REGULAR PAPER

Benjamin Bustos · Daniel Keim · Dietmar Saupe · Tobias Schreck · Dejan Vranić

An experimental effectiveness comparison of methods for 3D similarity search

Abstract Methods for content-based similarity search are fundamental for managing large multimedia repositories, as they make it possible to conduct queries for similar content, and to organize the repositories into classes of similar objects. 3D objects are an important type of multimedia data with many promising application possibilities. Defining the aspects that constitute the similarity among 3D objects, and designing algorithms that implement such similarity definitions is a difficult problem. Over the last few years, a strong interest in 3D similarity search has arisen, and a growing number of competing algorithms for the retrieval of 3D objects have been proposed. The contributions of this paper are to survey a body of recently proposed methods for 3D similarity search, to organize them along a descriptor extraction process model, and to present an extensive experimental effectiveness and efficiency evaluation of these methods, using several 3D databases.

Keywords 3D model retrieval · Feature based similarity search methods · Retrieval effectiveness

1 Introduction

The development of effective and efficient similarity search methods for multimedia data is an important research issue due to the growing amount of digital audiovisual information that is becoming available. In digital libraries that are built from heterogenous data sources, typically consistent annotations are not available in order to organize and access the objects. Therefore, automatic content-based methods for similarity estimation of multimedia objects are required. In the case of 2D images along with the growth of available data volumes, a wealth of similarity notions and retrieval systems has evolved. In 2000, Veltkamp et al. [1] surveyed 39 different content-based image retrieval systems. A similar

development can be expected for 3D data, as 3D objects are powerful means for information dissemination with applications in such important fields as design and construction, education, simulation and entertainment.

Similarity search methods for 3D objects have to address a number of problems in order to achieve desirable invariance properties with respect to position, scale and rotation. They also have to select suitable object characteristics for similarity estimation. Often, a *feature vector* approach is used for performing similarity search. Already, there exist a variety of proposed methods that can be used to implement 3D similarity search systems. As these methods are rather new, to date few comprehensive experimental or theoretical studies contrasting the different methods exist. We have developed a retrieval system that implements many different 3D descriptors from our own as well as other researchers' work. In this paper, we present a survey of all descriptors implemented in our system, and empirically evaluate their retrieval performance based on extensive similarity search experiments conducted on several ground-truth classified databases.

This paper is organized as follows. Section 2 introduces the main problems that 3D similarity search methods have to address. It distinguishes the feature vector approach from other paradigms for conducting similarity search. It also presents a possible scheme for the classification of 3D descriptors. Section 3 then reviews a body of different feature-based descriptors from the recent literature on 3D similarity search. In Sect. 4, an effectiveness evaluation realized by extensive ground-truth based retrieval experiments contributes towards a comparison of the algorithms reviewed in Sect. 3. Section 5 presents the conclusions.

2 Similarity search of 3D objects

3D objects may be very complex, both in terms of the data structures and methods used to represent and to visually render such objects, as well as in terms of the topological and geometric structures of the objects themselves. The primary

B. Bustos(✉) · D. Keim · D. Saupe · T. Schreck · D. Vranić
Department of Computer and Information Science, University of Konstanz, Universitaetsstr. 10, 78457 Konstanz, Germany.
E-mail: {bustos, keim, saupe, schreck, vranic}@informatik.uni-konstanz.de

goal in the 3D, as well as in other similarity search domains, is to design algorithms with the ability to effectively and efficiently execute similarity queries. Direct geometric matching is an option. Here, it is measured how easily a given object can be transformed into another one, and the cost associated with this transform serves as the metric for similarity [2]. However, directly comparing all objects of a database with a query object is time consuming and may be difficult, because 3D objects can be represented in many different formats and may exhibit widely varying complexity. Given that it is also not clear how to use geometry directly for efficient similarity search, in typical methods the 3D data is transformed in some way to obtain numerical descriptors for indexing and retrieval. These descriptors characterize certain features of 3D objects and can be efficiently compared to each other in order to identify similar shapes and to discard dissimilar ones.

The extraction of shape descriptors generally can be regarded as a multistage process (see Fig. 1). In this process, a given 3D object, usually represented by a polygonal mesh, is first preprocessed to achieve the required invariance and robustness properties. Then, the object is transformed so that its character is either of surface type, or volumetric, or captured by one or several 2D images. Then, a numerical analysis of the shape takes place, from the result of which finally the feature descriptors are extracted. We briefly sketch these basic steps in the following:

1. *Preprocessing.* Several requirements that suitable methods for 3D similarity search should fulfill can be identified. The methods should be *invariant* with respect to changes in rotation, translation, and scale of 3D models in their reference coordinate frame. Ideally, an arbitrary combination of translation, rotation and scale applied to one object should not affect its similarity measure with respect to another object. In other words, the features comprising the shape descriptor ideally should not depend on the arbitrary coordinate frames that the authors of 3D models have chosen. Suitable methods should also be *robust* with respect to variations of the level-of-detail, and to small variations of the geometry and topology of the models. In some applications, invariance with respect to anisotropic scaling may also be desirable.

2. *Type of object abstraction.* A polygonal mesh can be seen in different ways. We may regard it as an ideal mathematical surface, infinitely thin, with precisely defined properties of differentiability. Alternatively, we may look at it as a thickened surface that occupies some portion of volume in 3D space, or for watertight models as a boundary of a solid volumetric object. The transformation of a mesh into one of these forms is typically called voxelization. Statistics of the curvature of the object surface is an example of a descriptor based directly on a surface, while measures for the 3D distribution of object mass, e.g., using moment-based descriptors, belong to the volumetric type of object abstraction. A third way to capture the character of a mesh would be to project it onto one or several image planes producing renderings, corresponding depth maps, silhouettes, and so on, from which descriptors can be derived.

3. *Numerical transformation.* The main features of meshes in one of the types of object abstractions outlined before can be captured numerically using one of various methods. Voxels grids and image arrays can be Fourier or Wavelet transformed, and surfaces can be adaptively sampled. This yields a numerical representation of the underlying object. It is not required that the numerical representation allows the complete reconstruction of the 3D object. However, these numerical representations are set up to readily extract the mesh shape descriptors in the final phase of the process.

4. *Descriptor generation.* We propose to group the descriptors for 3D shape in three main categories based on their form.

   (a) *Feature vectors*, or *FVs*, consist of elements in a vector space equipped with a suitable metric. Usually, the Euclidean vector space is taken with dimensions that may easily reach several hundreds. Such feature vectors may describe conceptually different types of shape information, such as spatial extent, visual expression, surface curvature, and so forth.

   (b) In *statistical approaches,* 3D objects are inspected for specific features, which are summarized usually in the form of a histogram. For example, in simple cases this amounts to the summed up surface area in specified volumetric regions, or, more complex, it may collect statistics about distances of point pairs randomly selected from the 3D object.

   (c) The third category is better suited for *structural 3D object shape description* that can be represented in the form of a graph [3, 4]. A graph can more easily represent the structure of an object that is made up of or can be decomposed into several meaningful parts,
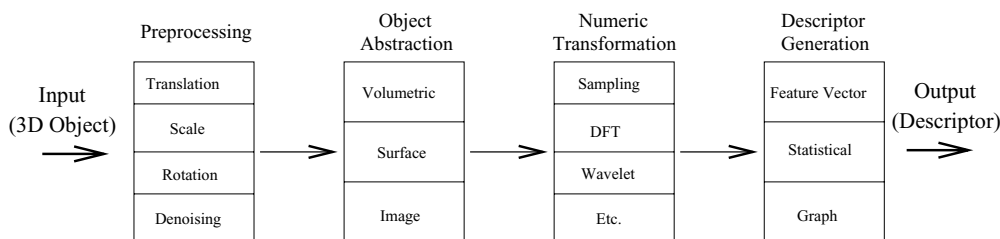
Preprocessing	Object Abstraction	Numeric Transformation	Descriptor Generation
Translation	Volumetric	Sampling	Feature Vector
Scale		DFT	
Rotation	Surface	Wavelet	Statistical
Denoising	Image	Etc.	Graph

Input (3D Object) → Preprocessing → Object Abstraction → Numeric Transformation → Descriptor Generation → Output (Descriptor)

Fig. 1 Descriptor extraction process model

such as the body and the limbs of objects modeling animals. However, finding a good dissimilarity measure for graphs is not as straightforward as for feature vectors, and, moreover, small changes in the 3D object may lead to large changes in the corresponding structural graph, which is not ideal for solving the retrieval problem.

For a classification of 3D object retrieval methods we use the type of object abstraction from the second stage of the extraction pipeline as the primary category. Thus, we ask whether the descriptor used in the respective method is derived directly from the surface, or whether it is based on an intermediate volumetric or image type of abstraction. For a second level of differentiation we propose to look at the form of descriptors (feature vector, statistical, or structural). Therefore, we adopt a classification based on the abstraction setting and the form of descriptors rather than the semantics behind them. Other classifications are possible, see for example the survey of Tangelder and Veltcamp [5] or Loncaric [6].

The methods in the feature vector class are efficient, robust, easy to implement, and provide some of the best approaches [5, 7, 8]. Therefore, these are the most popular ones that are explored in the literature. Also in this work, we restrict to this case as the currently dominant framework for 3D retrieval systems. We do not imply, however, that the other methods may be inferior and should therefore be discarded from future research. Most of these methods have their particular strengths and may well be the ideal candidate for a specific application.

In this section, we continue to discuss the main design problems with the feature vector approach to similarity search in 3D retrieval systems.

2.1 Invariance requirements and the principal component analysis

Invariance and robustness properties can be achieved in different ways. If only relative object properties are used to define the descriptor, then the invariance is not a problem, e.g., as in [9]. These methods are typically found in the class of statistical methods. Integrating a similarity measure over the space of transformations [10] is another approach. This space of transformations is large, however, requiring complex computations or numerical approximations, e.g., by using Monte-Carlo integration.

