3D Model Matching Based on Silhouette Image Matching

YOSHIHIRO OKADA
Graduate School of Information Science and Electrical Engineering, Kyushu University
6-1 Kasuga Koen, Kasuga, Fukuoka 816-8580 JAPAN

Abstract: - Recently a lot of 3D models have been created and stored because of the great demand in CG animation productions and 3D game productions. In this situation, we need the 3D model database system that allows us to retrieve required models easily and accurately. So the authors have been studying 3D model database systems. The main questions are that what is the best feature as the similarity measure among 3D models and what is the best way to specify the query. In this paper, the authors propose the silhouette image matching as the answer of the first question and the hand sketch image as the answer of the second question. The authors have developed a 3D model database system using the silhouette image matching as the similarity measure, and it accepts the hand sketch image as the query. This paper proposes such a database system and describes its availability showing its evaluation results.

Key-Words: - 3D model database, Hand sketch images, Silhouette image matching, 3D model matching.

1 Introduction

Recent advances in computer hardware technology made it possible to render 3D images in real time even if using a personal computer. As a result, many computer animation creation products were made and computer animations were created in many game productions and movie productions. In this situation, we need the 3D model database system that allows us to retrieve required models easily and accurately. So we have been studying 3D model database systems. The main questions are that what is the best feature as the similarity measure among 3D models and what is the best way to specify the query. This paper proposes the silhouette image matching as the answer of the first question and the hand sketch image as the query. We have developed a 3D model database system using the silhouette image matching as the similarity measure and it accepts the hand sketch image as the query. In this paper, we introduce such a database system and describe its availability showing its evaluation results.

Already some researches on the 3D model matching have been done. Paquet and Rioux propose a 3D model database system. This system employs many popular 3D model matching algorithms [1]. Osada et al proposed 3D model matching algorithm using distribution of distances between any two random points on the surface of a 3D model [2]. This method is called $D2$ and it has good evaluation results reported in their paper. Hiraga et al proposed new topology matching technique using Multi-resolutional Reeb Graphs (MRGs) [3]. Bandlow et al proposed recognition method of objects using their silhouette [4]. In this case, objects are not 3D models but 2D images. We have not met researches on the 3D model matching using their silhouette images. Our proposed method employs the silhouette image matching for the 3D model matching and we obtained good evaluation results.

The remains of this paper are organized as follows: Section 2 briefly describes overview of the 3D model database system and explains detail of the silhouette image matching. Section 3 shows experimental results to clarify the availability of the system. Finally Section 4 concludes the paper.

2 3D Model Matching

Basic concept is very simple as shown in Figure 1. First of all, the system automatically generates three orthogonal silhouette images of each 3D model for all 3D models and stores them as their feature information. In this case, silhouette images for each 3D model are generated along with its three principal axis directions. Next, the user enters hand sketch images as the query to search his/her required models. After that, the system retrieves the best match models according to the silhouette image matching result.
2.1 Principal axes

To generate silhouette images from a model automatically, it is necessary to determine view directions from the eye-position. Very common way is to use the principal axes [1][5]. Principal axis directions are determined as eigen vectors of the moment of a 3D model calculated by the following expression (1).

$$
I = \begin{bmatrix}
I_{xx} & I_{xy} & I_{xz} \\
I_{yx} & I_{yy} & I_{yz} \\
I_{zx} & I_{zy} & I_{zz}
\end{bmatrix}
$$

Here, each element $I_{qr}$ ($q, r \in \{x, y, z\}$) of the matrix $I$ is calculated by the following equation (2).

$$
I_{qr} = \frac{1}{n} \sum_{i=1}^{n} \left[ S_i \times (q_i - q_{CM}) \times (r_i - r_{CM}) \right]
$$

$n$ is the number of faces, $S_i$ is the area of $i$-th face. $q_i, r_i$ are $q, r$ elements of $i$-th face, $q_{CM}, r_{CM}$ are $q, r$ elements of the center of mass. Finally eigen vectors are calculated by the following equation.

$$
\lambda_i = \alpha_i, i \in \{1, 2, 3\}
$$

The eigen vector having the biggest eigen value is the first principal and the eigen vector having the second biggest eigen value is the second principal. The remainder is the third principal. Positive or negative directions are determined by distribution of amount of surface area.

2.2 Silhouette image generation

As shown in Figure 2, the system generates three black and white silhouette images each of those is corresponding to each of three principal axes by actually rendering such 3D scenes on a computer screen. A background color is set white and a 3D model color is set black. Each of three scenes is rendered with varying its view direction corresponding to each of three principal axes. Its calculation cost is very small because the most parts of its rendering process are supported by the hardware. After rendering one scene, its image is normalized into a square shape in a specific resolution, e.g., eight by eight, $16 \times 16$, $32 \times 32 \ldots$, and then stored in a storage device. The file size of the image of $32 \times 32$ pixels resolution is 128 bytes. The total file size of three images is 384 bytes. This value is enough small in comparison with the file size of a 3D model.

