First publ. in: ACM computing surveys 37 (2005), 4, pp. 345-387

Feature-based Similarity Search in 3D Object Databases

BENJAMIN BUSTOS, DANIEL A. KEIM, DIETMAR SAUPE, TOBIAS SCHRECK, and DEJAN V. VRANIĆ

Department of Computer and Information Science, University of Konstanz

The development of effective content-based multimedia search systems is an important research issue, due to the growing amount of digital audio-visual information. In the case of images and video, the growth of digital data has been observed since the introduction of 2D capture devices. A similar development is expected for 3D data, as acquisition and dissemination technology of 3D models is constantly improving. 3D objects are becoming 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 methods for 3D similarity search has arisen, and a growing number of competing algorithms for content-based retrieval of 3D objects have been proposed. We survey feature-based methods for 3D retrieval, and we propose a taxonomy for these methods. We also present experimental results, comparing the effectiveness of some of the surveyed methods.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Algorithms

Additional Key Words and Phrases: 3D model retrieval, Content-based similarity search

1. INTRODUCTION

The development of multimedia database systems and retrieval components is becoming increasingly important due to a rapidly growing amount of available multimedia data. As we see progress in the fields of acquisition, storage, and dissemination of various multimedia formats, one likes to apply effective and efficient database management systems to handle these formats. The need is obvious for image and video content. In the case of 3D objects, a similar development is expected in the near future. The improvement in 3D scanner technology and the availability of 3D models widely distributed over the Internet are rapidly contributing to create large

Authors' address: Department of Computer and Information Science, University of Konstanz. Universitaetsstr. 10 Box D78, 78457 Konstanz, Germany. E-mail: {bustos,keim,saupe,schreck,vranic}@informatik.uni-konstanz.de.

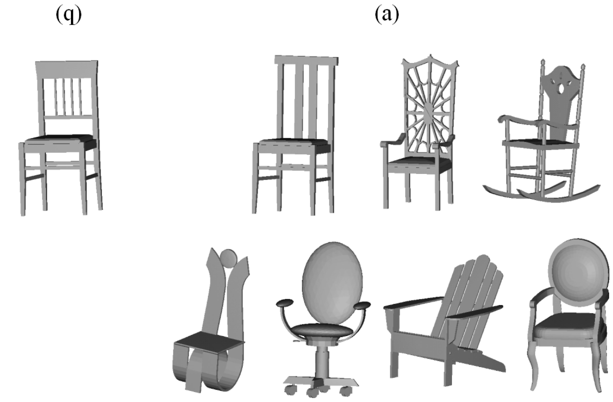


Fig. 1. Example of a similarity search on a database of 3D objects, showing a query object (q) and a set of possible relevant retrieval answers (a).

databases of this type of multimedia data. Also, the rapid advances in graphics hardware are making possible the fast processing of this complex data, making this technology available to a wide range of potential users at a relative low cost compared with the situation ten years ago.

One of the most important tasks in a multimedia retrieval system is to implement effective and efficient similarity search algorithms. Multimedia objects cannot be meaningfully queried in the classical sense (exact search), because the probability that two multimedia objects are identical is negligible, unless they are digital copies from the same source. Instead, a query in a multimedia database system usually requests a number of the most *similar objects* to a given query object or a manually entered query specification.

One approach to implement similarity search in multimedia databases is by using *annotation information* that describes the content of the multimedia object. Unfortunately, this approach is not very practicable in large multimedia repositories because in most cases, textual descriptions have to be generated manually, and are difficult to extract automatically. Also, they are subject to the standards adopted by the person who created them, and cannot encode all the information available in the multimedia object. A more promising approach for implementing a similarity search system is using the multimedia data itself, which is called *content-based search*. In this approach, the multimedia data itself is used to perform a similarity query. Figure 1 illustrates the concept of content-based 3D similarity search. The query object is a 3D model of a chair. The system is expected to retrieve similar 3D objects from the database, as shown in Figure 1.

1.1 Similarity search in 3D object databases

The problem of searching similar 3D objects arises in a number of fields. Example problem domains include Computer Aided Design/Computer Aided Manufacturing (CAD/CAM), virtual reality (VR), medicine, molecular biology, military applications, and entertainment:

—In medicine, the detection of similar organ deformations can be used for diagnostic purposes. For example, the current medical theory of child epilepsy assumes that an irregular development of a specific portion of the brain, called the hippocampus, is the reason for epilepsy. Several studies show that the size and shape of the deformation of the hippocampus may indicate the defect, and this is used to decide whether or not to remove the hippocampus by brain surgery. Similarity search in a database of 3D hippocampi models can support the decision process and help to avoid unnecessary surgeries [Keim 1999].