Invariance with respect to rotation can be achieved with energy summation in certain frequency bands of spectral representations of suitable spherical functions [7, 11]. In a generalization of this method to volumetric representations one may achieve rotational invariance by an appropriate combination of Zernike moments [12]. The invariance with respect to translation and to scale must be achieved in these methods by an *a priori* normalization step, i.e., by translating the center of mass of the 3D object to the origin

and by scaling the objects so that they can be compared at that same scale.

Otherwise, the invariance properties can be obtained approximately by an additional preprocessing normalization step, which transforms the objects so that they are represented in a canonical reference frame. In comparison to the above mentioned works, besides the translation of the coordinate origin and the definition of a canonical scale, also a rotational transformation must be applied in order to complete the normalization. In such a reference frame, directions and distances are comparable between different models. The predominant method for finding this reference coordinate frame is pose estimation by principal components analysis (PCA) [13, 14], also known as Karhunen–Loeve transformation. The basic idea is to align a model by considering its center of mass as the coordinate system origin, and its principal axes as the coordinate axes.

An extension to normalizing (isotropic) scale is to factor out also anisotropic scale [15] so that the variance of the object along any direction is unity. This is achieved by scaling the object along its principal axes by the inverses of the corresponding eigenvalues. The three eigenvalues can be appended to the feature vector of the rescaled object, and with an appropriate distance metric one may either completely disregard the anisotropy of the model or assign an arbitrary importance to it, depending on the application or user preferences [15].

While the majority of proposed methods employs PCA in some form or another, several authors have stability concerns with respect to the PCA as a tool for 3D retrieval. On the other hand, omitting orientation information also omits valuable object information. Thus, there is a tradeoff between achieving intrinsic rotation invariance without rotating the object in a canonical orientation, and the discrimination power that can additionally be attained by not proceeding this way. A detailed thorough empirical analysis would have to compare both cases to the retrieval performance achievable by optimal pairwise object alignment. This is a hard to do experiment and still outstanding. For a more detailed discussion see [2, 11, 16, 17].

Apart from these invariance requirements, another property that some descriptors possess is the embedded multi-resolution property. Here, one given object descriptor progressively embeds object detail, which can be used for similarity search on different levels of resolution. It eliminates the need to extract and store multiple descriptors with different levels of resolution if multi-resolution search is required, e.g., for implementing a filter-and-refinement step. The main class of descriptors that implicitly provide the multi-resolution property are those that perform a Fourier transformation of object measures.

2.2 Feature vector paradigm

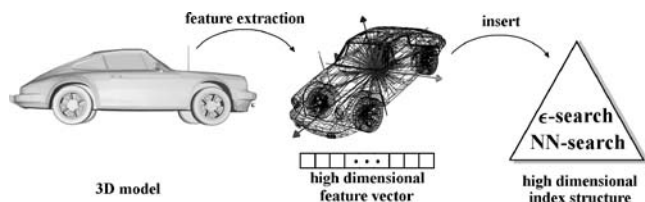The usage of feature vectors is the standard approach in multimedia retrieval [18]. Based on the real valued vectors

Fig. 2 Feature based similarity search

describing the objects in a database, a similarity query for a query object q is usually executed as a k-NN query, returning the k objects whose FVs have the smallest distance to q under a certain distance metric, sorted by increasing distance to the query. Figure 2 illustrates the basic idea of a FV-based similarity search system.

An important family of such distance metrics in vector spaces is the *Minkowski* (L_s) family of distances, defined as

$$L_s(\vec{x}, \vec{y}) = \left(\sum_{1 \leq i \leq d} |x_i - y_i|^s \right)^{1/s}, \qquad \vec{x}, \vec{y} \in I\!R^d.$$

Examples of these distance functions are L_1, which is called *Manhattan distance*, L_2, which is the *Euclidean distance*, and $L_\infty = \max_{1 \leq i \leq d} |x_i - y_i|$, which is called the *maximum distance*. Several extensions to the Minkowski distances have been studied, like the weighted Minkowski distance, where a weighting vector is assigned to the vector component distances, or the Mahalanobis distance, which engages a weight matrix to reflect cross-component similarity relationships between FVs (see for example, [19, 20]).

Figure 3 shows an example of a content-based similarity query in a 3D object database. The first object in the row is the query object (a model of a Formula-1 racing car), and the next objects are the nearest neighbors retrieved by the search system.

2.3 Effectiveness aspects

To provide effective retrieval, the retrieval algorithm is supposed to return the most relevant objects from the database on the first positions from the k-NN query, and to hold back irrelevant objects from this ranking. Therefore, it needs to implement discriminating methods to distinguish between similar and non-similar objects. The above described invariants should be provided. However, it is not possible to define

a unique notion of similarity because similarity is strongly application dependent. As is obvious from the number of different methods reviewed in Sect. 3, there exists a variety of concepts for geometric similarity. The most accessible one until now is global shape similarity. But, in spite of significant difference in their global shapes, two objects could still be considered similar given they belong to some kind of semantic class. Furthermore, partial similarity among different objects also constitutes an important similarity relationship within certain application domains. Most of the currently proposed methods for 3D similarity search are designed for global geometric similarity, while partial similarity still remains a largely unsolved problem.

3 Descriptors for 3D objects

In this section we follow the classification proposed in the previous section and review the discussed techniques, giving the main ideas. All of the methods are applicable to polygon meshes. For each method we present, we give a short descriptive name in italics, which will be used later on as a reference key in the experimental section.

3.1 Volumetric descriptors

3.1.1 A simple point cloud descriptor

In [21], the authors present a descriptor that relies on PCA registration but also is invariant to rotations of 90° along the principal axes. For the construction, an object is scaled into the unit cube with origin at the center of mass and axes parallel to the principal axes obtained by PCA. The unit cube is partitioned into $7 \times 7 \times 7$ equally sized cubic cells. For each of the cells, the frequency of some large number of points that are sampled uniformly from the surface, and that lie in the respective cell is determined, resulting in a coarse voxelization of the surface. To reduce the size of the descriptor, which would consist of 343 values, all grid cells are associated with one of 21 equivalence classes based on their location in the grid. All cells that coincide when performing arbitrary rotations of 90° about the principal axes are grouped together in one of the classes. For each equivalence class, the frequency data contained in the cells belonging to the respective equivalence class is aggregated, and
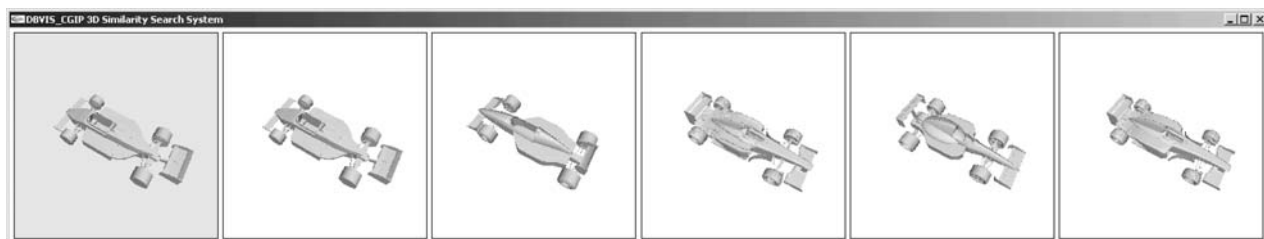

Fig. 3 Example of a 3D similarity query and the retrieved objects

the final descriptor of dimensionality 21 is obtained. The authors present retrieval performance results on a 3D database, on which $7 \times 7 \times 7$ is found to be the best grid dimensionality, but state that in general the optimal size of the descriptor may depend on the database chosen.

Please note that throughout this paper we refer to this method as the *rotational invariant FV*, although this is not precise as it is by design not invariant to arbitrary rotations.

3.1.2 Other descriptors based on surface voxelization

In [22] a FV based on the rasterization of a model into a voxel grid structure is presented, and the representation of this descriptor in either spatial or frequency domain is experimentally evaluated. The authors obtain their voxel descriptor by first subdividing the bounding cube of an object (after pose normalization) into $n \times n \times n$ equally sized voxel cells. Each of these voxel cells $v_{ijk}, i, j, k \in \{1, \ldots, n\}$ then stores the fraction $p_{ijk} = \frac{S_{ijk}}{S}$ of the object surface area S_{ijk} that lies in voxel v_{ijk}, where $S = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} S_{ijk}$ is the total surface area. The object's voxel cell occupancies then constitute the descriptor of dimension n^3 [23]. Efficient storage of voxel structures is feasible with octree structures, that avoid explicit storage of non-occupied parts of the voxel grid. Figure 4 illustrates a model represented in such an occupancy voxel grid. For similarity estimation under this feature, a metric can either be used directly on the voxel representations (*voxel FV*), or after the 3D Fourier-transform is applied on the voxelization (*3DDFT FV*). In the latter case, certain k magnitudes of lowest-frequency Fourier coefficients are used. The authors state to obtain better retrieval results using the Fourier-transformed voxel descriptor, instead of its spatial version.

3.1.3 Volume-based descriptors

In the preceeding method, triangle occupancies make up a FV for object description. This approach is appropriate when dealing with polygon meshes without further conditions. Such meshes typically come from heterogenous sources, e.g., from the Internet (also, informally referred to as "polygon soups"). On the other hand, if the 3D models are known to bound a solid object, then also volumetric occupancies
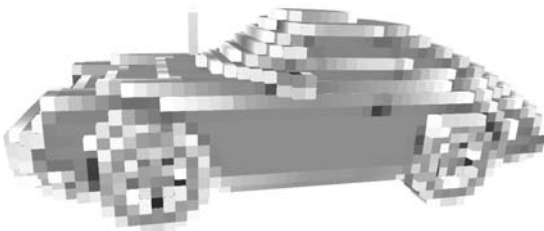
of the corresponding solid can be considered for FV construction. Several methods for similarity estimation based on voxelized volume data of normalized models have been proposed, e.g., in [13, 24, 25]. Another volume based FV is presented in [23]. Here, the six surfaces of an object's bounding cube are equally divided into n^2 squares each. Adding the object's center of mass to all squares, a total of $6n^2$ pyramid-like segments in the bounding cube are obtained. Assume that the polygon mesh bounds a solid object. The net proportion of volume occupied by the solid object in each segment of the bounding cube gives the components of the so-called *volume FV*. Figure 5 illustrates the idea in a 2D sketch.

