

3D MODEL SEARCH BASED ON STOCHASTIC ARG MATCHING

N. Nakamura¹, Y. Okada^{1,2}, K. Nijjima¹

¹Graduate School of Information Science and Electrical Engineering, Kyushu University

6-1, Kasuga-koen, Kasuga, Fukuoka, 816-8580, JAPAN

E-mail : {n-naka, okada, nijjima}@i.kyushu-u.ac.jp

²Intelligent Cooperation and Control, PRESTO, JST

ABSTRACT

Due to the high performance of recent computer graphics hardware, 3D CG and CG animations have become in great demand for various applications. Many 3D models have already been created and stored. We need any 3D model search system that allows us to retrieve our required 3D models accurately.

In this paper, the authors propose a 3D model search system that uses the attributed relational graph (ARG) of each 3D model, and that employs the stochastic ARG matching to measure the similarity among them. The proposed system gives better search results than those of D2 method in their experiments. This paper also describes what kinds of features of a 3D model are used as each vertex attributes for the stochastic ARG matching.

1. INTRODUCTION

Recently, a large number of 3D models can be seen in various applications, e.g., video games, CG movies, etc. Its reason is that 3D models are easily handled on a standard PC due to the high performance of recent computer graphics hardware. So many 3D models have already been created and stored. We need any tool that helps us to retrieve our required 3D models efficiently from such a 3D model pool.

In this paper, we propose a 3D model search system using stochastic ARG matching [5]. Our system has good experimental results in comparison with other methods or systems. We describe what kinds of attributes of a 3D model are used for each vertex of ARG in the stochastic ARG matching, and show its experimental results.

As related works, there are several researches on the 3D model search. Paquet and Rioux proposed a 3D model database system that employs many popular 3D model matching algorithms [4]. Osada, et al proposed new 3D model matching algorithm using distribution of distances between any two random points on the surface of a 3D model [3]. This method called D2 has good evaluation results reported in their paper. Our 3D model search system indicates better evaluation results than those of D2 method

in our experiments. ARG matching is strongly related to the topology information of 3D models. Hiraga, et al proposed topology matching technique of 3D models using Multi-resolutional Reeb Graphs (MRGs) [1]. Kaku, et al proposed a similarity measure between 3D models using their OBBTrees [2]. However, we have never seen any researches on the 3D model search using stochastic ARG matching.

The rest of this paper is organized as follows: In Sec. 2, we explain stochastic ARG matching method. Sec. 3 describes our 3D model search system using ARG matching. Sec. 4 is experiments and discussion about their results, and Sec. 5 concludes this paper.

2. STOCHASTIC ARG MATCHING METHOD

In this paper, we employ the stochastic ARG matching method [5] to measure the similarity between 3D models. We start with the definition of ARG.

Definition. An ARG is a triple $G = (V, E, A)$, where V is the vertex set, E is the edge set, and A is the attribute set that contains a unary attribute a_i attaching to each vertex $n_i \in V$ and a binary attribute a_{ij} attaching to each edge $e_{ij} = (n_i, n_j) \in E$.

To define the similarity measure between two graphs G^s and G^t , we introduce some notations. Let H_p denote a binary random variable corresponding to two hypotheses: The hypothesis $H_p = 1$ means that G^t is similar to G^s , and the hypothesis $H_p = 0$ implies that G^t is not similar to G^s . The unary features of G^s and G^t are denoted by Y_i^s , $1 < i < N$, and Y_k^t , $1 < k < M$, respectively, and the binary ones of G^s and G^t by Y_{ij}^s , $1 < i, j < N$ and Y_{kl}^t , $1 < k, l < M$ respectively. We define two vectors:

$Y^s = (Y_1^s, \dots, Y_N^s, Y_{11}^s, Y_{12}^s, \dots, Y_{NN}^s)$ and
 $Y^t = (Y_1^t, \dots, Y_M^t, Y_{11}^t, Y_{12}^t, \dots, Y_{MM}^t)$. Let
 $p(Y^t | Y^s, H_p = h)$ be the probability transforming G^s
into G^t . The similarity measure of G^s and G^t is defined
by

$$S(G^s, G^t) = \frac{p(Y^t | Y^s, H_p = 1)}{p(Y^t | Y^s, H_p = 0)}.$$

To calculate the probability $p(Y^t | Y^s, H_p = h)$, we need two transformation processes: vertex copy process (VCP) and attribute transformation process (ATP).

