

# 3D Model Retrieval

Dejan V. Vranić<sup>1</sup> and D. Saupe<sup>1</sup>

<sup>1</sup>University of Leipzig, Institut für Informatik

## Abstract

The topic of this paper is content-based retrieval of 3D models that are represented as triangle meshes. An object from some 3D geometry database can traditionally be accessed using attached structural information such as textual annotation. However, there are frequent requirements for a content-based retrieval of various multimedia contents. A content-based 3D model retrieval system has been implemented and this system is presented here. Models are retrieved by querying using another 3D model where the shape description for the query model is automatically created. The proposed feature vector is invariant with respect to translation, rotation and uniform scaling.

**Keywords:** 3D model, retrieval, triangle mesh, feature extraction, feature description, MPEG-7, content-based, similarity-based search.

## 1 Introduction

The volume of audiovisual information available in digital form is rapidly growing. A variety of representation forms for multimedia content is also present. This fact induces the development of systems, which are especially created in order to retrieve a desired kind of audiovisual content. Retrieval can be accomplished either by using structured meta data or by directly searching in libraries of binary objects (e.g. images, 3D models, audio sequences and videos). Current content-based retrieval systems are mainly designed for still image [2]-[5], audio and video [9] libraries, while no techniques for content-based 3D model retrieval have been presented yet. This paper presents an approach to fill this gap.

Multimedia professionals, like graphic designers in commercial, manufacturing, scientific and entertainment areas, often use 3D models of particular objects. The need for more detailed 3D models of buildings, tools, aircrafts, cars, characters, animals and humans can be anticipated.

This paper describes the first implementation of an original 3D model retrieval system that provides querying by a 3D model. The most difficult problem in this system's realization was to generate a feature vector that can be used for describing some 3D content. The

proposed feature vector is describing 3D shape. The forthcoming MPEG-7 standard [6]-[8], officially called "Multimedia Content Description Interface", will standardize tools to describe multimedia content. Nevertheless, the 3D Model Description Scheme (briefly DS) in the context of MPEG-7 is rather underspecified and provides an open area of research.

## 2 3D Model Retrieval System

Searching and indexing objects from a database is traditionally performed by using annotations. Mostly, annotations are created manually and represented by textual information. Unfortunately, textual annotations cannot encode all the information available in a 3D model. Thus, retrieval of 3D models using their content is highly desirable.

We propose the system which is shown in Figure 1.

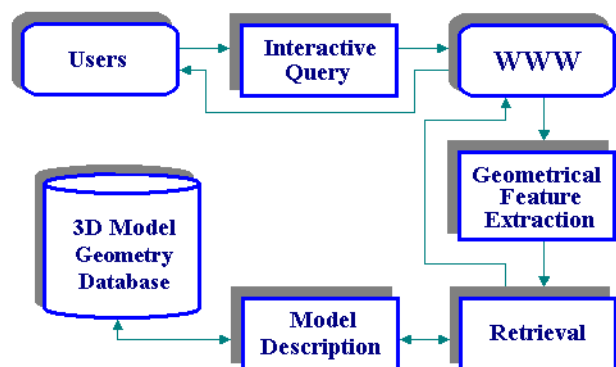


Figure 1. The scheme of proposed system.

This system comprises the following modules, which are undergoing development:

- **Interactive query** - allow users to specify the importance of each type of descriptor.
- **Geometrical feature extraction** - check the necessary conditions (e.g. is it a solid object) for each 3D model and extract appropriate feature vectors.
- **Model description** - attach to each object a description of its features.
- **Retrieval** - search the collection for 3D models matching the user query.

The system's architecture offers a user-interface that enables to non-specialist users an easy and effective access to (complex) 3D geometry database. The first

<sup>1</sup> { vranic | saupe }@informatik.uni-leipzig.de

realization of this system deals only with a single object in the scene. Browsing the database, interactive querying and querying by an existing 3D object are also provided.