3.1.4 Rotation invariant spherical harmonics descriptor

In [11], a descriptor based on the spherical harmonics representation of spherical functions [26] is proposed (named here *harmonics 3D*). The polygon mesh is voxelized into a grid with dimension $2R \times 2R \times 2R$, where cells are recorded as being either occupied or void. For the voxelization, the object's center of mass is translated into grid center position at (R, R, R), and it is scaled so that the average distance of the surface to the center of mass amounts to $\frac{R}{2}$, that is $\frac{1}{4}$ of the grid's edge length. By using this scale instead of scaling it so that the bounding cube fits tightly into the grid, it is possible to lose some object geometry. On the other hand, sensitivity with respect to outliers is expected to be reduced. The voxel grid is resampled yielding values of a binary spherical function $f_r(\theta, \phi)$, with integer radii r with respect to the grid origin up to length R. Thereby, the voxel space is transformed into a representation using spherical coordinates using R concentric shells. The resulting binary spherical functions are expressed using the spherical harmonics basis functions. The final feature vector is obtained by summing up squared magnitudes in each frequency band for each spherical function. These energy sums are invariant with respect to rotation about the center of mass, thus, the method does not require *a priori* pose normalization by PCA. An improvement can be obtained by replacing the binary spherical functions by values of a nonlinear distance transform (Michael Kazhdan, personal communication, 2003.).
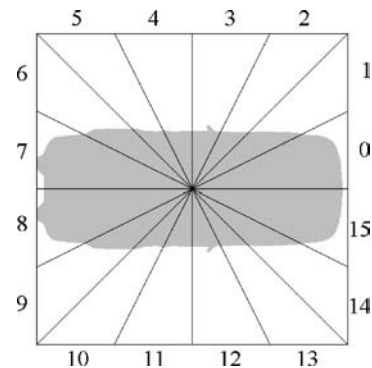

Fig. 4 The voxel-based feature vector compares occupancy fractions of voxelized models in the spatial or frequency domain

Fig. 5 Spatial partitioning scheme of the volume-based feature vector (2D illustration)

3.2 Desriptors directly based on surfaces

3.2.1 Geometric 3D moments

Statistical moments μ are scalar values that describe a distribution f. Parameterized by their order, moments represent a spectrum from coarse-level to detailed information of the given distribution [13]. In the case of 3D solid objects, which may be interpreted as a density function $f(x, y, z)$, the moment $\mu_{i,j,k}$ of order $n = i + j + k$ is defined in continuous form by:

$$\mu_{ijk} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y, z) x^i y^j z^k \, dx \, dy \, dz$$

As is well known, the complete (infinite) set of moments uniquely describes a distribution and vice versa. For a discrete form we consider a finite set of points P with unit mass per point. For this case, the moment formula becomes

$$\mu_{ijk} = \sum_{p \in P} x_p{}^i y_p{}^j z_p{}^k.$$

In [13] it is proposed to use the centroids of all triangles of a triangulated model (weighted by the area of the respective triangle) as input to moment calculation (*moments FV*), while in [27] object points found by the ray-based projection scheme described in 3.3.3 serve as the input (*ray-moments FV*). Because moments are not invariant with respect to translation, rotation and scale, PCA and scale normalization have to be applied prior to moment calculation. A FV can then be constructed by concatenating certain moments, e.g., all moments of order up to some value n.

3.2.2 Cords-based descriptor

A descriptor that combines information about the spatial extent and orientation of a 3D object is given in [13] (*cords FV*). The authors define a "cord" as a vector that runs from an object's center of mass to the centroid of a bounded surface region of the object, usually a triangle. For all object surface regions, such a cord is constructed. The descriptor is then built by calculating two histograms for the angles between the cords and the object's first two principal axes each, and one histogram for the distribution of the cord length. All three histograms are normalized by the number of cords and together make up the feature vector. Using the principal axes, the descriptor is approximately invariant to rotation and translation. It is also invariant to scale, as the length distribution is binned to the same number of bins for all objects. It can be inferred that the descriptor is not invariant to non-uniform tessellation changes.

3.2.3 Shape distribution with D_2

In [9], it is proposed to describe the shape of a 3D object as a probability distribution sampled from a shape function, which reflects geometric properties of the object. The algorithm calculates histograms called *shape distributions*, and

estimates similarity between two shapes by any metric that measures distances between distributions (e.g., Minkowski distances). The authors state that, depending on the shape function employed, shape distributions possess rigid transformation invariance, robustness against small model distortions, independence of object representation, and provide for efficient computation. The shape functions studied by the authors include the distribution of angles between three random points on the surface of a 3D object, and the distribution of Euclidean distances between one fixed point (specifically, the centroid of the boundary of the object was taken) and random points on the surface. Furthermore, they propose to use the Euclidean distance between two random points on the surface, the square root of the area of the triangle between three random points on the surface, or the cube root of the volume of the tetrahedron defined by four random points on the surface. Where necessary, a normalization step is applied for differences in scale.

As the analytic computation of distributions is feasible only for certain combinations of shape functions and models, the authors perform random sampling of many values from an object, and construct a histogram from these samples to describe the object shape. The authors perform retrieval experiments and report that the best experimental results are achieved using the distance function (distance between two random points on the surface), and using the L_1 norm of the probability density functions, which are normalized by aligning the mean of each two histograms to be compared (*D2 shape distribution FV*). Shape distributions for 3D retrieval have been further studied in [28, 29].

3.2.4 Shape spectrum descriptor

A descriptor for 3D retrieval proposed within the MPEG-7 framework for multimedia content description and reflecting curvature properties of 3D objects is presented in [30]. The *shape spectrum FV* is defined as the distribution of the *shape index* for points on the surface of a 3D object, which in turn is a function of the two principal curvatures at the respective surface point. The shape index gives the angular coordinate of a polar representation of the principal curvature vector, and it is implicitly invariant with respect to rotation, translation and scale. Because the shape index is not defined for planar surfaces, but 3D objects are usually approximated by polygon meshes, the authors suggest approximating the shape index by fitting quadratic surface patches to all mesh faces based on the respective face and all adjacent faces, and using this surface for shape index calculation. To compensate for potential estimation unreliability due to (near) planar surface approximations and (near) isolated polygonal face areas, these are excluded from the shape index distribution based on a threshold criterion, but their relative area is cumulated in two other attributes named *planar surface* and *singular surface*. These attributes together with the shape index histogram form the final descriptor.

Note that for the experiments to be presented in Sect. 4, we used the reference implementation of this descriptor

available from the MPEG-7 group [31], while for the rest of the descriptors we used our own implementations.

3.3 Image-based descriptors

3.3.1 Silhouette descriptor

A method called *silhouette FV* [23] characterizes 3D objects in terms of their silhouettes that are obtained from canonical renderings. The objects are first normalized using PCA and scaled into a unit cube that is axis-parallel to the principal axes. Then, parallel projections onto three planes, each orthogonal to one of the principal axes, are calculated. The authors propose to obtain descriptors by concatenating Fourier approximations of the three resulting contours. To obtain such approximations, a silhouette is sampled by placing a certain number of equally-spaced sequential points on the silhouette, and regarding the Euclidean distance between the image center and the consecutive contour points as the sampling values. These sampling values in turn constitute the input to the Fourier approximation. The concatenation of the magnitudes of certain low-frequency Fourier coefficients of the three contour images then gives the silhouette object descriptor. By PCA preprocessing, this descriptor is approximately rotation invariant. Figure 6 illustrates the contour images of a car object.

3.3.2 Depth buffer descriptor

Also in [23], another image-based descriptor is proposed. The so-called *depth buffer FV* descriptor starts with the same setup as the silhouette descriptor: The model is PCA-normalized and scaled into the canonical unit cube. Instead of three silhouettes, six grey-scale images are rendered using parallel projection, each two for one of the principal axes. Each pixel encodes in an 8-bit grey value the distance from the viewing plane (sides of the unit cube) to the object. These images correspond to the concept of z- or depth-buffers in computer graphics. After rendering, the six images are transformed using the standard 2D discrete Fourier transform, and the magnitudes of certain low-frequency coefficients of each image contribute to the depth buffer feature vector. Figure 7 shows the depth buffer renderings of a car object, as well as the star diagram visualizations of their respective Fourier transforms.
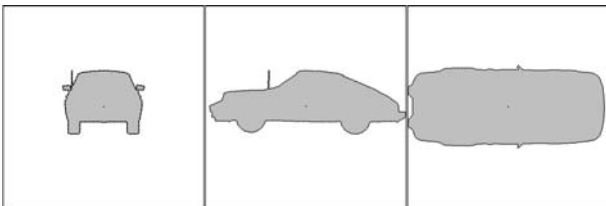

Fig. 6 Silhouettes of a 3D model. Note that, from left to right, the viewing direction is parallel to the first, second, and third principal axis of the model. Equidistant sampling points are marked along the contour
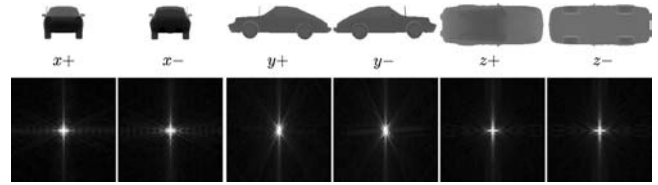

Fig. 7 Depth buffer based feature vector. The second row shows the Fourier transformation of the six images. Darker pixels in the first row indicate that the distance between view plane and object is smaller than on brighter pixels

3.3.3 Ray-based descriptors

In [27, 32] the authors propose a descriptor framework that is based on taking samples from a PCA-normalized 3D object by means of rays emitted from the center of mass O of an object in equally distributed directions u (directional unit vectors). For all such rays in direction u, starting from O the last intersection point $p(u)$ with a triangle t of the object is found, if such a point exists. Then, the distance $r(u) = |p(u) - O|$ is calculated, as well as the scalar product $x(u) = |u \cdot n(u)|$, where $n(u)$ is the normal vector of the respective triangle (if no intersection can be found for the ray u, $r(u)$ and $x(u)$ are set to zero). In the first proposed method, which considers spatial extent, the distances $r(u)$ make up the components of the so-called *ray FV*. A second descriptor, which considers polygon orientation, is obtained by setting the scalar products $x(u)$ as the feature components. The values $r(u)$ or $x(u)$ can be seen as samples of a function on the sphere. These samples, taken together, form a discrete spherical image, and therefore, we classify these descriptors as image-based.