In VCP, a copy \tilde{G}^t of G^t is made. The vertices of G^s are mapped into the vertices of \tilde{G}^t only when $H_p = 1$. We denote this mapping by X , referred to as correspondence matrix, whose elements are 0 or 1. Let x_{ik} be an element of X . $x_{ik} = 1$ means that the i -th vertex of G^s is mapped into the k -th vertex of \tilde{G}^t . To get a one-to-one correspondence between G^s and \tilde{G}^t , the constraints $\sum_i x_{ik} \leq 1, \sum_k x_{ik} \leq 1$ are required:

ATP is a process that changes the attributes on the vertices of \tilde{G}^t . An example of VCP, ATP and X is shown in Fig. 1.

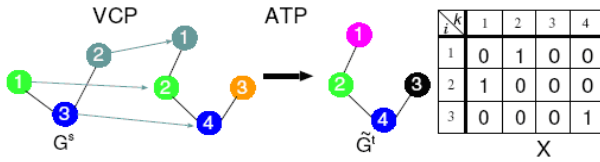


Fig. 1: ARG Matching Process

$p(Y^t | Y^s, H_p = h)$ can be decomposed as

$$p(Y^t | Y^s, H_p = h) = \sum_{X \in \chi} p(Y^t | Y^s, X, H_p = h) \times p(X | Y^s, H_p = h).$$

Here χ denotes a set of the matrix X , $p(X | Y^s, H_p = h)$ denotes a probability of VCP, and $p(Y^t | Y^s, X, H_p = h)$ denotes a probability of ATP.

We assume that X is statistically independent of Y^s for the given H_p . Then we have

$$p(X | Y^s, H_p = h) = \frac{1}{Z(h)} \prod_{(i,k)} \phi_h(x_{ik}) \times \prod_{(i,k),(j,l)} \psi_{ik,jl}(x_{ik}, x_{jl}),$$

with the one-vertex potential function

$$\phi(x_{ik}) = \begin{cases} 0, & x_{ik} = 0, \\ q_0, & x_{ik} = 1, h = 0, \\ q_1, & x_{ik} = 1, h = 1 \end{cases}$$

and the two-vertex potential function

$$\psi_{ik,jl}(x_{ik}, x_{jl}) = \begin{cases} 0, & i = j \text{ or } k = l, x_{ik} = x_{jl} = 1, \\ 1, & \text{otherwise.} \end{cases}$$

Here the partition function $Z(h)$ has the form

$$Z(h) = \sum_{i=1}^N \binom{N}{i} \binom{M}{i} i! q_h^i, \text{ where } q_0 \text{ and } q_1 \text{ are}$$

parameters to control the probability, and are learned from training data. $p(Y^t | Y^s, X, H_p = h)$ can be written as

$$p(Y^t | Y^s, X, H_p = h) = \prod_{(i,k)} p(y_k^t | y_i^s, x_{ik}) \times \prod_{(i,k),(j,l)} p(y_{kl}^t | y_{ij}^s, x_{ik}, x_{jl}).$$

We also assume that the probability of attributes transforming into other attributes is the normal distribution $N(y, \Sigma)$, i.e.,

$$p(y_k^t | y_i^s, x_{ik} = 0) = N(y_i^s, \Sigma_0),$$

$$p(y_k^t | y_i^s, x_{ik} = 1) = N(y_i^s, \Sigma_1),$$

$$p(y_{kl}^t | y_{ij}^s, x_{ik} \cap x_{jl} = 0) = N(y_{ij}^s, \Sigma_{00}),$$

$$p(y_{kl}^t | y_{ij}^s, x_{ik} = x_{jl} = 1) = N(y_{ij}^s, \Sigma_{11}),$$

where the covariance matrices Σ_0 , Σ_1 , Σ_{00} , and Σ_{11} are learned from training data.

3. 3D MODEL SEARCH SYSTEM

We construct a graph from each 3D model by considering components of the 3D model as its vertices and parent-child relationships between the two components as its edges. All components of a 3D model have features, e.g., surface area, volume and so on. Using these features as vertex attributes of the graph of a 3D model, we can obtain ARG of the 3D model. The upper part of Fig. 2 shows one composite 3D model and the lower part of Fig. 2 shows its ARG.

