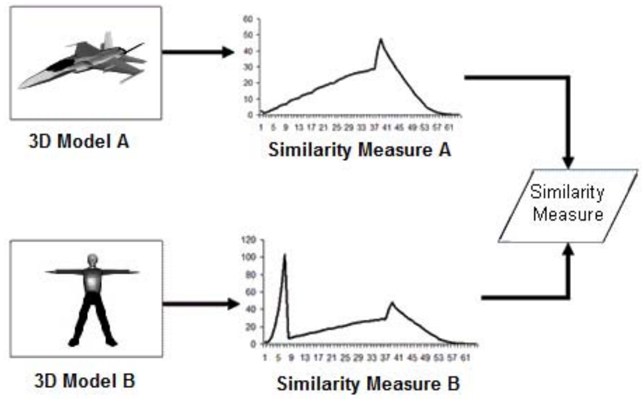

Figure 1. The general approach to shape-based comparison of 3D models.

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 [6], unlike with scale or translation. It is also possible to normalize the shape descriptor itself instead of the 3D model.

B. Shape Similarity Measures

There are numerous shape-based similarity measures developed for 3D models. This paper focuses on the shape distribution, D2 [10]. 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 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, et al. noted that the D2 shape distribution is distinctive for each 3D model [10], 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 *1,024* samples points, resulting in *1,024*1,024/2 + 1,024 = 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 are several ways to normalize a D2 distribution. The two simplest and most effective are aligning by mean and aligning by maximum distance.

After the shape distribution of a model has been computed it can now be compared with other 3D models. First the difference is determined based on a distance measure, such as the City-block distance (the L1-norm).

$$d_1(\overline{XS},\overline{YS}) = \sum_{i=1}^{N} | xs_i - ys_i | \quad (1)$$

Then the shape similarity function $\boldsymbol{SIM_{shape}}$ is computed using the following formula:

$$SIM_{shape}(\overline{XS},\overline{YS}) = 1 - d_1(\overline{XS},\overline{YS}) \quad (2)$$

C. Color Similarity Measures

We extend the color similarity measure from 2D images to 3D models. Once the color feature vector of two models has been obtained, computing the similarity of the color features is done.

Let ***XC*** and ***YC*** be the normalized color feature vectors of the query and 3D model collection, respectively. Computation of the final color similarity, denoted by $\boldsymbol{SIM_{color}}$ is:

$$SIM_{color}(XC,YC) = \sum_{i=1}^{m} [w_i \cdot xc_i] \quad (3)$$

where:

XC = the color feature vector of the query 3D model
YC = the color feature vector of the 3D model in the collection
i = a color in the histogram or feature vector
$\boldsymbol{w_i}$ = the overall similarity contribution by color ***i***
m = the number of elements in the color feature vector

The final color similarity describes a value that gives different weight to the contribution of each color ***i*** based on its percentage in the model. Formally $\boldsymbol{w_i}$, is written as:

$$w_i = sim(xc_i, yc_i) \cdot (1 + SIM_{PerceptualCol}(xc_i, YC)) \quad (4)$$

where:

i = the color being computed for
$\boldsymbol{sim(xc_i, yc_i)}$ = the similarity of the ***i***th color in ***XC*** and ***YC***
$\boldsymbol{SIM_{PerceptualCol}(xc_i, YC)}$ = the perceptual similarity of Color $\boldsymbol{xc_i}$ in ***YC***

This equation adds the perceptually similar colors, $\boldsymbol{SIM_{PerceptualCol}}$, with a lower contribution factor of the similarity of the exact color, $\boldsymbol{sim(xc_i, yc_i)}$. The similarity of two colors, denoted as ***sim(xc, yc)***, and the similarity contribution of perceptually similar colors for color ***i*** are defined as:

$$sim(xc, yc) = 1 - \frac{|xc - yc|}{\max(xc, yc)} \quad (5)$$

$$SIM_{PerceptualCol}(xc_i, YC) = \sum_{j=1}^{m} sim(xc_i, yc_j) \times \delta_{i,j} \quad (6)$$

where:

j = the color bin which color *i* is perceptually similar to
$\boldsymbol{\delta i,j}$ = the perceptual similarity between color *i* and *j*

The perceptual similarity between colors *i* and *j* is an element in the *m x m* perceptual similarity matrix denoted as:

$$D_{sim}(i,j) = \begin{pmatrix} 1 & \delta_{0,1} & \delta_{0,2} & \cdots & \delta_{0,m} \\ \delta_{1,0} & 1 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & 1 & \delta_{m,m-1} \\ \delta_{m,0} & \delta_{m,1} & \cdots & \delta_{m,m-1} & 1 \end{pmatrix} \quad (7)$$