In a second step, the authors propose, instead of using the sample values directly, to apply a transformation to the spherical functions, selecting certain low-frequency coefficient magnitudes as an embedded multi-resolution object descriptor. *Spherical harmonics* [26] provide the basis functions for the transform. In addition to using the spherical harmonics representation of either $r(u)$ (*rays-SH FV*) or $x(u)$ (*shading-SH FV*), also the combination of both measures in a complex function $y(u) = r(u) + i \cdot x(u)$ (with i denoting the imaginary unit) is considered by the authors, and called the *complex FV*. The authors demonstrate experimentally that this combined FV in spherical harmonics representation outperforms, in terms of retrieval effectiveness, both single versions in either spatial or spherical harmonics representation. The spherical harmonics transform is reversible. Figure 8 illustrates the ray-based sampling of $r(u)$, and a back-transform of the samples from the spherical harmonics representation to the spatial representation.

3.4 Summary

Table 1 presents an overview of the 3D shape descriptors reviewed in this section in the light of the processing pipeline from Fig. 1 and as discussed in Sect. 2. The column labeled
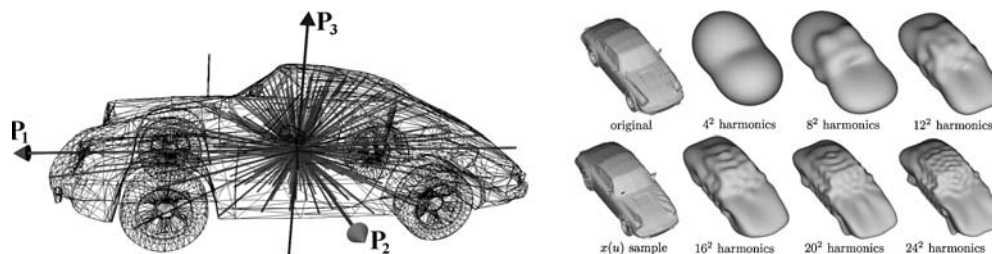
Fig. 8 The left image illustrates the ray-based feature vector. The right illustration shows the back-transform of the ray-based $r(u)$ samples from frequency to spatial domain

"Preprocessing" indicates the preprocessing steps that must be applied to the 3D object (R: Rotation, T: Translation, S: Scale). "Object abstr." indicates the classification with regard to the underlying object abstraction (volumetric-, surface-, and image-based). "Numerical transformation" indicates whether a numerical transformation is applied or not, and which kind of. Finally, "Descriptor type" indicates whether the final descriptor is a FV or a histogram.

4 Experimental comparison of 3D descriptors

4.1 Evaluation approach

The effectiveness of algorithms for similarity search can be assessed by different approaches. Under the *user oriented* approach, a number of users are to perform similarity search tasks using the algorithms under concern, and then certain measures of user satisfaction are aggregated. While this approach can reflect user satisfaction in real-world application settings, such experiments usually are not quantitatively reproducible and need careful definition of user tasks and selection of user groups.

Objective and reproducible effectiveness evaluations are possible if there exist generally accepted and readily available *ground-truth* classified data sets on which similarity search methods can be benchmarked. Examples include the

TREC text archives for information retrieval [34], or the UCI machine learning repository [35] for data mining research.

In evaluating 3D retrieval methods, until recently it was common practice for authors to individually compile databases and create ground-truth classifications on these databases for benchmarking purposes. These databases usually contain between hundred [36] up to tens of thousands [11] of 3D objects.

Given this practice, it was difficult to compare retrieval precision results reported by different authors, as the databases and the applied precision metrics usually differed. This situation may be about to change, as the Princeton Shape Retrieval and Analysis Group has recently released the *Princeton Shape Benchmark* (PSB) [37]. This benchmark consists of a carefully compiled set of 1,814 3D models in polygon mesh representation that were harvested from the Internet. The benchmark also includes object partitioning schemes on several different levels of abstraction, that is, several definitions of disjoint classes of objects, where all objects within the same class are to be considered similar. The benchmark is partitioned in one *Training* and one *Test* set, each containing half of the models. As to the types of objects considered, the PSB consists of models representing object classes that are familiar from the real world, such as animals, plants, vehicles, tools, or accessories. Not included are model classes from specialized application domains, e.g., CAD engineering or molecular biology. Of the different

Table 1 Overview of the methods discussed in this paper

Descriptor name	Section	Preprocessing	Object abstr.	Numerical transf.	Descriptor type
Rot. Inv. [21]	3.1.1	RTS	Volumetric	Sampling	Histogram
Voxel [23]	3.1.2	RTS	Volumetric	None	Histogram
3DDFT [22]	3.1.2	RTS	Volumetric	3D DFT	FV
Volume [23]	3.1.3	RTS	Volumetric	None	FV
Harmonics 3D [11]	3.1.4	TS	Volumetric	Spherical Harmonics	FV
Moments [13]	3.2.1	RTS	Surface	Sampling	FV
Ray Moments [27]	3.2.1	RTS	Surface	Sampling	FV
Cords [13]	3.2.2	RT	Surface	Sampling	Histogram
D2 Shape Dist. [9]	3.2.3	None	Surface	Sampling	Histogram
Shape Spectrum [30]	3.2.4	None	Surface	Curve fitting	Histogram
Silhouette [23]	3.3.1	RTS	Image	Sampling + DFT	FV
Depth Buffer [23]	3.3.2	RTS	Image	2D DFT	FV
Rays [33]	3.3.3	RTS	Image	Sampling	FV
Rays-SH [27, 32]	3.3.3	RTS	Image	Sampling + Sph. Harm.	FV
Shading-SH [32]	3.3.3	RTS	Image	Sampling + Sph. Harm.	FV
Complex-SH [32]	3.3.3	RTS	Image	Sampling + Sph. Harm.	FV

PSB classification schemes defined, the *PSB-Base* classification represents the most selective classification granularity, grouping objects strictly by function (semantic concept) as well as global shape. For our subsequent effectiveness evaluations, we consider this base classification.

In our own work, we had previously compiled a 3D database for evaluation purposes (the KN-DB) [38]. The KN-DB contains 1838 3D objects which we harvested from the Internet, and from which we subsequently manually classified 472 objects by global shape and function into 55 different model classes (the remaining models were left as "unclassified"). Comparing model types and classification philosophy in the PSB-Base and the KN-DB, we find that the partitioning of models into similarity classes was done in the same spirit, and both databases contain similar classes of objects. Having this in mind, the following evaluation, which is based on these two benchmarks, is valid for these 'real-world' 3D objects. Supposing that these model types form a significant part of the models freely available today on the Internet, the results may shed light on selecting algorithms for building general-purpose 3D Internet search engines. The results may not extend to the retrieval performance on specialized 3D content like repositories of machining parts. We presume that in order to assess the descriptors' retrieval performance in specialized 3D databases, separate test databases have to be designed and discussed first.

For performing the retrieval precision evaluation, we separately consider the three databases KN-DB, PSB-Train-Base and PSB-Test-Base. We use each of the classified objects within a given database as a query object, and the objects belonging to the same model class, excluding the query, were considered to be relevant to the query. Unclassified objects, or objects from classes different than the query object, were considered as irrelevant to the query.

For comparing the effectiveness of the search algorithms, we use *precision versus recall figures*, a standard evaluation technique for retrieval systems [39, 40]. *Precision* (P) is the fraction of the retrieved objects which are relevant to a given query, and *recall* (R) is the fraction of the relevant objects which have been retrieved from the database. That is, if N is the number of objects relevant to the query, A is the number of objects retrieved and R_A is the number of relevant objects in the result set, then

$$P = \frac{R_A}{A}, \qquad \text{and} \qquad R = \frac{R_A}{N}.$$

All our precision versus recall figures are based on the eleven standard recall levels (0%, 10%, ..., 100%) [39], and we average the precision figures over all test queries at each recall level. In addition to the precision at multiple recall points, we also employ the *R-precision* measure [39] (also known as *first tier*) for each query, which is defined as the precision when retrieving only the first N objects. The *R-precision* gives a single number to rate the performance of a retrieval algorithm.

$$R\text{-precision} = \frac{R_N}{N}$$

We evaluated the FVs using different levels of resolution, from 3 up to 512 dimensions, testing many different resolution settings as allowed by the individual methods. The resulting database-global retrieval performance values were obtained by averaging over all queries from a database given a feature vector of fixed dimensionality. For object preprocessing, we apply our variant of the principal component analysis [22] for those descriptors that require pose normalization. We used L_1 as the metric for distance computation, as this metric produced the best average retrieval results, as compared to the L_2 and $L_{\max}$ metric in our experiments.

4.2 Computational complexity of descriptors

Firstly, we compared the computational complexity of our 16 implemented descriptors. Typically, the computational cost of feature extraction is not of primary concern as extraction needs to be done only once for a database, while additional extraction must be performed only for those objects that are to be inserted into the database, or when a user submits a query object that is not yet indexed by the database.