We employ four features as unary attributes, and two features as binary attributes. Unary features include surface area, position, flatness and sphericity. Position means a distance between the center of the 3D model and the center

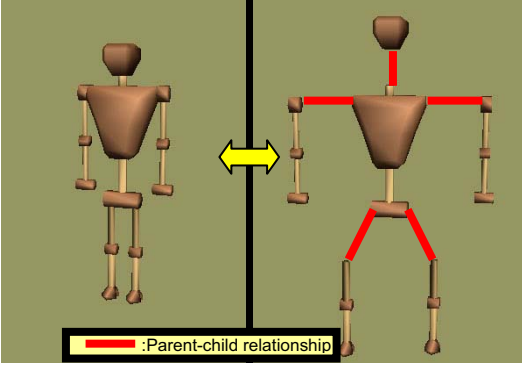


Fig. 2: ARG example of a 3D model

of its component corresponding to the vertex. Flatness F is defined as follows:

$$F = \frac{a + b}{2c},$$

where a and b are the lengths of the first and second longest edges of the bounding box of the component respectively, and c is that of the shortest edge. F is 1 when the shape of a component is like a cube. F becomes bigger when the shape becomes flatter like a board or longer like a pen. Sphericity S is defined as follows:

$$S = 36\pi \frac{V^2}{A^3},$$

where V and A are the volume and the surface area of the component respectively. S is 1 when the shape of a component is perfect sphere. S is regarded as complexity of a component shape. A simpler shape like a sphere has a smaller value and a more complex shape has a greater value. Binary features are the pairs of flatness of two components which have a parent-child relationship between them.

For 3D model search, we construct ARGs from 3D models stored in our database in advance. When entering a 3D model as a query, its ARG is constructed and our 3D model search system calculates the similarity between the query and each of stored 3D models with stochastic ARG matching method. And then the system outputs 3D models those have the similar topology and component shapes as those of the query.

4. EXPERIMENTS

Using the proposed 3D model search system, we performed experiments of the 3D model search. In advance, we made a 3D model database that contains 149 models classified into 21 classes, shown in Fig. 3. Parent-child relationships of these 3D models are defined manually.

We exploit 100 attributes pairs obtained from the database for learning each covariance Σ . To reduce the computational cost, the matching of a query and a 3D model in the database is not carried out when the difference of the vertices numbers of the ARGs exceeds a given threshold.






















1  airplane: 10	2  ball: 7	3  bed: 6	4  car: 10	5  chair: 7	6  chess: 11
7  cupboard: 6	8  helicopter: 6	9  lamp: 7	10  light: 7	11  missile: 6	12  monitor: 5
13  PC: 6	14  robot: 8	15  shelf: 5	16  ship: 6	17  sign: 10	18  sofa: 7
19  sword: 6	20  table: 7	21  toilet: 6		Class Number	
				3D Model	
				Class Name:	
				No. of Models	

Fig. 3: Classes of 3D Model Database

We used three evaluation measures “First tier”, “Second tier” and “Top match” described in [3]. They are defined as follows:

First tier: This criterion represents the percentage of top $(k - 1)$ matches (excluding the query) from the query’s class, where k is the number of 3D models of the class.

$$\text{First tier} = \frac{\text{Top}(k - 1)\text{matches}}{k - 1}$$

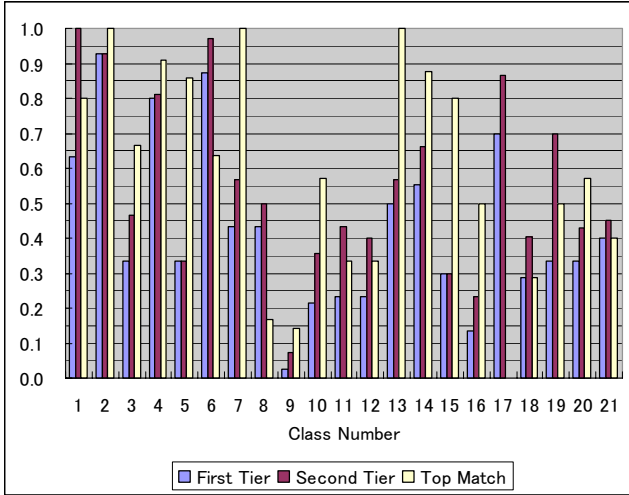
Second tier: This criterion is the same type of “First tier”, but for the top $2(k - 1)$ matches.

$$\text{Second tier} = \frac{\text{Top}2(k - 1)\text{matches}}{k - 1}$$

Top match: This criterion means the percentage of test in which the top match was from the query’s class.