Our current application utilizes the following low-level features: number of vertices, number of polygons, surface area, bounding box, examination of closedness, volume of 3D model and volume/surface ratio. The original algorithms are applied in order to approximate the volume of a closed 3D object as well as to examine the closedness of a model and to close a model which is made up of an open polygonal surface. The last part is accomplished by introducing new polygons while the number of vertices is not changed. The only restriction regarding models is that we suppose the objects are well formed, e.g., there is no overlap of polygons, disconnected vertex, polygon inside a solid object, etc.

### 3 Content-Based Retrieval

The main task is to create a set of features that will enable an efficient description of a 3D model. The feature extraction is performed mostly automatically. Several features can be optionally defined by manual annotations. Moreover, some features can not be extracted automatically (e.g. a date when a model was created or modified, the name of the author) and comments can be of use, too.

Some simple low-level features (e.g. number of vertices and polygons, surface area, bounding box, closedness, volume of 3D model, etc.) are not good enough for effective retrieval. These features might be of interest only in some special queries related to the complexity of a model. More useful features will be derived from 3D shape of the objects, in particular when they are invariant with respect to rotation, translation and scaling.

#### 3.1 The Shape Descriptor

A feature vector, which effectively captures the shape of a 3D model, is difficult to generate. Our goal is to develop such a vector which possesses the characteristics mentioned in the previous subsection. We have incorporated a modification of the Principal Component Analysis (briefly PCA, also known as discrete Karhunen Loeve transform or Hotteling transform) [1] in the geometrical feature extraction module. This transformation changes the coordinate system axes to new ones which coincide with the directions of the three largest spreads of the point (i.e. vertex) distribution. A 3D object represented as a triangle mesh consists of geometry, topology and attributes. Geometry is determined by the vertex coordinates, information how vertices are connected in order to form triangles is called topology and attributes are color, texture, etc. In our system, attributes are still not under considerations because the stress is on representing spatial relations within a 3D model i.e. geometry and topology.

#### 3.1.1 Modification of PCA

The purpose of the principal component analysis applied to the 3D model is to make the resulting shape feature vector independent of translation and rotation as much as possible. The PCA will be based on the collection of vertex vectors. To account for the differing sizes of the corresponding triangles we introduce weighting factors proportional to the corresponding surface area.

Let  $V^{(0)} = \{\vec{v}_1^{(0)}, \vec{v}_2^{(0)}, \dots, \vec{v}_N^{(0)}\}$  be the given set of vertices associated to a triangular mesh. We have:

$$\vec{v}_k^{(i)} \in R^3; k = 1, \dots, N; i = 0, 1; \quad (1)$$

where  $(i)$  denotes an index as explained further below.

Let the "mean vertex"  $\vec{m}^{(0)}$  be defined by:

$$\vec{m}^{(0)} = \frac{1}{N} \sum_{k=1}^N w_k \vec{v}_k^{(0)}, \quad w_k = \frac{N \cdot S_k}{3S}, k = 1, \dots, N; \quad (2)$$

where  $w_k$  is the weight associated to the vertex  $\vec{v}_k^{(0)}$ ,  $S$  is the surface area of the mesh (i.e. the sum of the areas of all triangles in the mesh) and  $S_k$  is the sum of surfaces of all triangles that have  $\vec{v}_k^{(0)}$  as a vertex. It is obvious that:

$$\sum_{k=1}^N w_k = N. \quad (3)$$

These weights introduce additional stability in the feature vector extraction. For instance, if we add a new vertex  $\vec{v}_{N+1}^{(0)}$  to  $V^{(0)}$ , which belongs to the area of the triangle made up of vertices  $\vec{v}_a^{(0)}$ ,  $\vec{v}_b^{(0)}$  and  $\vec{v}_c^{(0)}$  ( $a, b, c \in \{1, \dots, N\}$ ), and form the set  $V^* = V^{(0)} \cup \vec{v}_{N+1}^{(0)}$  with the mean vertex  $\vec{m}^*$ , then it can easily be shown that  $\vec{m}^{(0)} = \vec{m}^*$ .