Nevertheless, we present some efficiency measures taken on an Intel P4 2.4 GHz platform with 1 GB of main memory, running Microsoft Windows 2000, when extracting FVs from the KN-DB database. We observed that in general feature calculation is quite fast for most of the methods and 3D objects. Shape spectrum is an exception. Due to the approximation of local curvature from polygonal data by fitting of quadratic surface patches to all object polygons, this method is rather expensive. In general, PCA object preprocessing only constitutes a minor fraction of total extraction cost, as on average the PCA cost was only 3.59 s for the complete database of 1838 objects (1.95 ms per object on average).

Figure 9 shows the average extraction time per model as a function of the dimensionality of a descriptor. We did not include in this chart some of the descriptors that posses
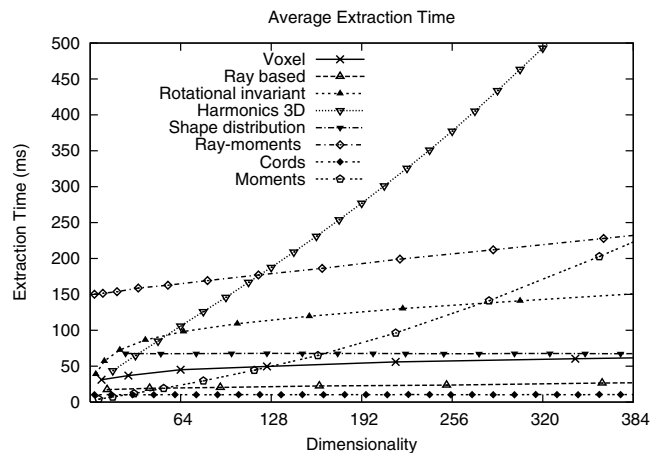

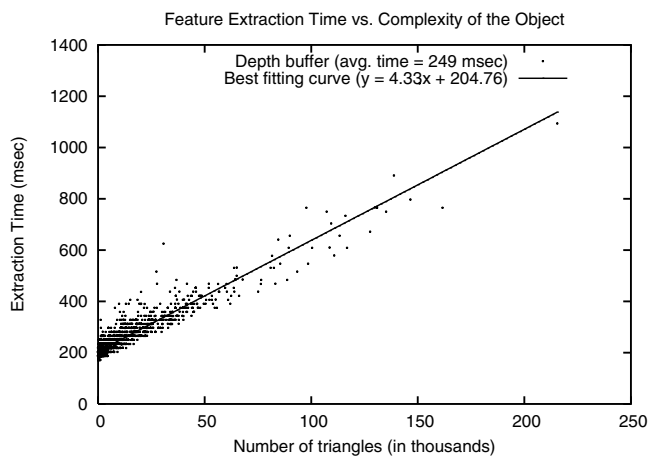**Fig. 9** Average extraction time for some of the descriptors while varying their corresponding dimensionality

Fig. 10 Best fitting curve for the extraction time of depth buffer

Table 2 Descriptor computation complexity

Descriptor	Avg. time (μs)
Depth buffer	249
Voxel	60
Complex	166
Rays-SH	162
Silhouette	50
3DDFT	1.545
Shading-SH	166
Ray-based	19
Rotation invariant	153
Harmonics 3D	167
Shape distribution	68
Ray-moments	228
Cords based	10
Moments	12
Volume	388
Shape spectrum	6.439

the multi-resolution property (because we computed those descriptors only once, using the maximum possible dimensionality), and we also discarded the curves for shape spectrum (almost constant and one order of magnitude higher than the others) and volume (a constant value for all possible dimensions, 387 ms). It follows that the extraction complexity depends on the implemented descriptor. For example, one of them has constant extraction complexity (shape distribution), others produce sub-linear curves (e.g., rotation invariant and cords), others produce linear curves (e.g., ray-moments), and the rest produce super-linear curves (e.g., harmonics 3D and moments).

If the dimensionality of the descriptor is fixed, then it is possible to produce a point cloud visualizing extraction time as a function of the number of triangles of the 3D object. Using this point cloud, we computed the best fitting linear curve by performing a linear regression. Figures 10 and 11 show two examples of best fitting curves for depth buffer and harmonics 3D descriptors respectively, using their best dimensionality according to Table 3 (see Sect. 4.3.1 for more details). Finally, Table 2 summarizes the extraction times (in

Table 3 Average R-precision of the 3D descriptors (KN-DB)

Descriptor	Best dim.	Avg. R-prec.
Depth buffer (DB)	366	0.3220
Voxel (VX)	343	0.3026
Complex (CP)	196	0.2974
Rays-SH (RS)	105	0.2815
Silhouette (SL)	375	0.2736
3DDFT (DF)	365	0.2622
Shading-SH (SH)	136	0.2386
Ray based (RA)	42	0.2331
Rotation invariant (RI)	406	0.2265
Harmonics 3D (H3)	112	0.2219
Shape distribution (SD)	188	0.1930
Ray moments (RM)	363	0.1922
Cords based (CO)	120	0.1728
Moments (MO)	31	0.1648
Volume (VL)	486	0.1443
Shape spectrum (SS)	432	0.1119

milliseconds) for all examined descriptors using their optimal dimensionality.

4.3 Effectiveness comparison between descriptors

4.3.1 Average results

Table 3 shows the best average R-precision values obtained for all implemented descriptors over all queries from the KN-DB, and their corresponding best dimensionality settings. The most effective descriptor according to this measure is the depth buffer with 366 dimensions.

Figures 12 and 13 show the precision vs. recall figures for all the implemented descriptors, evaluated on the KN-DB. Figure 12 shows the curves for the first eight descriptors according to Table 3, and Fig. 13 shows the curves for the last eight descriptors, according to the table.

The difference of the average R-precision values between the best performing descriptors is small, which
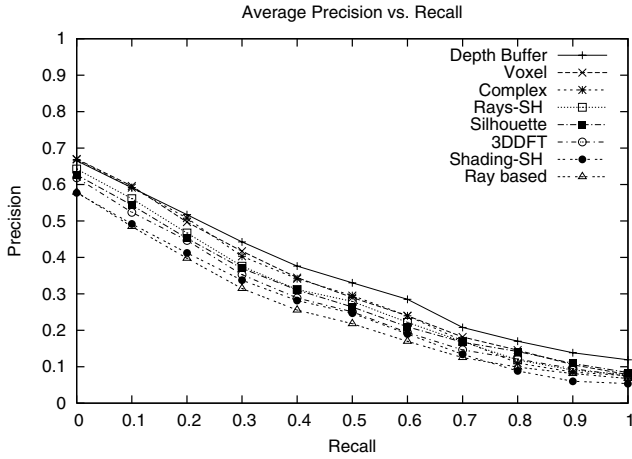

Fig. 11 Best fitting curve for the extraction time of harmonics 3D

Fig. 12 Average precision vs. recall with best dimensionality (KN-DB), first eight descriptors according to Table 3
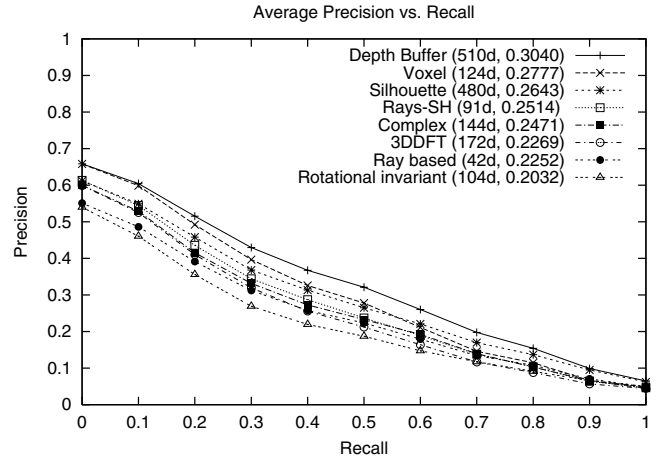

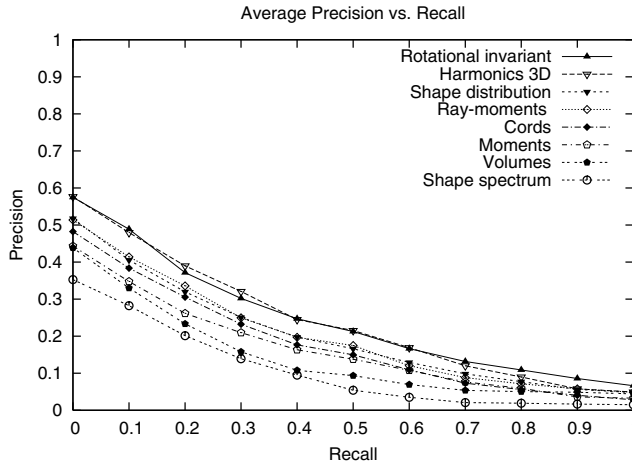**Fig. 14** Average precision vs. recall with best dimensionality (PSB-Test), first eight descriptors

Fig. 13 Average precision vs. recall with best dimensionality (KN-DB), last eight descriptors according to Table 3
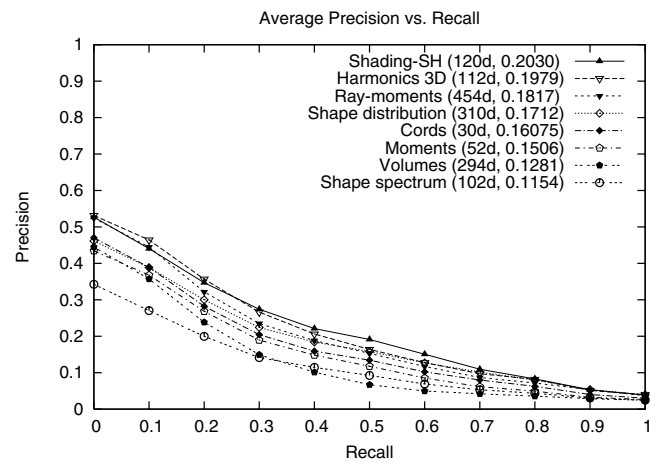

Fig. 15 Average precision vs. recall with best dimensionality (PSB-Test), last eight descriptors

implies that in practice these FVs should all be suited equally well for retrieval of "general-purpose" polygonal 3D objects. As a contrast, the effectiveness difference between the best and the least performing descriptor is significant (up to a factor of 3). We observed that descriptors which rely on consistent polygon orientation like shape spectrum or volume exhibit low retrieval rates, as consistent orientation is not guaranteed for many of the models retrieved from the Internet. Also, the moment-based descriptors in this test seem to offer only limited discrimination capabilities.