—Structural classification is a basic task in molecular biology. This classification can be successfully approached by similarity search, where proteins and molecules are modeled as 3D objects. Inaccuracies in the molecule 3D model due to measurement, sampling, numerical rounding, and small shift errors must be handled accordingly [Ankerst et al. 1999b].

—For a number of years many weather forecast centers include pollen-forecasts in their reports in order to warn and aid people allergic to different kinds of pollen. Ronneberger et al. [2002] developed a pattern recognition system that classifies pollen from 3D volumetric data acquired using a confocal laser scan microscope. The 3D structure of pollen can be extracted. Grey scale invariants provide components of feature vectors for classification.

—Forensic institutes around the world must deal with the complex task of identifying tablets with illicit products (drug pills). In conjunction with chemical analysis, physical characteristic of the pill (e.g., shape and imprint) are used in the identification process. The shape and imprints recognition methods include object bounding box, region-based shape and contour-based shape, which can be used to define a 3D model of the pill. A similarity search system can be used to report similarities between the studied pill and the models of known illicit tablets [Geradts et al. 2001].

—A 3D object database can be used to support CAD tools, because a 3D object can model exactly the geometry of an object, and any information needed about it can be derived from the 3D model, e.g., any possible 2D view of the object. These CAD tools have many applications in industrial design. For example, standard parts in a manufacturing company can be modeled as 3D objects. When a new product is designed, it can be composed by many small parts that fit together to form the product. If some of these parts are similar to one of the already designed standard parts, then the possible replacement of the original part with the standard part can lead to a reduction of production costs.

—Another industrial application is the problem of *best fitting shoes* [Novotni and Klein 2001a]. A 3D model of the client's foot is generated using a 3D scanning tool. Next, a similarity search is performed to discard the most unlikely fitting models according to the client's foot. The remaining candidates are then exactly inspected to determine the best match.

—A friend/foe detection system is supposed to determine whether a given object (e.g., a plane or a tank) is considered friendly or hostile, based on its geometric classification. This kind of system has obvious applications in military defense. One way to implement such a detection system is to store 3D models of the known friendly or hostile objects, and the system determines the classification of a given object based on the similarity definition and the database of reference objects. As such decisions must be reached in real-time and are obviously critical, high efficiency and effectiveness of the retrieval system is a dominant requirement for this application.

—Movie and video game producers make heavy usage of 3D models to enhance realism in entertainment applications. Re-usage and adaptation of 3D objects by similarity search in existing databases is a promising approach to reduce production costs.

As 3D objects are used in diverse application domains, different forms for object representation, manipulation, and presentation have been developed. In the CAD domain, objects are often built by merging patches of parametrized surfaces, which are edited by technical personnel. Also, constructive solid geometry techniques are often employed, where complex objects are modeled by composing primitives. 3D acquisition devices usually produce voxelized objects approximations (e.g., computer tomography scanners), or clouds of 3D points (e.g., in the sensing phase of structured light scanners). Other representation forms like swept volumes or 3D grammars exist. Probably the most widely used representation is to approximate a 3D object by a mesh of polygons, usually triangles. For a survey on important representation forms, see Campbell and Flynn [2001]. For 3D retrieval, basically all of these formats may serve as input to a similarity query. Where available, information other than pure geometry data can be exploited, e.g., structural data that may be included in a VRML representation. Many similarity search methods that are presented in the literature up to date rely on triangulations, but could easily be extended to other representation forms. Of course, it is always possible to convert or approximate from one representation to another one.

Research on describing shapes and establishing similarity relations between geometric and visual shape has been done extensively in the fields of *computer vision*, *shape analysis* and *computational geometry* for several decades. In computer vision, it is usually tried to *recognize* objects in a scene by *segmenting* a 2D image and then *matching* these segments to a set of a priori known reference objects. Specific problems involve accomplishing invariance with respect to lighting conditions, view perspective, clutter and occlusion. From the database perspective, it is assumed that the objects are already described in their entity, which can be directly used. Problems arise in the form of heterogeneous object representations (often certain properties of 3D objects cannot be assured), and the decision problem per se is difficult: What is the similarity notion? Where is the similarity threshold? How much tolerance is sustainable in a given application context, and which answer set sizes are required? In addition, the database perspective deals with a possibly large number of objects, therefore the focus lies not only on accurate methods, but also on fast methods providing efficient answer times even on large object repositories.