As shown in Figure 2, to generate a square image, there are two kinds of normalizations, (A) and (B). The (A) uses the maximum length of a 3D model’s bounding box as a size of the square. In this case, information of ratio among a width, height and depth size of 3D model’s bounding box is kept. Therefore, this information influences similarity measure strongly. On the other hand, the case (B), a 3D model’s bounding box is normalized into a cube shape and the 3D model is also normalized. Then three silhouette images are generated and stored. In this case, information of ratio among a width, height and depth size of 3D model’s bounding box is ignored. To compensate this, scaling information is used as another part of similarity measure as explained later.

2.3 Silhouette image matching

The similarity measure is treated as that if two 3D models have a smaller error between their silhouette images, they have a larger similarity.

2.3.1 Error between two models’ silhouette images

Error between two model’s silhouette images is determined as the ratio of number of the different pixels to the total pixels as follows.
When calculating error between two model's silhouette images, actually minimum value is used among four error values of four kinds of images because four image patterns exist as shown in Figure 3. They are an original, a vertical flip, a horizontal flip, and a 180 degrees rotation image. As previously mentioned, positive or negative directions of first and second principal axis are determined by distribution of amount of the surface area of a model. So if these directions are different between two models to be compared, their three image directions become different as shown in Figure 3. Case 1 is that the first principal axis direction is opposite, Case 2 is that second principal axis direction is opposite, and Case 3 is that both first and second principal axis are opposite. Then these three images, i.e., a vertical flip, a horizontal flip, and a 180 degrees rotation image, and the original image are necessary to compare.

Each 3D model has its three silhouette images. Total error is determined as mean value of three image matching errors as follows:

$$\text{Error}_{\text{two images}} = \frac{\text{different pixels}}{\text{total pixels}}$$  (4)

For example, in the case of $$P_{\text{Error}}$$ between a model A and a model B, $$\text{Error}_1$$ is an error value between model A’s front image and model B’s front image, $$\text{Error}_2$$ is an error value between model A’s top image and model B’s top image, and $$\text{Error}_3$$ is an error value between model A’s side image and model B’s side image. In the other case of $$P_{\text{Error}}$$, $$\text{Error}_1$$ is between model A’s front image and model B’s front image, $$\text{Error}_2$$ is between model A’s top image and model B’s side image, and $$\text{Error}_3$$ is between model A’s side image and model B’s top image.

Then we obtain final error value between two 3D models by the next equation.

$$\text{Error}_{\text{image}} = \min(P_{\text{Error}_1}, P_{\text{Error}_2}, \cdots, P_{\text{Error}_6})$$  (7)

2.3.2 Error for scaling factor between two models

As previously mentioned, there are two sets of normalized silhouette images (A) and (B) as shown in Figure 2. As a result of similarity evaluation, the set (B) has a good result. However, in this case, the scaling factor of a 3D model disappears. To compensate this defect, information of ratio among a width, height and depth size of a 3D model should be used. Then we employ two scaling values, i.e., the ratio of the second principal axis length to the first principal axis length and the ratio of the third principal axis length to the first principal axis length.

Next equation (8) means an error between two 3D models concerning their scaling factor. Mean square root value is used since it is popular measure.

$$P_{\text{Error}} = \frac{1}{3} \sum_{j=1}^{6} \text{Error}_j$$  (6)
Here $l_2, l_3$ are the ratio of the second principal axis length to the first principal axis length and the ratio of the third principal axis length to the first principal axis length respectively for one of the two 3D models to be compared. Similarly $m_2, m_3$ are the same elements for the other of the two 3D models.

### 2.3.3 Total error between two models

Finally we calculate a total error value using the next equation (9).

$$ Error_{scale} = \sqrt{(l_2 - m_2)^2 + (l_3 - m_3)^2} $$

Here $l_2, l_3$ are the ratio of the second principal axis length to the first principal axis length and the ratio of the third principal axis length to the first principal axis length respectively for one of the two 3D models to be compared. Similarly $m_2, m_3$ are the same elements for the other of the two 3D models.

### 2.4 Prototype system

Figure 4 shows a screen image of the prototype system. The system generates three silhouette images for each 3D model automatically by actually rendering them on a computer screen. Then the user can enter the query as his/her hand sketch images on a computer screen and retrieve required 3D models interactively. The three right upper small windows are for entering the query of three hand sketch images. The main window on the center shows a beethoven model and its bounding box rendered according to its three principal axes.