Each element $\boldsymbol{\delta}$ is pre-computed as the maximum distance between all CIE Luv colors pairs in the color bin *i* and *j*. Formally it is written as:

$$\delta_{i,j} = \max(\sqrt{(L_{ik} - L_{jk})^2 + (u_{ik} - u_{jk})^2 + (v_{ik} - v_{jk})^2}) \quad (8)$$

$k = 1$ to b

where:

b = the number of actual colors in the color bin.
$\boldsymbol{L_{ik}}$ = the ***k***th ***L*** component of the CIE Luv color in color bin ***i***
$\boldsymbol{u_{ik}}$ = the ***k***th ***u*** component of the CIE Luv color in color bin ***i***
$\boldsymbol{v_{ik}}$ = the ***k***th ***v*** component of the CIE Luv color in color bin ***i***
$\boldsymbol{L_{jk}}$ = the ***k***th ***L*** component of the CIE Luv color in color bin ***j***
$\boldsymbol{u_{jk}}$ = the ***k***th ***u*** component of the CIE Luv color in color bin ***j***
$\boldsymbol{v_{jk}}$ = the ***k***th ***v*** component of the CIE Luv color in color bin ***j***

IV. SIMILARITY MEASURE EXTENSIONS

This section explains the two extensions we propose to augment existing shape-based similarity measure. The first extension is created by adding a new dimension to the D2 shape descriptor. The second approach used the color feature of 3D models to create a multi-feature similarity measure.

A. 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:

$$ar(F_i, F_j) = \frac{\min(area(F_i), area(F_j))}{\max(area(F_i), area(F_j))} \quad (9)$$

Allying the equation 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 Fig. 2.

It can be seen that for objects with varied face areas (Fig. 2), 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 (Fig. 2), 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.

B. *Multi-feature Similarity Measure*

The color feature of a 3D model gives the global color characteristics of the model. It is represented as a 159-bin color histogram. Each bin represents the percentage of the color in the model. Similar to shape features, the color feature vector of all 3D models are also computed before running a query.

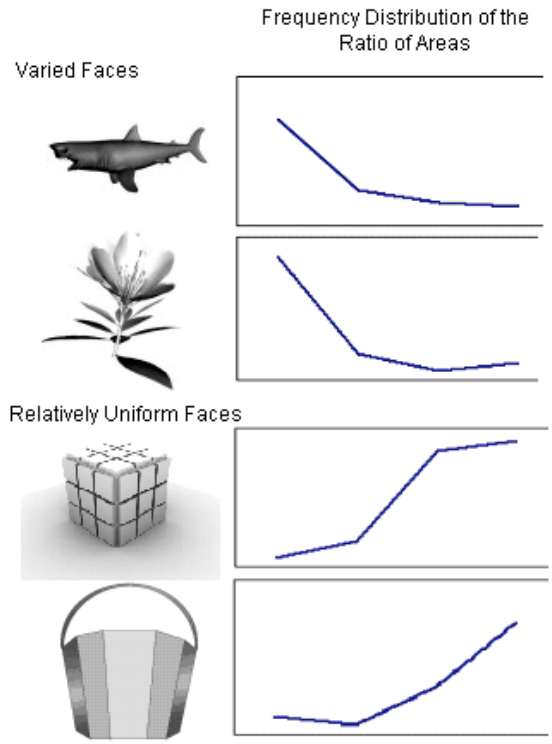

Figure 2. Shown are 3D objects and the graphs of their respective area ratio frequency distributions. The x-axis represents the ratio of areas, while the y-axis is the frequency.

The algorithm for obtaining the color features is as follows:

```
GetColor()
Open the colored OFF file
   Extract all the polygons of the 3D model
   Get one polygon
Compute the area of the polygon
   Get the RGB color of the polygon
   Convert RGB color to CIE Luv
   Identify the index of the color in the
      159-bin histogram.
   Add the area to the identified bin and to
      the total surface area of the polygon.
Normalize the color histogram.
   For all bins in histogram, divide the
      value by the total area of the model.
   The total area is computed as the sum of
      all the surface area of polygons.
Write the histogram into the file
COLOR_FV.COD
```

The similarity of two 3D models is described as follows:

Let { ***XC***, ***XS*** } and { ***XC***, ***YS*** } be the color and shape vectors that fully describe two models ***X*** and ***Y***, then the similarity measure of 3D models ***X*** and ***Y***, denoted as ***SIM***(***X***, ***Y***), is given as:

$$\boldsymbol{SIM(X,Y)} = \sqrt{\alpha \boldsymbol{SIM}_{color}(\boldsymbol{XC},\boldsymbol{YC}) + \beta \boldsymbol{SIM}_{shape}(\boldsymbol{XS},\boldsymbol{YS})} \quad (10)$$

where:

SIM_{color}, SIM_{shape} = color and shape similarity functions
α, β = non-negative weighting factor for the color and shape similarity measures, respectively
XS = shape feature vector of the query 3D model
YS = shape feature vector of the 3D model from the database
XC = color feature vector of the query 3D model
YC = color feature vector of the 3D model from the database

V. RESULTS

Precision and recall were used to evaluate the improvements of the different approaches. Recall measures the ability of the system to retrieve all models that are relevant. Precision measures the ability of the system to retrieve only the relevant models.