1.2 Feature vector paradigm

The usage of feature vectors (FVs) is the standard approach for multimedia retrieval [Faloutsos 1996], when it is not clear how to compare two objects directly. The *feature-based approach* is general and can be applied on any multimedia database, but we will present it from the perspective of 3D object databases.

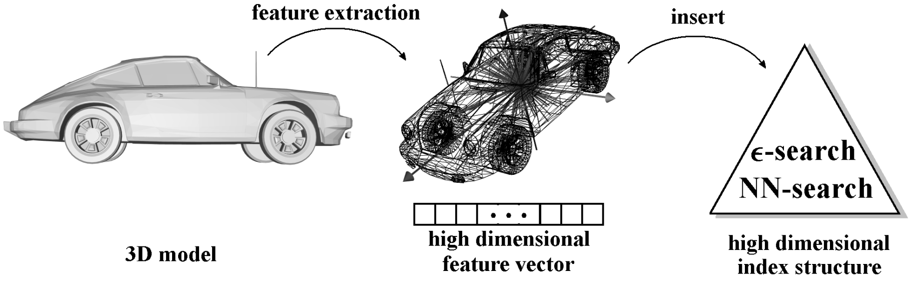


Fig. 2. Feature based similarity search.

1.2.1 *Feature vector extraction.* Having defined certain object aspects, numerical values are extracted from a 3D object. These values describe the 3D object and form a feature vector (FV) of usually high dimensionality. The resulting FVs are then used for indexing and retrieval purposes. FVs describe particular characteristics of an object based on the nature of the extraction method. For 3D objects, a variety of extraction algorithms have been proposed, ranging from basic ones, e.g., properties of an object's bounding box, to more complex ones, like the distribution of normal vectors or curvature, or the Fourier transform of some spherical functions that characterize the objects. It is important to note that different extraction algorithms capture *different characteristics* of the objects. It is a difficult problem to select some particular feature methods to be integrated into a similarity search system, as we find that not all methods are equally suited for all retrieval tasks. Ideally, a system would implement a set of "fundamentally" different methods, such that the appropriate feature could be chosen based on the application domain and/or user preferences. After a method is chosen and FVs are produced for all objects in the database, a distance function calculates the distance of a query point to all objects of the database, producing a ranking of objects in ascending distance.

Figure 2 shows the principle of a feature based similarity search. The FV is extracted from the original 3D query object, producing a vector $v \in \mathbb{R}^d$ for some dimensionality d.

The specific FV type and its given parametrization determine the extraction procedure and the resulting vector dimensionality. In general, different levels of resolution for the FV are allowed: More refined descriptors are obtained using higher resolutions. After the FV extraction, the similarity search is performed either by a full scan of the database, or by using an index structure to retrieve the relevant models.

1.2.2 *Metrics for feature vectors.* The similarity measure of two 3D objects is determined by a non-negative, real number. Generally, a similarity measure is therefore a function of the form

$$\delta : Obj \times Obj \rightarrow \mathbb{R}_0^+$$

where Obj is a suitable space of 3D objects. Small values of δ denote strong similarity and high values of δ correspond to dissimilarity.

Let $\mathbb{U}$ be the 3D object database and let q be the query 3D object. There are basically two types of similarity queries in multimedia databases:

—*Range queries*: A range query (q, r), for some tolerance value $r \in \mathbb{R}^+$, reports all objects from the database that are within distance r to q, that is, $(q, r) = \{u \in \mathbb{U}, \delta(u, q) \leq r\}$.

—*k Nearest neighbors (k-NN) queries*: It reports the k objects from $\mathbb{U}$ closest to q, that is, it returns a set $C \subseteq \mathbb{U}$ such as $|C| = k$ and for all $u \in C$ and $v \in \mathbb{U} - C$), $\delta(u, q) \leq \delta(v, q)$.