Applying the known procedure of PCA, the covariance matrix  $C^{(0)}$  (in this case a 3x3 matrix) is determined by:

$$C^{(0)} = \frac{1}{\sum w_k} \sum_{k=1}^N w_k (\vec{v}_k^{(0)} - \vec{m}^{(0)}) \cdot (\vec{v}_k^{(0)} - \vec{m}^{(0)})^T. \quad (5)$$

$C^{(0)}$  is a symmetric, real matrix. The next step is finding the eigenvalues of  $C^{(0)}$  and sorting them in decreasing order. After forming the transformation matrix  $A^{(0)}$ , that has the normalized eigenvectors as rows with non-negative elements on the main diagonal, we transform the original vertices to the new ones:

$$\vec{v}_k^{(1)} = A^{(0)} (\vec{v}_k^{(0)} - \vec{m}^{(0)}), k = 1, \dots, N. \quad (6)$$

If matrix  $C^{(0)}$  was non-diagonal then data were correlated. This is usually the case with an arbitrary triangle mesh. After the PCA transformation (6) the data is in a canonical orientation which is invariant to translation and rotation.

#### 3.1.2 Example of PCA

The effect of PCA is depicted on figure 2. The "Stanford Bunny" model (Source: Stanford University Computer Graphics Laboratory) is used for this purpose and its appearance is shown on the left side of figure 2.

Originally, the triangle mesh was opened and our algorithm for closing a model has been used. This model contains  $N=1494$  vertices and 2984 triangles. The axes of the original coordinate system are denoted with  $x, y, z$ , while the principal components are marked with  $P_1, P_2$ , and  $P_3$ .

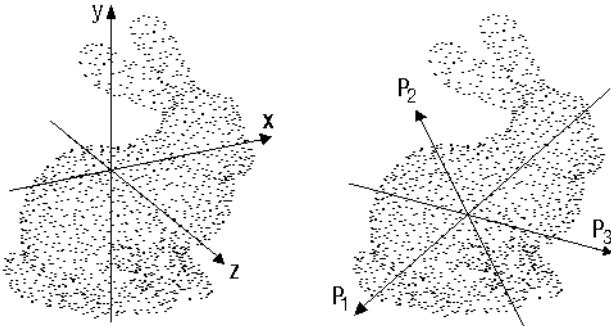


Figure 2. Principal Component Analysis.

### 3.1.3 Feature Extraction

A first simple feature vector can be calculated from the transformed model. Suppose we have a given set of  $l$  directional vectors  $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_l\}$ . Then we intersect the triangle mesh with the ray emanating from the origin of the PCA coordinate system and traveling in the direction  $\vec{u}_i$  ( $i \in \{1, \dots, l\}$ ). The distance to the farthest intersection is taken as the  $i$ -th component of the feature vector which is scaled to Euclidean unit length to ensure scale invariance.

In our current application,  $l=20$  (figure 3). The vertices of a dodecahedron, with the center in the coordinate origin, are taken as directions. This feature is invariant with respect to rotation and translation because of the fact that initial coordinate axes are transformed. The scaling invariance is accomplished by normalizing the feature vector, as mentioned above.

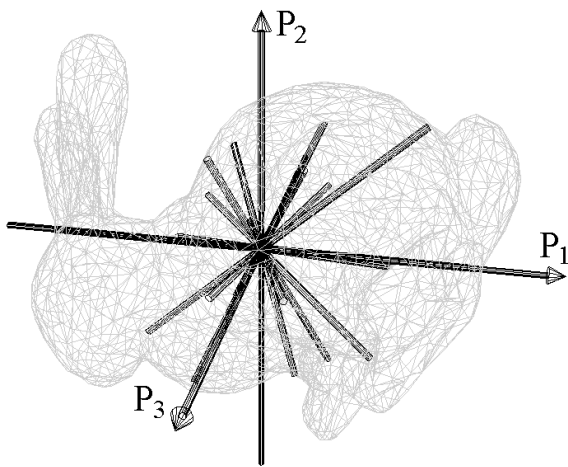


Figure 3. Extraction of Shape Descriptor.

### 3.1.4 Feature Description

After extraction of features the next step is their formal description. As mentioned above, the forthcoming MPEG-7 standard will provide a rich set of standardized mechanisms and means aimed at describing multimedia content. The MPEG-7 terminology has been adopted and mutual relation between a descriptor and a feature is explained in the following definition: "A descriptor is a representation of a feature. A descriptor defines the syntax and the semantics of the feature representation" [6], [7].

Therefore, the descriptor of our current feature vector is determined with 20 non-negative real numbers, where the  $i^{\text{th}}$  component is the object extension in direction of the  $i^{\text{th}}$  vertex of mentioned dodecahedron, which is defined (the vertex coordinates and the numbering) internally. This defines semantics of the descriptor. The syntax will be defined by DS for real vectors. This scheme is still to be determined. The current status is:

```

<!-- Definition of "Vector of reals" -->
<DType name="VectorR">
  <attribute name="Size" datatype="integer"/>
  <seq minOccurs="Size" maxOccurs="Size">
    <!-- Inner definition issue has still to be resolved -->
    <!--<DType name="ValueVectorR" datatype="float"/>-->
  </seq>
</DType>

```

However, MPEG-7 will not be a restrictive system for audio-visual content description. It will be a flexible and extensible scope for describing multimedia data with developed set of methods and tools. The following tools will be standardized [6]:

- A set of **descriptors** (briefly D),
- A set of **description schemes** (DS),
- **Description Definition Language** (DDL) - to specify description schemes (and perhaps descriptors), and
- At least one way to create **coded descriptions**.

Since this standard is still developing and D/DS for 3D models have not been specified yet, we can make a contribution in this area. The 3D Model DS should support "the hierarchical representation of different descriptors in order that queries may be processed more efficiently in successive levels (where  $N$  level descriptors complement  $(N-1)$  level descriptors)" [7]. Hence, different features on different levels of detail will be considered. We have recently been encouraged by the reflector of the MPEG-7 DS group to implement our own DS for 3D models. This DS should comply with MPEG-7 specification [6], [7].

## 4 Retrieval Example

An example of presented 3D model retrieval technique is shown in this section. Note that the operating version of the system is still not finished and this demonstration is created by separate modules. The starting screen is depicted in figure 4. The possibility to upload a query

model will be provided or the internal 3D geometry database can be browsed in order to choose a query model. Retrieval can be performed using some low-level features (e.g. the range of the number of polygons or the number of vertices can be defined in order to specify level of details or complexity) and/or added annotations in a query (e.g. kind of a model).

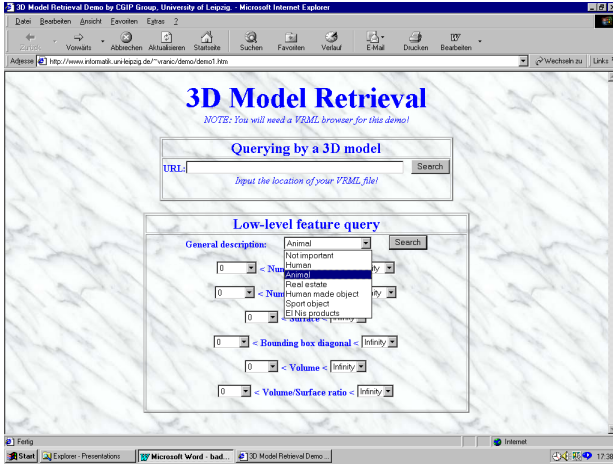


Figure 4. Selecting a query model.

When the query object is loaded, the next step is feature extraction. Fine-tuning for better search results is also provided. This part is displayed in figure 5. Calculating the presented feature vector is followed by performing the retrieval algorithm.

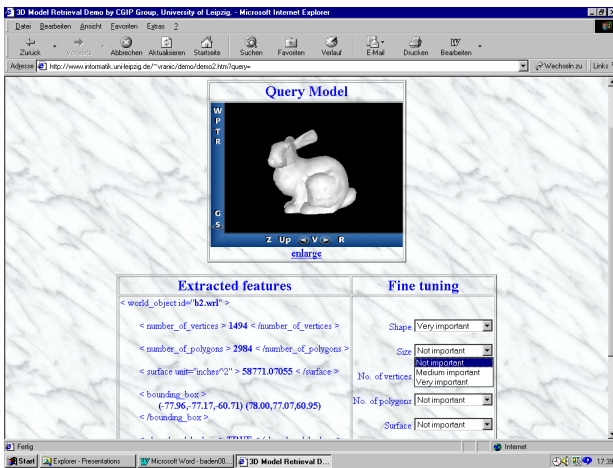


Figure 5. Fine-tuning for optimal search results.