A. 3D Model Collection

The 3D model collection is composed of different 3D models in the Object File Format (OFF). There were 2 kinds of OFF files used. One OFF is for the shape information, which is obtained by converting VRML models into OFF using 3D model tools. The other OFF is a modified version of the original OFF. In addition to the shape information it also includes the RGB color information for each of the polygons in the 3D mode l. For faster performance, the shape and color feature vectors of the models are preprocessed. Preprocessing includes computing the feature descriptions of the models before the search begins.

At the start of this research it was desired to collect only models with the number of polygons greater than 10,000 so that the color information can be diverse. Upon collection of the models, it was discovered that the high quality models use only 1 color (i.e., gray, or white).

One example of this is the plant model category. To determine the performance of adding color information to the models, lesser quality models were also included for testing. The test data contain an average of 2440 vectors, 3432 polygons and 3 colors. The models fall under 26 categories. The categories with the highest number of models were used for comparing the performance of the different similarity measure constructed.

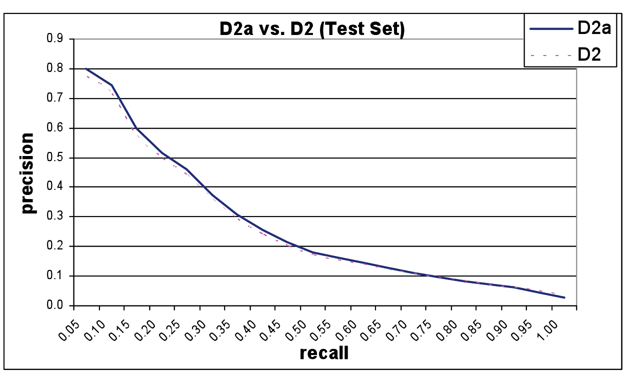

Figure 3. Precision-Recall plot for D2 and D2a using the TEST set of 907 models

B. D2a Shape Descriptor Results

Based on the cases that were tested, the best results were obtained using the following parameters:

Sample points**:** *1024*
Distance bins/Area bins**:** *64 / 4*
Normalization method: *Align by Max*
Distance Metric: *L1 Manhattan Distance*

The Princeton Shape Benchmark (PSB) utility program psbplot.exe was used to generate the precision-recall values, which were plotted using Microsoft Excel. psbplot.exe takes the distance matrix binary file (output of the proponent's software) and the PSB class file (.cla) used, and outputs the precision-recall values to a text file.

The precision-recall plot for the test data set in Fig. 3 shows only a slight improvement for D2a over D2. Numerically, the average precision for D2a was 26.93% vs. 26.11% for D2. This is an improvement of 3.16%.

C. Multi-feature Similarity Measure

An experiment was done to show the result of using the multi-feature similarity measure. This experiment examines the behavior of the multi-feature similarity feature using different threshold levels and different color (α) and shape (β) weight values.

Using L_1*-norm* as the distance metric, results with the Humanoid model set having the best F-measure (64.52) obtained with a color weight(α) of 10%, shape weight(β) of 90% and threshold at 70% (Table 1).

Visually, the precision-recall plot for the test set in Fig. 6 above shows an obvious improvement for multi-feature retrieval (using both shape and color) over shape or color alone. Numerically, the average precision for the multi-feature similarity was 53.90% vs. 50.50% for shape. This is an improvement of 6.73% over shape similarity.

TABLE I. SUMMARY OF SETTINGS AND RESULTS USING L1-NORM SIMILARITY MEASURE

	Best settings			
	α	**β**	**Threshold**	**F-measure**
Airplanes	50%	50%	25%	36.87
Chairs	40%	60%	60%	48.78
Humanoids	10%	90%	70%	64.52
Plants	60%	40%	75%	58.73

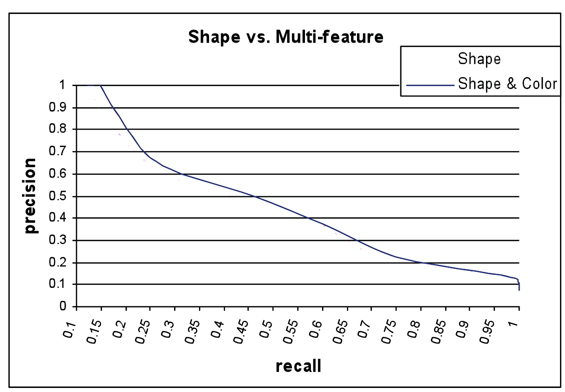

Figure 4. Precision-Recall plot for Multi-feature Measure using α=10%, β= 90% and Shape-based Similarity Measure

VI. Conclusion and Recommendations

This study shows that combining two shape features (distance between point pairs and ratio of areas of surfaces) into a single shape descriptor; and combining shape and color features can result in better overall classification and retrieval performance.

From the experiments it was observed that including color features improves the precision and recall of a retrieval system. However, this is not always true for all cases. For instance, simple models having only less than 5,000 polygons must emphasize more weight on the shape of the object to have better results. A color weight of 1%-10% and a threshold of 70%-75% are recommended for simple models. Improvement of retrieval for simple models was only from 1%-5%. In some cases performance even decreased.

On the other hand, the developed similarity measure proved to be very effective in increasing precision and recall for more complicated models (more than 10,000 polygons). One contribution of this research is analyzing the effects of combining color information with shape information obtained using the Depth Buffer-based feature descriptor. In the experiments in section V a bigger weight was usually given to shape feature over the color feature, because it was observed that the shape feature is still a better discriminator for 3D models.

Also, it was observed that the models collected contain very few colors. On average, the models only had three colors. Normally, models would just have the color gray or white as its color for the entire model. Constructing a database of models that have more color in them may change the results of this research and is thus recommended for future work. For the models that have more than one color, it was observed that the colors were used to identify different parts of the model. For example, the wings of the airplane were colored differently from the body, wheels or propeller. This color categorization of the model parts may be useful if the color of the model can be used as a filter in determining parts of an object.

Additional research is needed to determine the effect of applying the similarity measure on other feature descriptors, such as those that are 3D-geometry based, statistical, topological, or features that are functions on a sphere.

Furthermore, 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.

With regard to the multi-feature similarity measure based on shape and color, it must be noted that most high quality models nowadays are texture mapped rather than just simple material coloring. The possibility of incorporating the textures of 3D models can also be explored to replace the inherent color of the polygons. The color features of the textures can be extracted and similarity measures for 2D images can be used.

References

[1] M. Ankerst, G. Kastenmüller, H. Kriegel, and T. Seidl, "Nearest neighbor classification in 3d protein databases," Proc. 7th Int. Conf. on Intelligent Systems for Molecular Biology. [online] Available: http://www.aaai.org (1999)

[2] D. Barker, "Methods for Content-Based Retrieval of 3D Model," 2nd Annual Conference on Multimedia Systems (January, 2003)

[3] R. Carey, G. Bell, and C. Marrin, "The Virtual Reality Modeling Language," ISO/IEC DIS 14772-1. [online]. Available: http://www.web3d.org/Specifications (April, 1997).

[4] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, "A Search Engine for 3d Models," ACM Transactions on Graphics. pp. 83–105 (2003)

[5] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A Reflective Symmetry Descriptor for 3D Models," Algorithmica, pp. 201-225 (October, 2003)

[6] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptor," Eurographics Symp. on Geometry Processing. (2003)

[7] H. Kriegel, and T. Seidl, "Geometric Search in 3D Protein Database," University of Munich. [online]. Available: www.dbs.informatik.uni-muenchen.de/Forshung/Similarity/Demos/protein/ (2003)

[8] J. Lee, and T. Funkhouser, "Sketch-Based Search and Composition of 3D Models," Proc. EUROGRAPHICS Workshop on Sketched-Based Interfaces and Modeling 2008 (2008)

[9] R. Ohbuchi, T. Minamitani, and T. Takei, "Shape-Similarity Search of 3D Models by using Enhanced Shape Functions," Proc. of the Theory and Practice of Computer Graphics. IEEE Computer Society (2003)

[10] R. Osada, T. Funkhouser, B. Chazelle and D. Dobkin, "Matching 3D Models with Shape Distributions," Shape Matching Int., 154-166. (2001).

[11] C. Ruiz, L. Monteverde and Z. Huang, "A Shape Distribution for Comparing 3D Models," Proc. International Multimedia Modeling Conference 2007. pp 54-63 (MMM 2007)

[12] D. Saupe, and D. Vranić, "3D Model Retrieval with Spherical Harmonics and Moments," in: Proc. DAGM *2001* (editors B. Radig and S. Florczyk), Munich, Germany, pp. 392-397. (September, 2001).

[13] P. Shilane, and T. Funkhouser, "Selecting Distinctive 3D Shape Descriptors for Similarity Retrieval," Shape Modeling International. (June, 2006).