Assume that a FV of dimension d is taken for a similarity search. In typical retrieval systems, the similarity measure $\delta(u, v)$ is simply obtained by a metric distance $L(\vec{x}, \vec{y})$ in the d-dimensional space of FVs, where $\vec{x}$ and $\vec{y}$ denote the FVs of u and v, respectively. An important family of similarity functions in vector spaces is the *Minkowski* (L_s) family of distances, defined as:

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

Examples of these distance functions are L_1, which is called *Manhattan distance*, L_2, which is the *Euclidean distance*, and the *maximum distance* $L_\infty = \max_{1 \leq i \leq d} |x_i - y_i|$.

A first extension to the standard Minkowski distance is to apply a weight vector w, that weighs the influence that each pair of components exerts on the total distance value. This is useful if a user has knowledge about the semantics of the FV components. Then, she can manually assign weights based on her preferences with respect to the components. If no explicit such knowledge exists, it is still possible to generate weighting schemes based on relevance feedback, see e.g., [Elad et al. 2002]. The basic idea in relevance feedback is to let the user assign relevance scores to a number of retrieved results. Then, the query metric may automatically be adjusted such that the new ranking is in better agreement with the supplied relevance scores, and thereby (presumably) producing novel (previously not seen) relevant objects in the answer set.

If the feature components correspond to histogram data, several further extensions to the standard Minkowski distance can be applied. In the context of image similarity search, color histograms are often used. The descriptors then consist of histogram bins, and cross-similarities can be used to reflect natural neighborhood similarities among different bins. One prominent example for employing cross-similarities is the QBIC system [Ashley et al. 1995], where results from human perceptual research are used to determine a suitable cross-similarity scheme. It was shown that quadratic forms are the natural way to handle these cross-similarities formally, and that they can be efficiently evaluated for a given database [Seidl and Kriegel 1997]. If such intra-feature cross-similarities can be identified, quadratic forms may also be used for 3D similarity search, as done, e.g., in the *shape histogram* approach (cf. Section 3.4.2). Apart from Minkowski and quadratic forms, other distance functions for distributions can be borrowed from statistics and information theory. But, this variety also makes it difficult to select the appropriate distance function, as the retrieval effectiveness of a given metric depends on the data to be retrieved and the extracted features [Puzicha et al. 1999].

1.3 Overview

The remainder of this article presents a survey of approaches for searching 3D objects in multimedia databases under the feature vector paradigm. In Section 2, we first discuss fundamental issues of similarity search in 3D objects databases. In Section 3, we then review and classify feature-based methods for describing and comparing 3D objects that are suited for database deployment. A comparison in Section 4 tries to contrasts the surveyed approaches with respect to important characteristics, and gives experimental retrieval effectiveness benchmarks that we performed on a number of algorithms. Finally, in Section 5 we draw some conclusions and outline future work in the area.

2. PROBLEMS AND CHALLENGES OF 3D SIMILARITY SEARCH SYSTEMS

Ultimately, the goal in 3D similarity search is to design database systems that store 3D objects and effectively and efficiently support similarity queries. In this section, we discuss the main problems posed by similarity search in 3D object databases.

2.1 Descriptors for 3D similarity search

3D objects can represent complex information. The difficulties to overcome in defining similarity between spatial objects are comparable to those for the same task applied to 2D images. Geometric properties of 3D objects can be given by a number of representation formats, as outlined in the introduction. Depending on the format, surface and matter properties can be specified. The object's resolution can be arbitrarily set. Given that there is no founded theory on a universally applicable description of 3D shapes, or how to use the models directly for similarity search, in a large class of methods for similarity ranking the 3D data is transformed in some way to obtain *numeric descriptors* for indexing and retrieval. We also refer to these descriptors as *feature vectors* (FVs). The basic idea is to extract numeric data that describe the objects under some identified geometric aspect, and to infer the similarity of the models from the distance of these numerical descriptions in some metric space. The similarity notion is derived by an application context that defines which aspects are of relevance for similarity. Similarity relations among objects obtained in this way are then subject to the specific similarity model employed, and may not reflect similarity in a different application context.

The feature-based approach has several advantages compared to other approaches for implementing similarity search. The extraction of features from multimedia data is usually fast and easily parametrizable. Metrics for FVs, as the Minkowski distances, can also be efficiently computed. Spatial access methods [Böhm et al. 2001] or metric access methods [Chávez et al. 2001] can be used to index the obtained FVs. All these advantages make the feature-based approach a good candidate for implementing a 3D object similarity search engine.