Finally, figure 6 represents the screen with retrieved models. Our system supports descriptions allowing a ranking of the content by the degree of similarity with the query. Various types of similarity may be considered. For example, if a shape descriptor is used, a query using a bunny may not only retrieve bunnies but also other objects with a similar shape. This is the case in our example.

The best match in figure 6 is the model that has been obtained by rotating the set  $V^{(0)}$  of the bunny model around  $x$ -axes for the angle of  $\alpha = \pi/3$ , translating for

the vector  $\vec{t} = (10, 20, -30)$  and scaling with factor  $\sqrt{5}$ . The result verifies the affine invariance of our feature vector. The second and the third match models have also been derived from the bunny model. The match 2 has been obtained by adding some noise to  $V^{(0)}$  (i.e. random displacement of vertices), rotating around  $y$ -axes for  $\alpha$ , translating for  $\vec{t}$  and scaling with factor  $\sqrt{3}$ . The match 3 encompasses more noise and it has been rotated around  $y$ -axes for  $\alpha$ , translated for  $\vec{t}$  and scaled with  $\sqrt{7}/2$ .

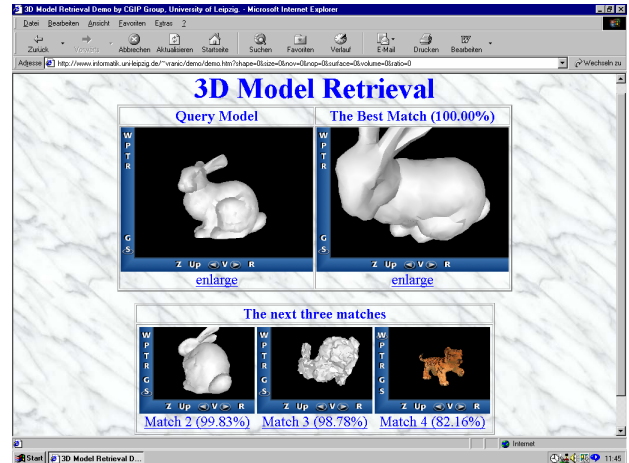


Figure 6. Retrieved models.

The distance between two vectors  $\vec{f}$  and  $\vec{g}$ , which describe shape, has been normalized in the following manner:

$$dist(\vec{f}, \vec{g}) = 100 \left( 1 - \frac{\sum_{i=1}^{20} |f_i - g_i|}{2\sqrt{10}} \right).$$

We recall that:

$$\begin{aligned} \vec{f} &= (f_1, \dots, f_{20}), \quad \vec{g} = (g_1, \dots, g_{20}), \\ f_i, g_i &\geq 0, i = 1, \dots, 20; \quad \|\vec{f}\| = \|\vec{g}\| = 1, \\ \Rightarrow \max_{\vec{f}, \vec{g} \in R^{20}} \left( \sum_{i=1}^{20} |f_i - g_i| \right) &= 2\sqrt{10}. \end{aligned}$$

The algorithm for measuring distance between two feature vectors will be explored in a forthcoming paper.

## 5 Further Improvements

After the testing phase the following task will be to incorporate some mesh-simplification techniques such as in [13] in the system in order to speed up the feature extraction algorithm. Since complexity of models is proportional to level of details, triangle meshes created by 3D scanners usually contain more than 50,000 vertices and even more faces (i.e. triangles). Having in mind that, in our application, the feature vector is made up of (only) 20 real components, we expect that the execution time will be shorter if mesh-simplification is performed before our algorithm. Naturally, this stands only for models with "a lot of triangles". For simpler models there is no need to perform simplification first.

The threshold regarding the number of vertices and the number of triangles (i.e. complexity), in order to decide whether to simplify a mesh or not, will be determined. According to the definition of our shape descriptor, if we use the same model with several levels of details the feature vectors for each level will be approximately the same.

We are planning to realize some other feature vectors that will also depict spatial relations inside a 3D object. For instance, volume distribution for a solid model and moment distribution can be derived in a similar way as it is described above. We will try to optimize the dimension in each kind of these vectors. In other words, we will not use only directions of dodecahedron's vertices. We consider equidistant points of a cube with center in the coordinate origin as a substitution for the dodecahedron. In this case, the dimension of feature vector can easily be changed. Further measurement and evaluation of these features will be done.

Recognition of features on 3D surfaces (e.g. smoothness, roughness and curvature) might also be of interest for some applications. We will explore the usefulness of illumination of an object in order to obtain the information about the distribution of light (local variants).

The retrieval algorithm for 3D model geometry database search will be investigated further. It should support combination of different features, and application of a weight function  $w_i$  for each particular feature  $i$  in a feature vector [9]. Hence, users with more skills would be able to use fine-tuning for optimum search results, e.g. adjust particular weight between the different descriptors available.

The size of our geometry database is continuously increasing. Models are collected from the Internet or they are created or modified by programming. Controlled modifications of existing objects [12] are particularly of interest for our application because of the fact that we perform similarity-based retrieval.

All models are stored as VRML 2.0 [10] files in the same directory. We will also consider some other 3D file formats (e.g. DXF, 3DS). In the future, we will realize a distributed 3D model geometry database [11], i.e., the collection of objects will not be at one location, and the search will be performed net-wide.

## 6 Discussion and conclusion

A novel technique for content-based 3D model retrieval is presented. The used feature vector aimed at describing shape is defined. Querying by some low-level features is enabled but the spatial organization of these features has also been considered.

The development of 3D model DS will be aimed at designing descriptors that allow a fast, hierarchical search procedure. A search engine will be developed on the basis of this DS, with similarity-based retrieval from a 3D model geometry database.

## Acknowledgements

This work was supported by an award from the Deutsche Forschungsgemeinschaft (DFG), grant GRK 446/1-98 for the Graduiertenkolleg Wissensrepräsentation (graduate study program on knowledge representation) at the University of Leipzig.

## References

- [1] M. Petrou, P. Bosdogianni, *Image Processing: The Fundamentals*, John Wiley, 1999.
- [2] J. Malik, C. Carson, S. Belongie, *Region-Based Image Retrieval*. Proceedings DAGM'99, Mustererkennung, Springer Verlag (1999), pp. 152-154.
- [3] University of California, Berkeley, Digital Library Project, Image Retrieval by Image Content, <http://galaxy.cs.berkeley.edu/photos/blobworld/>
- [4] S. Ravela, R. Manmahta, *On computing global similarity in images*. Proceedings of IEEE Workshop on Applications of Computer Vision (WACV98), Princeton (1998), pp. 82-87.
- [5] University of Massachusetts, Center for Intelligent Information Retrieval, Image Retrieval Demo, <http://cowarie.cs.umass.edu/~demo/Demo.html>
- [6] MPEG Requirements Group, *Overview of the MPEG-7 Standard*. Doc. ISO/MPEG N3158, Maui, Hawaii, December 1999.
- [7] MPEG Requirements Group, *MPEG-7 Requirements Document V.10*. Doc. ISO/MPEG N2996, Melbourne, October 1999.
- [8] MPEG DDL Group, *MPEG-7 Description Definition Language Document V2*. Doc. ISO/MPEG N2997, Melbourne, October 1999.
- [9] J.-R. Ohm, et al., *A multi-feature Description Scheme for image and video database retrieval*. IEEE Multimedia Signal Processing Workshop, Copenhagen, September 1999.
- [10] J. Hartman, J. Wernecke, *The VRML 2.0 Handbook: Building Moving Worlds on the Web*, Addison Wesley, 1996.
- [11] B. MacIntyre, S. Feiner, *A Distributed 3D Graphics Library*. Proceedings of ACM SIGGRAPH 98, Orlando (1998), pp. 361-370.
- [12] M. Teichmann, S. Teller, *Assisted Articulation of Closed Polygonal Models*, MIT Tech. report, 1998.
- [13] A. Ciampalini, P. Cignoni, C. Montani, R. Scopigno, *Multiresolution decimation based on global error*, The Visual Computer, Springer International, 13(5), 1997, pp.228-246.