Figures 14 and 15 give the query-average precision vs. recall curves for the PSB-Test database when using the feature vector resolution providing the best average R-precision for this database (we include database-specific optimal dimensionality setting and achieved R-precision numbers in the legend). It is interesting to note that the results from the PSB-Test are quite similar to the ones obtained with the KN-DB. Despite the two databases having differences in size

and classification, the ranking of descriptors by retrieval performance, as well as the absolute performance figures are well comparable. Comparing the descriptor rankings from the KN-DB and the PSB-Test, there occur certain switches in the rankings, but all switches take place on roughly the same R-precision level. The two best performing descriptors and the four least performing descriptors retain their positions. We attribute the similarity of the retrieval performance results to the fact that both databases contain a comparable distribution of models, and manual classification was done in a comparable manner (function and shape).

We also evaluated the descriptors' retrieval performance on the PSB-Train database. While the absolute retrieval performance level using the PSB-Train (as measured by R-precision) is slightly higher than on the PSB-Test (about one to two percentage points), the descriptor rankings by retrieval performance are the same on both PSB partitions, except for one adjacent rank switch occurring between the eighth and ninth position in the ranking. This is not
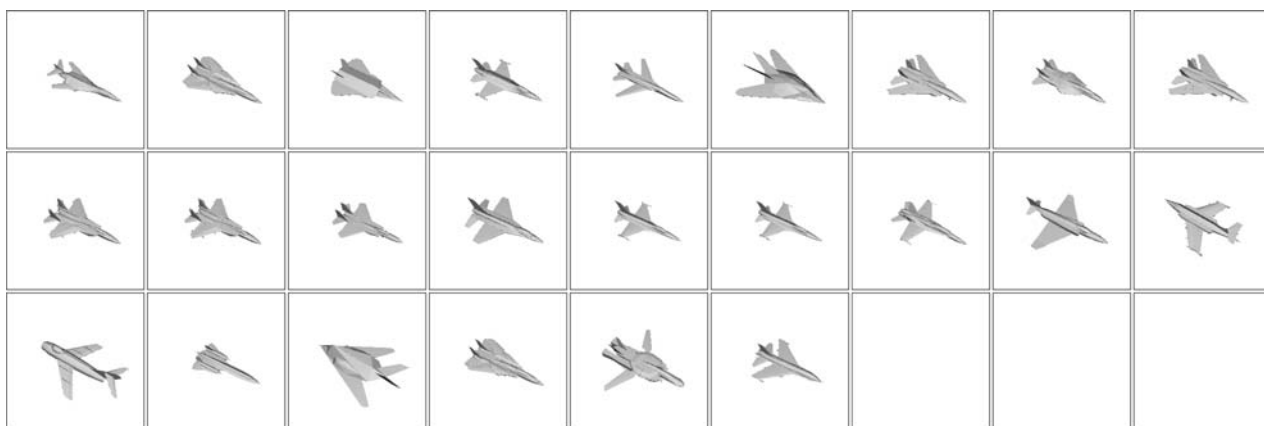
Fig. 16 The models from the planes model class (KN-DB)

surprising, considering the construction of the PSB Training and Test partitions [37].

4.3.2 Specific query classes

Many of the individual query classes from all three databases reflect the effectiveness ranking obtained from the database average, while certain shifts in the rankings are possible. Figures 16–21 illustrate two query classes from the KN-DB, namely one class with planes and one class with swords. The charts give the effectiveness results obtained with the descriptors for these query classes.

While the shape spectrum descriptor scores the least on database average, interestingly it achieves the best retrieval result in a KN-DB query class containing 56 models of humans (34% R-precision). As this descriptor considers the distribution of local curvature, it is able to retrieve human models that have different postures, while the other descriptors retrieve only those models where model posture is roughly the same (see Fig. 22 for an illustration).
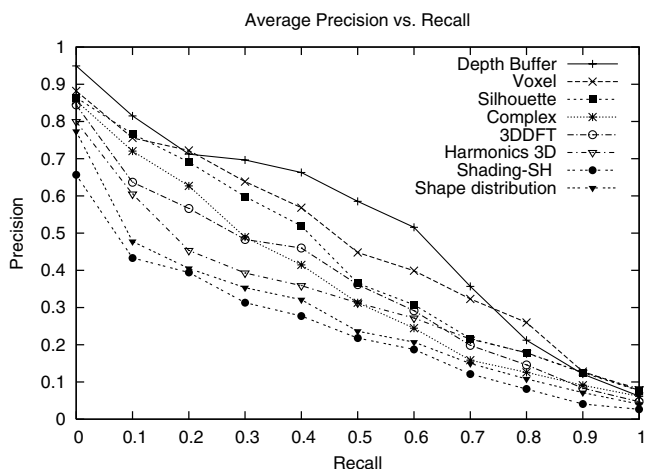
4.3.3 Level-of-detail

Robustness of the retrieval with respect to the level-of-detail in which models are given in a database is an important descriptor property. We test for this property using a query class from the KN-DB that contains seven different versions of the same model, in varying levels of resolution (specifically, models of a cow with 88 up to 5804 polygons). Except shape spectrum and cords, all descriptors manage to achieve perfect or near-perfect retrieval results. Figure 23 shows one example query in this class for three descriptors, and Fig. 24 gives the average R-precision numbers for all descriptors in this query class.

4.3.4 Principal axes

PCA normalization is required by most descriptor methods. For certain model classes, the PCA gives alignment results that are not in accordance with the alignment a user would intuitively expect based on semantic knowledge of the objects. For example, in the KN-DB we have defined a query
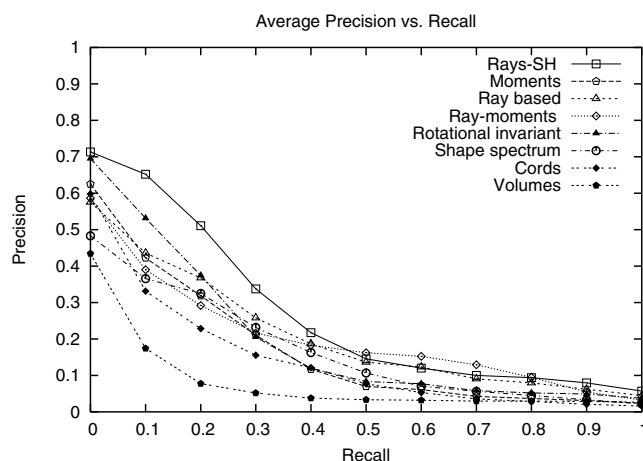

Fig. 17 Average precision vs. recall, planes model class (KN-DB), best eight descriptors for this class

Fig. 18 Average precision vs. recall, planes model class (KN-DB), last eight descriptors for this class
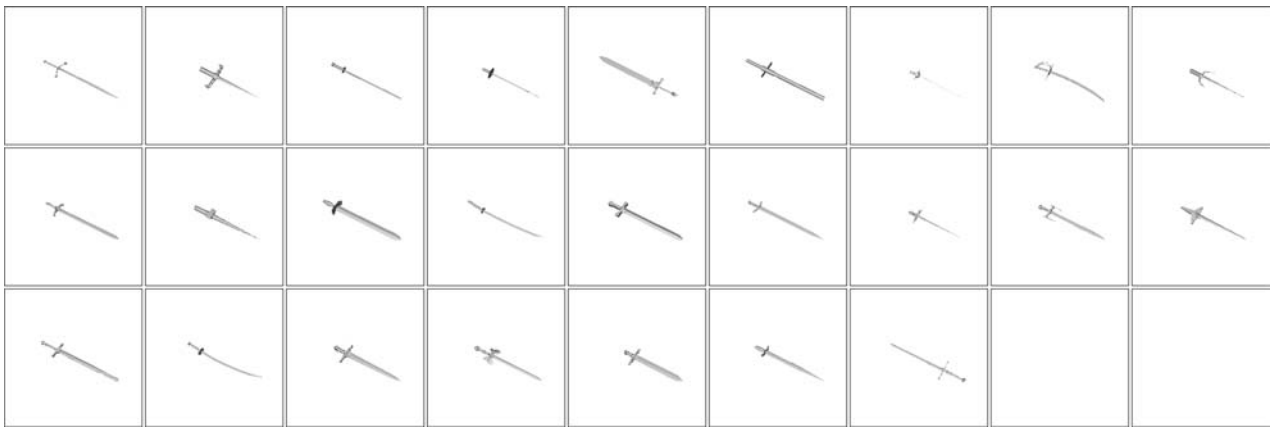
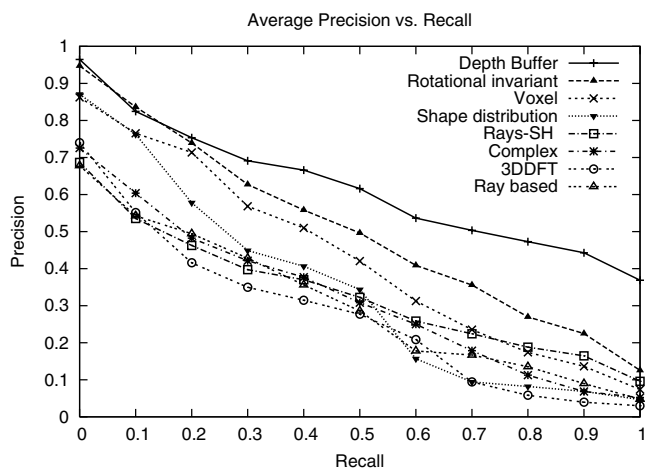**Fig. 19** The models from the swords model class (KN-DB)

Fig. 20 Average precision vs. recall, swords model class (KN-DB), best eight descriptors for this class
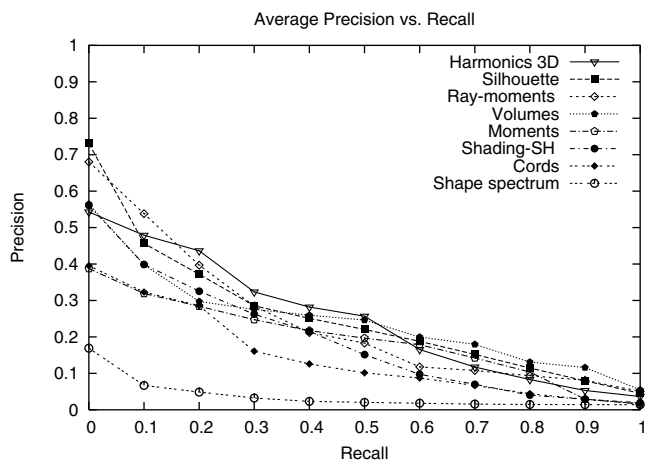

Fig. 21 Average precision vs. recall, swords model class (KN-DB), last eight descriptors for this class
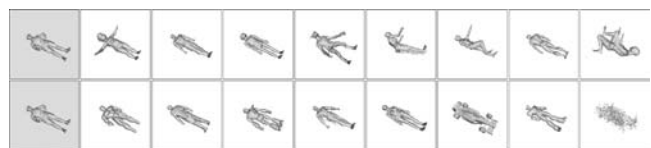

Fig. 22 Example query in the humans class (KN-DB). The first and second rows show the eight nearest neighbors using the shape spectrum and the depth buffer descriptors, respectively
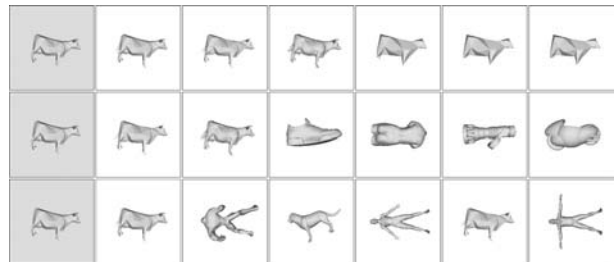

Fig. 23 Retrieval results for one example cow query object (KN-DB). The descriptors used are harmonics 3D, cords, and shape spectrum from the first through the third query row, respectively. All queries use the average-optimal descriptor resolution
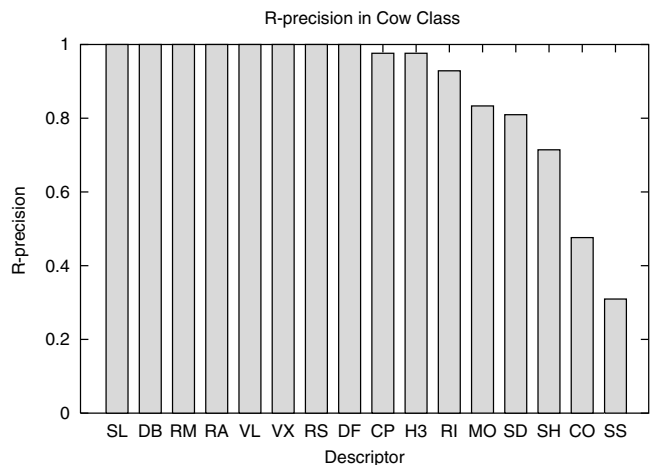

Fig. 24 *R*-precision values for the cows model class (KN-DB)

class with 4 arm chairs (see Fig. 25). In this class, PCA results are counterintuitive. While we cannot give an in-depth discussion of the PCA here, we note that in this query class an inherently rotational-invariant descriptor (harmonics 3D) provides the best class-specific retrieval performance (see Fig. 26).

4.4 Effectiveness as a function of the dimensionality of the descriptor

It is possible to calculate feature vectors at different resolutions, e.g., by specifying the number of rays with which to scan the objects, by specifying the number of Fourier coefficients to consider, etc. We are therefore interested in assessing the effect of descriptor resolution on retrieval effectiveness. Figures 27 and 28 (first eight and last eight descriptors, respectively) show the effect of the descriptor dimensionality on the query-average effectiveness in the KN-DB. Figures 29 and 30 show the same charts for the PSB-Test; again, the descriptors' retrieval performance behaves similarly in both databases. The figures show that the precision improvements are negligible for roughly more than 64 dimensions for most FVs, which means that it is not possible to improve the effectiveness of the search system by increasing the resolution of the FV over some dimensionality. It is interesting to note that this saturation effect is reached for most descriptors at roughly the same dimensionality

Fig. 25 Alignment problems of PCA in some classes. All objects are rendered with the camera looking at the center of mass along the least important principal axis
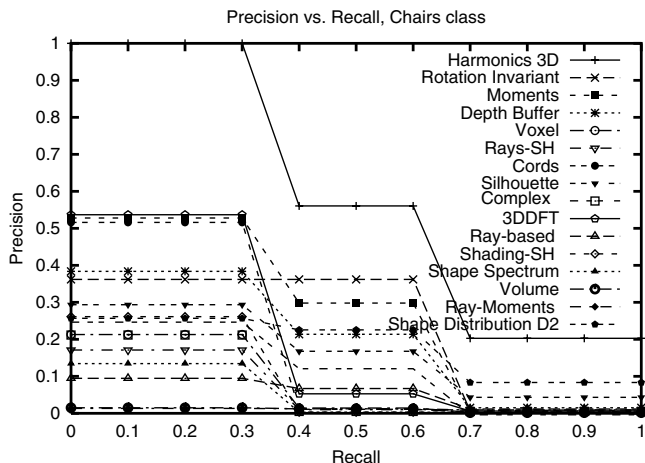

Fig. 26 Precision-recall curves for the chairs model class (KN-DB). The rotation-invariant descriptor harmonics 3D shows the best retrieval performance
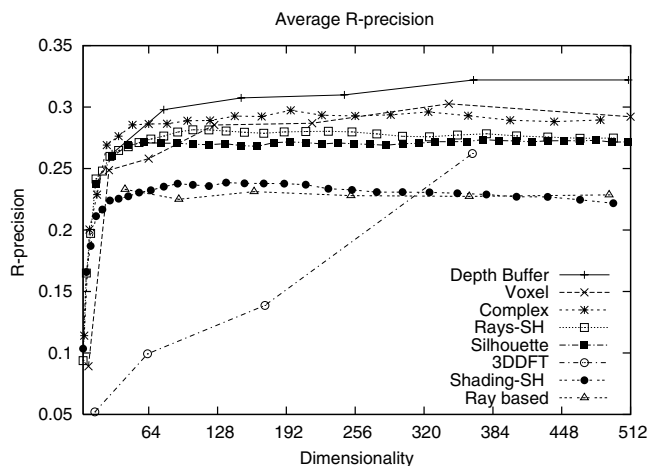

Fig. 27 Dimensionality vs. *R*-precision (KN-DB), first eight descriptors according to Table 3
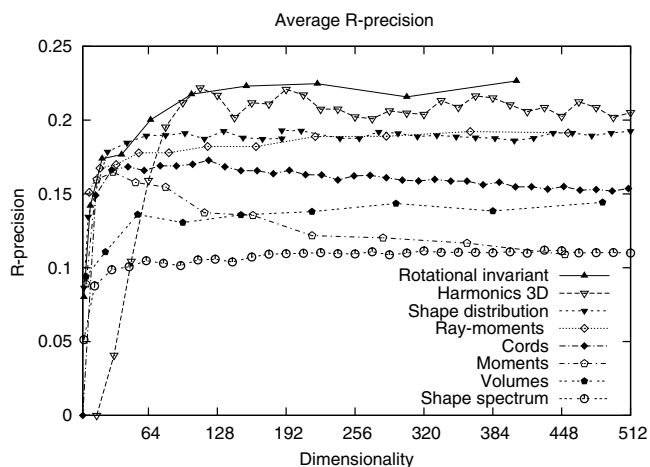

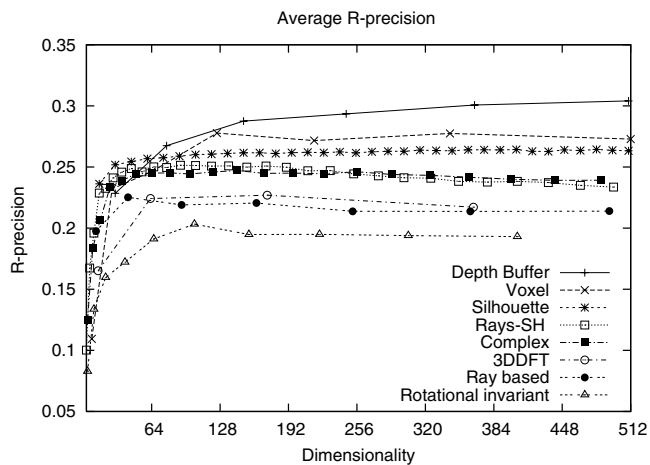**Fig. 28** Dimensionality vs. *R*-precision (KN-DB), last eight descriptors according to Table 3

Fig. 29 Dimensionality vs. *R*-precision (PSB-Test), first eight descriptors
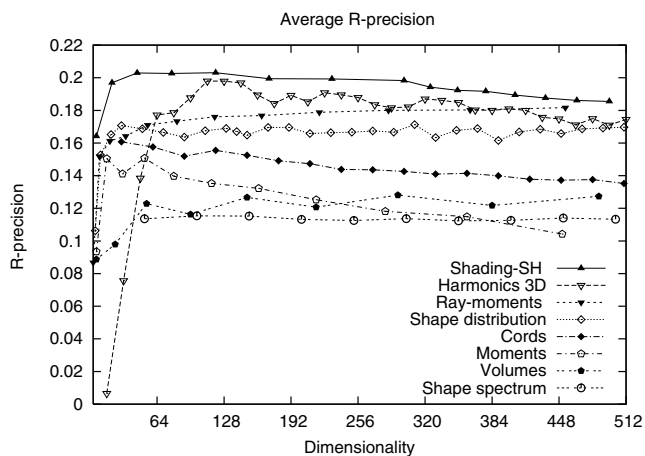
Fig. 30 Dimensionality vs. *R*-precision (PSB-Test), remaining descriptors

level. This is an unexpected result, considering that different FVs describe different characteristics of 3D objects.

5 Conclusions

In this paper, we discussed main problems involved in designing methods for the content-based retrieval of 3D objects by global shape similarity. We focused on methods that employ numerical descriptions (feature vectors and histograms) calculated from certain features of the objects as input to similarity estimation. We surveyed a body of recently proposed methods from this descriptor class, and organized them along a descriptor extraction process model.

We then experimentally evaluated the performance of these descriptors on two 3D object databases, formed by "general-purpose" models compiled from the Internet, using standard effectiveness measures from Information Retrieval (precision vs. recall diagrams, and *R*-precision values). We first compared the computational complexity of our implemented feature vectors. In practice, the normalization step and the descriptor computation cost is small, and almost all considered descriptors can be computed in less than a second for an object on average, and on a standard workstation. The experimental effectiveness comparison shows that there is a number of descriptors that have good database-average effectiveness and work well in many query classes (e.g., depth buffer, voxel and complex descriptors). Other descriptors work well with some specific model classes (e.g., shape spectrum with the human model class), and some of them are effective when the normalization step using PCA is not effective (e.g., harmonics 3D with the chair model class). Regarding the level-of-detail, the experimental results show that most descriptors can be considered robust, as they can retrieve similar objects with different level of detail.

It is interesting to note that the best performing descriptor within our experimental setup (depth buffer) is based on a

number of 2D object projections. This result is in accordance with [37, 41], where an advanced image-based descriptor that considers silhouettes rendered from many different directions is shown to produce excellent retrieval results. Further exploration of image-based 3D similarity search methods seems promising, as here it is possible to revert to many of the similarity models proposed in content-based 2D shape and image retrieval.

There remain significant problems in the research of content-based description and retrieval of 3D objects. We recall that the main effectiveness results presented in this paper refer to the average retrieval performance regarding global shape similarity in databases of general-purpose 3D models. Whether or not these results extend to domain-specific model databases, e.g., in a CAD context, has to be evaluated on adequately defined reference databases. Also, similarity notions conceptually higher than those contained in the database classifications used in this evaluation (namely, function and global shape), should be considered. Evaluating existing descriptors as well as designing new descriptors supporting similarity search in specialized 3D content remains to be explored.

The definition and efficient implementation of partial similarity search notions among 3D objects also remains a challenge. While graph-based descriptions of structural object features seem a natural approach for addressing partial similarity, applicability of these methods to similarity search in large 3D object databases remains open due to efficiency and robustness concerns.

How to improve the efficiency of numerical description-oriented search systems is also an open issue. The need for appropriate indexing techniques, considering the high dimensionality of the descriptors, is obvious. Moreover, if we consider the segmentation of objects as a possible approach for partial similarity search, then a database originally consisting of say, a few thousands of models, might be transformed into a database with tens or hundreds of thousands of models, where efficiency considerations become mandatory.

References

1. Veltkamp, R., Tanase, M.: Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, University Utrecht (2000)
2. Novotni, M., Klein, R.: A geometric approach to 3D object comparison. In: Proceedings of International Conference on Shape Modeling and Applications, pp. 167–175. IEEE CS Press (2001)
3. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.: Topology matching for fully automatic similarity estimation of 3D shapes.

In: Proceedings of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01), pp. 203–212. ACM Press (2001)

4. Sundar, H., Silver, D., Gagvani, N., Dickinson, S.: Skeleton based shape matching and retrieval. In: Proceedings of International Conference on Shape Modeling and Applications (SMI'03), pp. 130–142. IEEE CS Press (2003)

5. Tangelder, J., Veltkamp, R.: A survey of content based 3D shape retrieval methods. In: Proceedings of International Conference on Shape Modeling and Applications (SMI'04), pp. 145–156. IEEE CS Press (2004)

6. Loncaric, S.: A survey of shape analysis techniques. Pattern Recogn. **31**(8), 983–1001 (1998)

7. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In: Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP'03), pp. 156–164. Eurographics Association (2003)

8. Vranić, D.: 3D Model Retrieval. PhD thesis, University of Leipzig (2004)

9. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Trans. Graphics **21**(4), 807–832 (2002)

10. Ronneberger, O., Burkhardt, H., Schultz, E.: General-purpose object recognition in 3D volume data sets using gray-scale invariants—classification of airborne pollen-grains recorded with a confocal laser scanning microscope. In: Proceedings of International Conference on Pattern Recognition (2002)

11. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D.: A search engine for 3D models. ACM Trans. Graphics **22**(1), 83–105 (2003)

12. Novotni, M., Klein, R.: Shape retrieval using 3d zernike descriptors. Comp. Aided Design **36**(11), 1047–1062 (2004)

13. Paquet, E., Murching, M., Naveen, T., Tabatabai, A., Rioux, M.: Description of shape information for 2-D and 3-D objects. Signal Process Image Commun. 16:103–122 (2000)

14. Vranić, D., Saupe, D., Richter, J.: Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics. In: Proceedings of IEEE 4th Workshop on Multimedia Signal Processing, pp. 293–298 (2001)

15. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Shape matching and anisotropy. ACM Trans. Graphics **23**(3), 623–629 August (2004)

16. Tangelder, J., Veltkamp, R.: Polyhedral model retrieval using weighted point sets. Int. J. Image Graphics **3**(1), 209–229, (2003)

17. Vranić, D.: An improvement of rotation invariant 3D-shape descriptor based on functions on concentric spheres. In: Proceedings of IEEE International Conference on Image Processing (ICIP'03), Volume III, pp. 757–760 (2003)

18. Faloutsos, C.: Searching Multimedia Databases by Content. Kluwer, Dordrecht (1996)

19. Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P., Faloutsos, C., Taubin, G.: The QBIC Project: Querying images by content, Using color, Texture, and Shape. In: Proceedings of Storage and Retrieval for Image and Video Databases (SPIE), pp. 173–187 (1993)

20. Seidl, T., Kriegel, H.-P.: Efficient user-adaptable similarity search in large multimedia databases. In: Proceedings of 23rd International Conference on Very Large Databases (VLDB'97), pp. 506–515. Morgan Kaufmann (1997)

21. Kato, T., Suzuki, M., Otsu, N.: A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, pp. 2946–2952 (2000)

22. Vranić, D., Saupe, D.: 3D shape descriptor based on 3D Fourier transform. In: Proceedings of EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS'01), pp. 271–274. Comenius University (2001)

23. Heczko, M., Keim, D., Saupe, D., Vranić, D.: Methods for similarity search on 3D databases. Datenbank-Spektrum **2**(2), 54–63 (2002) in German

24. Keim, D.: Efficient geometry-based similarity search of 3D spatial databases. In: Proceedings of ACM International Conference on Management of Data (SIGMOD'99), pp. 419–430. ACM Press (1999)

25. Paquet, E., Rioux, M.: Nefertiti: A tool for 3-D shape databases management. Image and Vision Computing **108**, 387–393 (2000)

26. Healy, D., Rockmore, D., Kostelec, P., Moore, S.: FFTs for the 2-sphere - Improvements and variations. Journal of Fourier Analysis and Applications **9**(4), 341–385 (2003)

27. Vranić, D., Saupe, D.: 3D model retrieval with spherical harmonics and moments. In: Proceedings of DAGM-Symposium, LNCS 2191, pp. 392–397. Springer, (2001)

28. Ip, C., Lapadat, D., Sieger, L., Regli, W.: Using shape distributions to compare solid models. In: Proceedings of 7th ACM Symposium on Solid Modeling and Applications, pp. 273–280. ACM Press (2002)

29. Ohbuchi, R., Minamitani, T., Takei, T.: Shape similarity search of 3D models by using enhanced shape functions. In: Proceedings of Theory and Practice in Computer Graphics, pp. 97–104 (2003)

30. Zaharia, T., Prêteux, F.: 3D shape-based retrieval within the MPEG-7 framework. In: Proceedings of SPIE Conference on Nonlinear Image Processing and Pattern Analysis XII, vol. 4304, pp. 133–145 (2001)

31. MPEG-7 Video Group. MPEG-7 visual part of experimentation model. V.9. ISO/IEC N3914, MPEG-7, Pisa, January (2001)

32. Vranić, D., Saupe, D.: Description of 3D-shape using a complex function on the sphere. In: Proceedings of IEEE International Conference on Multimedia and Expo (ICME'02), pp. 177–180 (2002)

33. Vranić, D., Saupe, D.: 3D model retrieval. In: Proceedings of Spring Conference on Computer Graphics and its Applications (SCCG'00), pp. 89–93. Comenius University (2000)

34. US National Institute of Standards and technology. Text retrieval conference, http://trec.nist.gov/.

35. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)

36. Ohbuchi, R., Otagiri, T., Ibato, M., Takei, T.: Shape-similarity search of three-dimensional models using parameterized statistics. In: Proceedings of 10th Pacific Conference on Computer Graphics and Applications, pp. 265–274 (2002)

37. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Proceedings of International Conference on Shape Modeling and Applications (SMI'04), pp. 167–178. IEEE CS Press (2004)

38. Konstanz 3D Model Database. http://merkur01.inf.uni-konstanz.de/CCCC/.

39. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading, MA (1999)

40. van Rijsbergen, C.: Information Retrieval, 2nd ed. Butterworths, London, (1979)

41. Chen, D., Tian, X., Shen, Y., Ouhyoung, M.: On visual similarity based 3D model retrieval. In: Proceedings of Eurographics 2003, volume **22**(3) of Computer Graphics Forum, pp. 223–232. Blackwell, New York (2003)