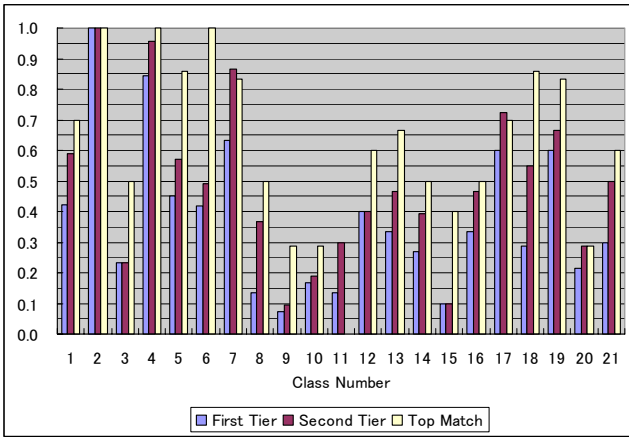
Experimental results are shown in Fig. 4, Fig. 5 and Table 1. In the two figures, search time means the total calculation time to obtain all the results shown in the figures. Fig. 4 shows results of the proposed method, and Fig. 5 shows results of D2. Table 1 also shows averages of the all results of each method. D2 uses a histogram of distances between any two random points on a 3D model surface, so D2 has good results for models whose shape is simple and typical.

However, D2 is poor for searching complicated models. In our proposed system, the composition structure of each model is important factor for the similarity measure. Due to the reason, results of 3D model classes which have complicated and typical composition structure, e.g., airplane, robot, are good. As for both “First tier” and “Second tier”, the averages of our system’s results are better than those of D2.



Search time : 612 sec.

Fig 4: Results of our 3D model search system



Search time : 30 sec.

Fig 5: Results of D2

	First Tier	Second Tier	Top Match
Our System	0.43	0.55	0.59
D2	0.38	0.49	0.61

Table 1: Averages of all results of each method

Our system needs huge time to search a 3D model because of huge computational cost of the ARG matching. Its main calculation part is for transform probabilities $p(Y^t | Y^s, X, H_p = h)$ and $p(X | Y^s, H_p = h)$ for all $X \in \chi$. The size of χ depending on the numbers of vertices of current treated graphs are comparatively large. Hence, by parallel computing for transform probabilities, the computational time can be drastically reduced. In addition, we can also reduce the computational time by employing the speeding up technique for the stochastic ARG matching mentioned in [5].

5. CONCLUSION

In this paper, we proposed new 3D model search system using the stochastic ARG matching, and evaluated it by comparing the results of it with those of D2 method. As described in Sec. 4, the results of our system are better than those of D2 in terms of averages of “First tier” and “Second tier”.

In particular, results of classes which have complicated and typical structure are good. However, our system needs huge computational time. Hence, as future works, we are supposed to realize a faster system.

REFERENCES

- [1] M. Hiraga, et al, “Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes”, Proc. SIGGRAPH2001, pp.203-212, 2001.
- [2] K. Kaku, Y. Okada and N. Nijima, “Similarity Measure Based on OBBTree for 3D Model Search”, Proc. of International Conference on Computer Graphics, Imaging and Visualization <CGIVo4>, IEEE CS Press, pp. 46 – 51, 2004.
- [3] R. Osada, et al, “Matching 3D Models with Shape Distributions”, Int. Conf. on Shape Modeling & Applications, pp. 154 – 165, 2001.
- [4] E. Paquet, M. Rioux, “Nefertiti: a query by content system for three-dimensional model and image databases management”, *Image and Vision Computing* 17, 157-166, 1999.
- [5] D. Zhang and S. Chang, “Stochastic Attributed Relational Graph Matching for Image Near-Duplicate Detection”, tech. rep., Dept. of EE, The University of Columbia, 2004.