### 3. Evaluation results

This section presents evaluation results of silhouette image matching as the similarity measure for 3D model matching to clarify its effectiveness. An error value is calculated using the equation (9).

Here we use these values as $w_1 = 3, w_2 = 1$.

### 3.1 Preparation

We made a 3D model database by collecting its 3D models from Internet. It consists of 150 models as shown in Figure 5. We downloaded most of them from 3DCAFE [6]. Then we classified 150 models into 30 classes so that each class consists of five same kinds of models. Our prototype system generated three silhouette images for each model and totally 450 silhouette images for all models. The silhouette image resolution is $32 \times 32$ pixels. Its generation time for one set of three silhouette images is about two seconds with using a standard PC (Pentium II 450MHz, 256MB memory).

### 3.2 Evaluation measures

We use the same three evaluation measures described in [2]. They are "First Tier", "Second Tier" and "Nearest Neighbor" as follows.

**First Tier**

This criteria means the percentage of top $(k - 1)$ matches (excluding the query) from the query's class, where $k$ is the size of the class, i.e., $k$ is five in our 3D model DB. This criteria is stringent, since each
model in the class has only one chance to be in the first tier.

\[
\text{First Tier} = \frac{\text{Top } (k-1) \text{ matches}}{k-1} \quad (10)
\]

**Second Tier :**
This criteria is the same type of result, but for the top \(2(k-1)\) matches.

\[
\text{Second Tier} = \frac{\text{Top } 2(k-1) \text{ matches}}{k-1} \quad (11)
\]

**Nearest Neighbor :**
This criteria means the percentage of test in which the top match was from the query's class.

### 3.3 Results and discussion
Table 2 shows experimental results. Mean values, 0.452, 0.583 and 0.707, are corresponding to First Tire, Second Tire, and Nearest Neighbor respectively. Each of these three values is each mean of all the 3D model’s values. These values are better than values in the paper [2] since the corresponding values of \(D2\) actually applied on the same 3D model database are 0.317, 0.418 and 0.447. Osada et al present their own evaluation results in their paper. These values are 0.49, 0.66 and 0.66 for their own 3D model database. These values are depending on a 3D model database. Any way, our results are very close to them. Our total search time for 150 model queries is 87 seconds using PC (Pentium II 450 MHz CPU, 256MB memory) so that one search time is 0.58 second. This includes 150 times compares and 150 times file accesses. This value is not so bad since \(D2\) also takes about 0.15 second for 150 times compares. This includes 1024 \(int\) values actually reported in the paper. Furthermore, the size of a secondary information file concerning \(D2\) seems more than 4 Kbytes since it generates 1024 \(int\) values for one 3D model. This value is more than the size of our silhouette images file. Therefore, it can be said that our silhouette image matching is efficient as similarity measure for 3D models.

### Table 2 Evaluation result 1 (silhouette image resolution : \(32 \times 32\) pixels)

<table>
<thead>
<tr>
<th>Classes</th>
<th>First Tier</th>
<th>Second Tier</th>
<th>Nearest Neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jet Plane</td>
<td>0.5</td>
<td>0.95</td>
<td>0.6</td>
</tr>
<tr>
<td>Helicopter</td>
<td>0.4</td>
<td>0.55</td>
<td>0.6</td>
</tr>
<tr>
<td>Space Ship</td>
<td>0.2</td>
<td>0.35</td>
<td>0.4</td>
</tr>
<tr>
<td>Human</td>
<td>0.7</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Deep color columns are classes that have a greater value than 0.5, i.e., a better value rather than others. These classes apt to have three principal axes those are clearly distinguished each other. So our method is especially efficient for models belong to such classes.

Table 3 shows evaluation results according to silhouette image resolution varying. As you see this table, at most $16 \times 16$ resolution is enough. Of course, search time is also better than the cases of high resolutions. All cases the same PC (Pentium II 450MHz CPU, 256MB memory) is used. Consequently $16 \times 16$ resolution is the best for our method.

4. Concluding remarks
This paper proposed a 3D model database system based on the silhouette image matching and presented experimental results to clarify the effectiveness of proposed method. Especially this paper described that it is useful to enter hand sketch images as the query for retrieval of required 3D models. To create hand sketch images is very intuitive manner so that using this prototype system, the user can enter the query easily and rapidly. This is one feature of our proposed system.

As future works, our 3D model matching is based on the 2D image matching. Already many researches on the 2D image matching have been done [7]. Using such more efficient algorithms, we will make our 3D model database system able to retrieve 3D models more accurately.

Acknowledgement
We wish to thank Mr. Fumiki Nakano for his help to make 3D model database.

References: