3D Shape Descriptor Based on 3D Fourier Transform

D. V. Vranić and D. Saupe
Institute of Computer Science
University of Leipzig
P.O. Box 920, D-04009 Leipzig, Germany
E-mail: vranic@informatik.uni-leipzig.de

Keywords: 3D object, triangle mesh, retrieval, feature vector, content-based, voxel.

Abstract - In this paper, we propose a new method for describing 3D-shape in order to perform similarity search for polygonal mesh models. The approach is based on characterization of spatial properties of 3D-objects by suitable feature vectors, i.e., the goal is to define 3D-shape descriptors in such a way that similar objects are represented by “close” points in the feature vector space. We present a descriptor which is invariant with respect to translation, rotation, scaling, and reflection and robust with respect to level-of-detail. A coarse voxelization of a 3D-model is used as the input for the 3D Discrete Fourier Transform (3D DFT), while the absolute values of obtained (complex) coefficients are considered as components of the feature vector. Multiple levels of abstraction of the feature are embedded by the applied transform. The performance of the proposed method is compared to some previous approaches by means of precision/recall tests. Generally, results show that the new approach introduces improvements in the 3D-model retrieval process.

I. INTRODUCTION

The amount of unique information produced in the world is rapidly increasing. The most recent studies (like [7]) suggest that this production exceeds 1 exabyte (i.e., 10^{18} bytes) of new information per year, which is roughly 250 megabytes for every human on earth. Magnetic storage is becoming the universal medium for information storage. At the same time, much data are available on-line to a broad range of users. Therefore, the actual need for efficient data-access has led to the development of different search tools. The role of multimedia is also increasingly important in many real-world applications such as e-commerce, communication or education. Consequently, several multimedia standards (e.g., MPEG-7 [2,3]) define open specifications of various kinds of audiovisual information. The aim of these standards is to provide efficient retrieval and enable interoperability between applications.

The topic of this communication is content-based 3D-object retrieval [1,4,6,8-10]. A 3D model, represented as a triangle (polygonal) mesh, is used as a query. Retrieved models should be ordered by the degree of shape-similarity to the query. Generally, there are three major modules in 3D-model retrieval systems:

- *Pose normalization*. 3D-models have arbitrary scale, orientation, and position in the 3D-space. In order to capture some features, a model has to be placed into a canonical coordinate frame. Thereby, if we scaled, translated, rotated, or flipped a model, then the placing into the canonical frame would be the same. Furthermore, if a model is given in multiple levels-of-detail, canonical representations of different levels should be approximately the same. The normalization step is not necessary when local features are considered (e.g., curvature [3]).
- *Feature extraction*. The feature vectors are aimed at characterizing 3D-shape. Besides the invariance with respect to translation, rotation, scaling, and reflection, basic requirements that definitions of feature vectors should fulfill are robustness with respect to level-of-detail and multiple levels of abstraction (changeable dimension). Usually, the features are stored as vectors with real-valued components and fixed dimensions. There is a trade-off between the required storage, computational complexity, and the resulting retrieval performance.
- *Search in the feature vector space*. All models from an available database are compared to a query object by calculating distance between feature vectors of selected type. In other words, the feature vectors are considered as points in the search space and the best match is the nearest neighbor. The l_1 or l_2 norms are conventionally used to calculate distances in the feature space. However, other metrics (e.g., a modification of Hausdorff distance) can be more suitable in some cases.

We introduce a 3D-shape descriptor based on the 3D DFT which is applied to a voxelized model in the canonical coordinate frame. Our procedure for normalizing the pose of an object [10] is presented. We also give a brief overview of the previous work and compare the new descriptor with the approaches described in [4,6,10].

II. PREVIOUS WORK

The most prominent tool for accomplishing the pose normalization is the Principal Component Analysis (PCA). Conventionally, the PCA [5] is applied only to a set of points (e.g., vertices or centroids of triangles), thus, the differing sizes of triangles cannot be taken into account. In order to account the *differing* sizes of triangles of a mesh Vranić and Saupe [8] introduced weighting factors associated to vertices, while Paquet with co-authors [4] established weights associated to centers of gravity of triangles. Both methods represent improvements comparing to the classical PCA. These "weighted" PCA analyses were designed to approximate the PCA of the whole point set of a model. In the case of the "continuous" PCA presented in [10] (see section III), the calculation of the parameters is slightly more expensive comparing to the classical case, while the accuracy is limited only by the applied arithmetic (e.g., double precision) and we do not have any systematical errors.

The "rotation invariant" 3D-shape descriptor proposed in [6] is invariant with respect to rotations of 90 degrees around the coordinate axes. This restricted rotation invariance is attained by a very coarse shape representation (by clustering point clouds). Since the normalization step is omitted, if an object is rotated around an axis (e.g., by 45 degrees), the feature vector differs significantly. Therefore, in our experiments we add the normalization as the preprocessing step before the feature extraction.

Cords-based, moments-based, and wavelet transform-based descriptions are presented in [4]. In our tests the cords-based feature shows better performance than the moments-based. In the experiments in [4], the cords-based descriptor is used as the most efficient of the proposed descriptors. A cord is defined as a vector that points from the center of mass of a model to the center of mass of a triangle of a mesh. Before determining a cord, the model is normalized. After the calculation of all cords, the feature vector is composed from three histograms: the distribution of the angles between the cords and the first principal axis, the distribution of the angles between the cords and the second principal axis, and the distribution of the cord lengths. The histograms are normalized using the total number of cords. The number of bins of the histograms determines the dimension of the feature vector. The definitions of cords-based and moments-based descriptors, as well as our tests, suggest that these feature vectors are not robust with respect to the level-of-detail of a model.

The forthcoming MPEG-7 standard [2,3] will define tools to describe multimedia content. The MPEG-7 3D-shape descriptor [3] exploits some local attributes of the 3D surface, therefore, the pose normalization is not necessary. The shape index is defined as a function of the two principal curvatures and its value is not defined for planar surfaces. The shape spectrum of the 3D mesh is the histogram of the shape indices calculated over the entire mesh. The estimation of the principal curvatures is the key step of the feature extraction. The curvature estimation involves the following three steps: estimation of the normal vector for each face, local parametric surface fitting around each face, and estimation of the principal curvatures. However, since a 3D-mesh model is assumed to be an orientable surface without multiple edges, isolated faces or vertices, or any other topological singularities, a filtering of the model is highly recommended. Our first tests show that this descriptor is not robust with respect to level-of-detail.

In the recent paper [10], we introduced the application of spherical harmonics to the problem of 3D-object retrieval. Basically, instead of probing the geometry in only a few directions and using these values as components of the feature vector [1] (spatial domain), we improved robustness by sampling a spherical function in many points but characterizing the map by just a few parameters, using spherical harmonics (frequency domain). Our other approaches for characterizing the global 3D shape include an enhancement of the ray-based feature vector [8,1] as well as volume-based, voxel-based, silhouette-based, and depth buffer-based feature vectors [1]. The features possess properties (section I) desirable for retrieval applications. An account of the MPEG-7 description scheme based on our descriptors is given in [9].

Since there is no founded theory what the best way to describe a 3D-shape is, we study fundamentally different types of descriptors and compare their retrieval effectiveness.

III. Canonical Coordinate Frame

We recall our modification of the PCA, so-called "continuous" PCA, which was introduced in [10]. The pose normalization step is needed to insure the invariance requirement for most of the 3D-shape descriptors. By pose normalization we assume finding a canonical position, orientation, and scaling, or briefly a *canonical coordinate frame*.

Let $T = \{T_1,...,T_m\}$ ($T_i \subset \mathbb{R}^3$) be the set of triangles of a mesh, $P = \{\mathbf{p}_1,...,\mathbf{p}_n\}$ ($\mathbf{p}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$) the set of vertices, and $I = \cup_{i=1,..,m} T_i$. Each triangle is considered as a continuous set of interior points, while the point set I is actually the surface of an object. The goal is to find an affine map τ: $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ in such a way that for an arbitrary concatenation σ of translations, rotations, reflections, and scaling the equation $P' = \tau(P) = \tau(\sigma(P))$ is valid.

Let S_i be the surface area of triangle T_i, then the surface area of the whole object is given by $S := S_1+...+ S_m = \iint_I ds$.

- The *translation invariance* is accomplished by finding the center of gravity of a model
$$\mathbf{c} = S^{-1}\iint_I \mathbf{v}ds \quad (\mathbf{v}\in I)$$
and forming the point set $I' = \{\mathbf{u} \mid \mathbf{u} = \mathbf{v}-\mathbf{c}, \mathbf{v}\in I\}$.
- To secure the *rotation invariance* we apply the "continuous" PCA on the set I'. First, we calculate the covariance matrix C (type 3x 3) by
$$C = S^{-1}\iint_{I'} \mathbf{u}\cdot\mathbf{u}^T ds \quad (\mathbf{u}\in I')$$
Matrix C is a symmetric real matrix, therefore, its eigenvalues are positive real numbers. Then, we sort the eigenvalues in the non-increasing order and find the corresponding eigenvectors. The eigenvectors are scaled to the Euclidean unit length and we form the rotation matrix R, which has the scaled eigenvectors as rows. We rotate all the points of I' and form the new point set $I'' = \{\mathbf{w} = (w_x, w_y, w_z) \mid \mathbf{w} = R\cdot\mathbf{u}, \mathbf{u}\in I'\}$
- The *reflection invariance* is obtained using the matrix F = diag(sign(f_x), sign(f_y), sign(f_z)), where
$$f_x = S^{-1}\iint_{I''}\mathrm{sign}(w_x)w_x^2 ds \quad (f_y, f_z \text{ analogously}).$$
- The *scaling invariance* is provided by the scaling factor $s = \sqrt{(s_x^2+s_y^2+s_z^2)/3}$, where s_x, s_y, and s_z represent average distances of points $\mathbf{w} \in I''$ from the origin along **x**, **y**, and **z** axes, respectively. These distances are calculated by
$$s_x = S^{-1}\iint_{I''}|w_x|ds \quad (s_y, s_z \text{ analogously}).$$

Finally, the affine map τ is defined by
$$\tau(\mathbf{p}) = s^{-1}\cdot F\cdot R\cdot(\mathbf{p}-\mathbf{c}).$$
The canonical coordinates are obtained by applying τ to the initial point set I. In practice, we transform only the set of vertices P into the canonical coordinates P', because the topology remains the same. Numerous examples confirm that the "continuous" analysis performs better than the "weighted" PCAs. We applied the described method for normalizing a 3D-model before extracting features presented in [1,4,6,8], thereby the retrieval performance of the obtained descriptors is improved.

IV. Feature Vector Based on 3D Discrete Fourier Transform

After finding the canonical position and orientation of a model, the next step is feature extraction. The definition of the new feature vector is based on a modification of the idea presented in [1], voxel-based feature. The extraction is performed in two steps:

1. Voxelization using the bounding cube and
2. Application of the 3D Discrete Fourier Transform.

The bounding cube (BC) of a 3D-model is defined to be the tightest cube in the canonical coordinate frame that encloses the model, with the center in the origin and the edges parallel to the coordinate axes. After determining the BC, we perform voxelization in the following manner: we subdivide the BC into N^3 (N is a power of 2) equal sized cubes and calculate the proportion of the total surface area of the mesh inside each of the new cubes (cells). We regard the cell with the attributed value as the *voxel* at the given position. Obviously, of all voxels inside BC the fraction having values greater than zero decreases with increasing N. Therefore, a suitable way of storing a voxel-based feature vector is an octree structure. Thus, we have an efficient hierarchical feature representation.

The information contained in this octree can be used in several ways. In [1], we used a similar voxelization as a feature in the spatial domain with a reasonably small N. The feature vector had N^3 components and the l_1 or l_2 norms were engaged for calculating distances. The proposed modification is the following: we choose a greater value of N and represent the feature in the frequency domain by applying the 3D Discrete Fourier Transform (DFT) to the voxelized model (i.e., calculated values in the N^3 cells).

Let $Q = \{ q_{ikl} \mid q_{ikl} \in \mathbb{R}, -N/2 \leq i, k, l < N/2 \}$ be the set of all voxels. We transform the set Q into the set $G = \{ g_{uvw} \mid g_{uvw} \in \mathbb{C}, -N/2 \leq u, v, w < N/2 \}$ by

$$g_{uvw} = \frac{1}{\sqrt{N^3}} \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} q_{ikl} \exp\left(-j \frac{2\pi}{N} (iu + kv + lw) \right).$$

Finally, we find the absolute values of the coefficients g_{uvw} with indices $-K \leq u, v, w \leq K$ (the lowest frequencies). Except the coefficient g_{000}, all selected complex numbers are pairwise conjugate. Therefore, the feature vector consists of $((2K+1)^3+1)/2$ real-valued components. In our experiments, we select K = 1, 2, 3, i.e., the descriptors possess 14, 63, and 172 components, respectively.

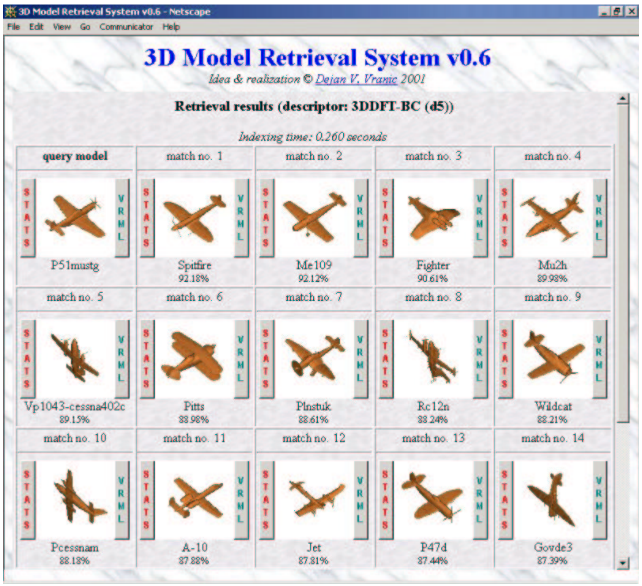
The value of parameter N (the resolution of voxelization) should be sufficiently large in order to capture spatial properties of a model by the 3D DFT. In practice, we select $N = 128$ and on average about 20000 voxels (out of 128^3 elements of the set Q) have values greater than zero. This makes the octree representation very efficient. During the 3D DFT, we compute only those elements of the set G that are used in the feature vector (14, 63, or 172 out of 128^3).

The proposed descriptor shows better retrieval performance than the voxel-based feature presented in [1]. Having in mind that the ray-based descriptor [8,1] was improved by incorporating spherical harmonics [10], we infer that if the l_1 or l_2 norms are engaged, representation of a feature in the frequency domain is more efficient than representation of the same feature in the spatial domain.

V. Experimental Results

The 3D model database used for experiments contains 1830 3D-models (mostly collected from *www.3dcafe.com* and *www.viewpoint.com*) in different 3D file formats (VRML, DXF, 3DS, OFF, etc.). On average a model contains 5682 vertices and 10356 triangles. We manually classified models by shape. For example, we have 33 models of cars, 63 airplanes, etc. We use this classification in our precision/recall test. Briefly, *recall* is the proportion of the relevant models actually retrieved and *precision* is proportion of retrieved models that is relevant. By examining the precision/recall diagrams for different queries (and classes) we obtain a measure of the retrieval performance for a selected descriptor and matching criterion.

A retrieval example is shown in Fig. 1. A model of an airplane is used as the query, while the l_1 norm was applied to the proposed feature vector of dimension 63. The models are visualized from the same direction in the original coordinate frame (before the pose normalization). The query model is displayed in the upper-left corner and the first 14 matches are airplanes. It is questionable if all the matches are relevant to the query. Our manual classification is used only to determine the relevance, i.e., it is used as the ground truth for the precision/recall test. According to the classification, the match number 6 (a biplane) is considered to be non-relevant.


3D Model Retrieval System v0.6 - Netscape
File Edit View Go Communicator Help
STATS
VRML

3D Model Retrieval System v0.6

Idea & realization © Dejan V. Vranic 2001

Retrieval results (descriptor: 3DDFT-BC (d5))

Indexing time: 0.260 seconds

query model	match no. 1	match no. 2	match no. 3	match no. 4
P51mustg	Spitfire 92.18%	Me109 92.12%	Fighter 90.61%	Mu2h 89.98%
match no. 5	match no. 6	match no. 7	match no. 8	match no. 9
Vp1043-cessna402c 89.15%	Pitts 88.98%	Plnstuk 88.61%	Rc12n 88.24%	Wildcat 88.21%
match no. 10	match no. 11	match no. 12	match no. 13	match no. 14
Pcessnam 88.18%	A-10 87.88%	Jet 87.81%	P47d 87.44%	Govde3 87.39%

Fig. 1. Query for an airplane.

In a series of experiments we compared the proposed descriptor with the descriptors presented in [4,6,10] (see section II). We tested the retrieval performance on the categories of airplanes and cars (limousines). First, we determined the optimal dimensions of the feature vectors. The 3D DFT based feature is the most efficient if the vector dimension is 172, while the best choices for vector dimensions in the cases of the cords-based, the "rotation invariant", and the ray-based with spherical harmonic representation feature are 120, 66, and 66, respectively. Afterwards, we calculated the average precision/recall diagram for all models belonging to the selected category and for all four descriptors, using the l_1 norm for the retrieval. The results of the tests are given in Fig. 2. In the case of the models of cars, all descriptors are reasonably effective. The mean values of the average precision/recall curves are given in brackets. These values can also be used in the comparison. We observe that the presented 3D DFT based descriptor shows the best overall performance. However, the behavior of the ray-based feature in the frequency domain is the best if we consider only recall values between 0 and 50 %. The category of airplane models is, generally, more difficult for retrieval applications. In this case, the performances of the cords-based and the "rotation invariant" descriptor drop significantly. The overall performances of the frequency domain descriptors are also weaker, but the precision is still good for the small recall values. We stress that the pose normalization based on the "continuous" analysis (section III) is used to modify and improve the cords-based and the "rotation invariant" descriptors.

These results are obtained on a PC with an 850 MHz Pentium III processor running Windows 2000. The frequency domain features are more efficient, but the computational complexity is higher. On average, the times needed for the extraction of the frequency features are less than 1 second, the "rotation invariant" feature is extracted in less than 0.2 s, while the cords-based descriptor needs only 0.03 s to be extracted. However, we consider that the retrieval performance is more important in this trade-off.

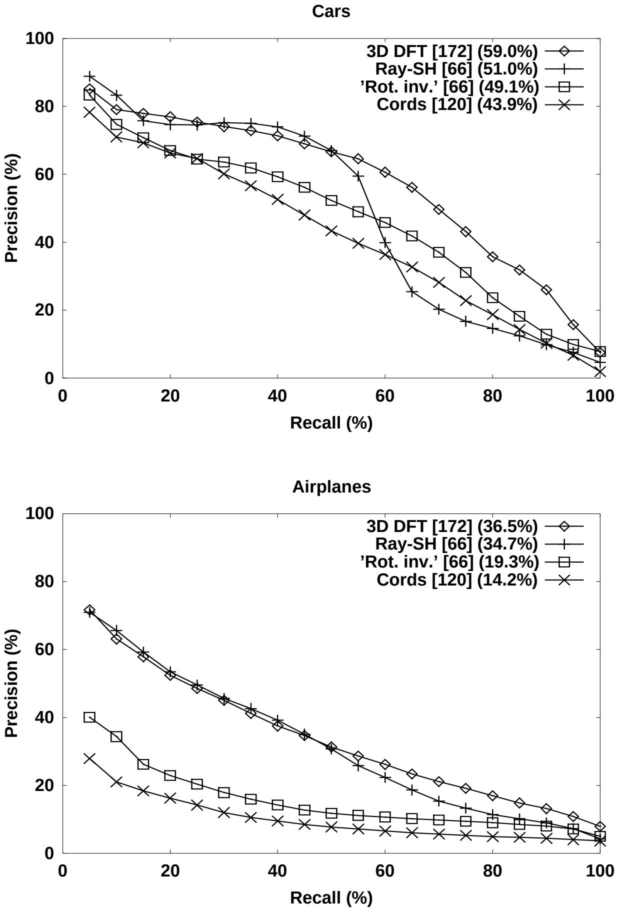


Fig. 2. Average precision vs. recall of queries for cars and airplanes using the proposed descriptor, the ray-based with spherical harmonic representation [10] (Ray-SH), the "rotation invariant" [6], and the cords-based descriptor [4]. The mean precision values and the dimensions are given in the brackets.

VI. Discussion and Conclusion

In this paper, a new approach for characterizing spatial properties of triangle mesh models is presented. The pose normalization step secures invariance properties desirable for retrieval applications, while the robustness with respect to level-of-detail is provided by the definition of the feature vector. A voxelized model obtained in the way explained in the section III can be regarded as a feature in the spatial domain. The frequency domain representation is obtained by applying a suitable transform, i.e., 3D DFT. Thereby, a representation of the feature is more compact and effective for retrieval applications.

A drawback of the presented descriptor is that problems with outliers may occur, because of the use of the bounding cube. An approach to solve this problem is given in [9], where the feature is encoded in the spatial domain (octree). As already mentioned, the l_1 or l_2 norms are ineffective when dealing with features represented in the spatial domain, therefore, we consider a modification of Hausdorff distance as well as some alternative approaches for a similarity metric.

As a proof of concept we provided results from a couple of experiments. An example of evaluation, used to conclude which type of feature vectors is the most suitable for a given class of 3D-objects is shown, as well. Generally, the proposed method is better than the previous approaches [1,4,6,10].

References

[1] M. Heczko, D. Keim, D. Saupe, and D. V. Vranić, "A method for similarity search of 3D objects", *Proc. BTW 2001*, Oldenburg, Germany, pp. 384, 2001. (in German)

[2] MPEG Requirements Group, "Overview of the MPEG-7 Standard (version 3.0)", Doc. ISO/MPEG N3445, MPEG Geneva Meeting, 2000.

[3] MPEG Video Group, "MPEG-7 Visual part of eXperimetation Model (version 9.0)", Doc. ISO/MPEG N3914, MPEG Pisa Meeting, 2001.

[4] E. Paquet and M. Rioux, "Nefertiti: a Query by Content System for Three-Dimensional Model and Image Databases Management", *Image and Vision Computing*, vol. 17, p. 157, 1999.

[5] M. Petrou and P. Bosdogianni, *Image Processing: The Fundamentals,* John Wiley, 1999.

[6] M. T. Suzuki, T. Kato, and N. Otsu, "A Similarity Retrieval of 3D Polygonal Models Using Rotation Invariant Shape Descriptors", *Proc. SMC 2000*, Nashville, Tennessee, p. 2946, 2000.

[7] Univ. California at Berkeley, School of Information Management and Systems project, How Much Information? http://www.sims.berkeley.edu/how-much-info/

[8] D. V. Vranić and D. Saupe, "3D Model Retrieval", *Proc. SCCG 2000*, May 3-6, Budmerice, Slovakia, p. 89, 2000.

[9] D. V. Vranić and D. Saupe, "A Feature Vector Approach for Retrieval of 3D Objects in the Context of MPEG-7", *Proc. ICAV3D 2001*, Mykonos, Greece, pp. 37, 2001.

[10] D. V. Vranić, D. Saupe, and J. Richter, "Tools for 3D-object retrieval: Karhunen-Loeve Transform and spherical harmonics", *IEEE 2001 Workshop Multimedia Signal Processing*, Cannes, France, (in press), 2001.