3D similarity can also be estimated under paradigms other than the FV approach. Generally, non-numeric descriptions can be extracted from 3D objects, like structural information. Also, direct geometric matching is an approach. Here, it is measured how easily a certain object can be transformed into another one,

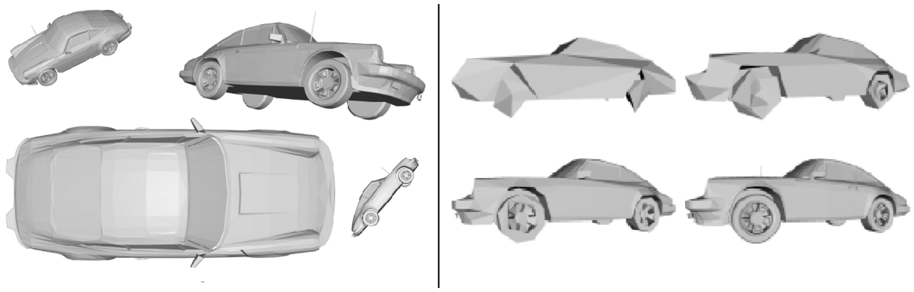


Fig. 3. A 3D object in different scale and orientation (left), and also represented with increasing level-of-detail (right).

and a cost associated by this transform serves as the metric for similarity. Usually these metrics are computationally costly to compute and do not always hold the triangle inequality, therefore it is more difficult to index the database under these alternative paradigms.

2.2 Descriptor requirements and 3D pose normalization

Considering the descriptor approach, one can define several requirements that effective FV descriptors should meet. Good descriptors should abstract from the potentially very distinctive design decisions that different model authors make when modeling the same or similar objects. Specifically, the descriptors should be *invariant* to changes in the orientation (translation, rotation and reflection) and scale of 3D models in their reference coordinate frame. That is, the similarity search engine should be able to retrieve geometrically similar 3D objects with different orientations. Figure 3 (left) illustrates different orientations of a Porsche car 3D object: The extracted FV should be (almost) the same in all cases. Ideally, an arbitrary combination of translation, rotation and scale applied to one object should not affect its similarity score with respect to another object.

Furthermore, a descriptor should also be *robust* with respect to small changes of the level-of-detail, geometry and topology of the models. Figure 3 (right) shows the Porsche car 3D object at four different levels of resolution. If such transformations are made to the objects, the resulting similarity measures should not change abruptly, but still reflect the overall similarity relations within the database.

Invariance and robustness properties can be achieved implicitly by those descriptors that consider relative object properties, e.g., the distribution of surface curvature of the objects. For other descriptors, these properties can be achieved by a *preprocessing normalization step*, which transforms the objects so that they are represented in a canonical reference frame. In such a reference frame, directions and distances are comparable between different models, and this information can be exploited for similarity calculation. The predominant method for finding this reference coordinate frame is pose estimation by principal component analysis (PCA) [Paquet et al. 2000; Vranić et al. 2001], also known as Karhunen-Loève transformation. The basic idea is to align a model by considering its center of mass and principal axes. The object is translated to align its center of mass with

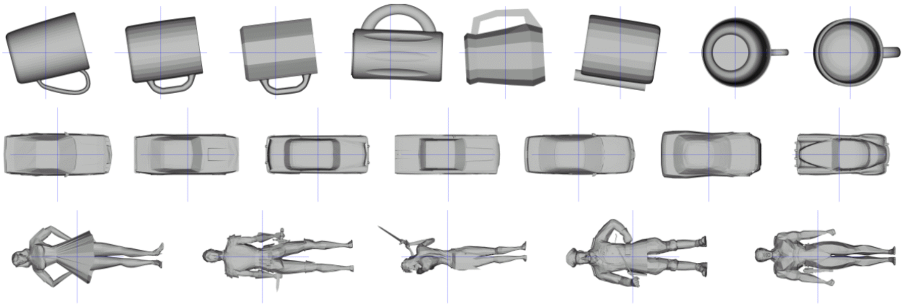


Fig. 4. Pose estimation using the PCA for three classes of 3D objects.

the coordinate origin (*translation invariance*), and then is rotated around the origin such that the x, y and z axes coincide with the three principal components of the object (*rotation invariance*). Additionally, *flipping invariance* may be obtained by flipping the object based on some moment test, and *scaling invariance* can be achieved by scaling the model by some canonical factor. Figure 4 illustrates PCA-based pose- and scaling normalization of 3D objects. For some applications, matching should be invariant with respect to anisotropic scaling. For this purpose, Kazhdan et al. [2004] proposed a method that scales objects, such that they are maximally isotropic before computing FVs for shape matching.

While PCA is a standard approach to pose estimation, several variants can be employed. When a consistent definition of object mass properties is not available, as is usually the case in mesh representations, one has to decide on the input to the PCA. Just using polygon centers or mesh vertices would make the outcome dependent on the tessellation of the model. Then, it is advantageous to use a weighing scheme to reflect the influence that each polygon contributes to the overall object distribution when using polygon centers or mesh vertices [Vranić and Saupe 2000; Paquet and Rioux 2000]. Analytically, it is necessary to integrate over all of the infinitesimal points on a polygon [Vranić et al. 2001]. Others use a Monte-Carlo approach to sample many polygon points [Ohbuchi et al. 2002] to obtain PCA input.

A few authors articulate fundamental critique on the PCA as a tool for 3D retrieval. Funkhouser et al. [2003] find PCA being unstable for certain model classes, and consequently propose descriptors that do not rely on orientation information. On the other hand, omitting orientation information may also omit valuable object information.

A final descriptor property that is also desirable to have is the *multi-resolution property.* Here, the descriptor embeds progressive model detail information, 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. A main class of descriptors that implicitly provide the multi-resolution property are those that perform a discrete Fourier or Wavelet transform of sampled object measures.

2.3 Retrieval system requirements

There are two major concerns when designing and evaluating a similarity search system: *Effectiveness* and *efficiency*. To provide effective retrieval, the system is supposed to return the most relevant objects from the database on the first ranks given a 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 clear what the exact meaning of similarity is. As obvious from the number of different methods reviewed in Section 3, there exist a variety of concepts for geometric similarity. The most formalizable one until now is global shape similarity, like illustrated in the first row of chairs shown in Figure 1. 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, for example like in the second row of chairs in Figure 1. Furthermore, partial similarity among different objects also constitutes a similarity relationship within certain application domains. Most of the current methods are designed for global geometric similarity, while partial similarity still remains a difficult problem.

On the other hand, the search system has to provide efficient methods for descriptor extraction, indexing and query processing on the physical level. This is a need, because it can be expected that 3D databases will grow rapidly once 3D scanning and 3D modeling become commonplace. In databases consisting of millions of objects with hundreds of thousands of voxels or triangles each, which need to be automatically described and searched for, efficiency becomes mandatory. Two broad techniques exist to efficiently conduct fast similarity search [Faloutsos 1996]. A *filter-and-refinement* architecture first restricts the search space with some inexpensive, coarse similarity measure. On the created candidate set, some expensive but more accurate similarity measure is employed in order to produce the result set. It is the responsibility of such filter measures to guarantee for no false dismissals, or at least only a few, in order to generate high-quality answer sets. Second, if the objects in a multimedia database are already feature-transformed to numerical vectors, specially suited high-dimensional data structures along with efficient nearest-neighbor query algorithms can be employed to avoid the linear scan of all objects. Unfortunately, due to the *curse of dimensionality* [Böhm et al. 2001], the performance of all known index structures deteriorates for high-dimensional data. Application of dimensionality reduction techniques as a post-processing step can help improving the indexability of high-dimensional FVs [Ngu et al. 2001].

Finally, note that in traditional databases the key-based searching paradigm implicitly guarantees full effectiveness of the search, so efficiency aspects are the major concern. In multimedia databases, where effectiveness is subject to some application and user context, efficiency and effectiveness concerns are of equal importance.

2.4 Partial similarity

Almost all available methods for similarity search in 3D object databases focus on global geometric similarity. In some application domains, also the notion of *partial similarity* is considered. In partial similarity, similarities in parts or sections of the objects are relevant. In some applications, complementarity between solid object segments constitutes similarity between objects, e.g., in the molecular docking problem [Teodoro et al. 2001]. In the case of 2D polygons, some solutions to the partial similarity problem have been proposed [Berchtold et al. 1997]. For 3D objects, to date it is not clear how to design fast segmentation methods that lead to suited object partitions, which could be compared pairwise. Although partial similarity is an important research field in multimedia databases, this survey focuses on global geometric similarity.

2.5 Ground truth

A crucial aspect for objective and reproducible effectiveness evaluation in multimedia databases is the existence of a widely accepted *ground truth*. Up to now, this is only partially the case for the research in 3D object retrieval. Up to now, each research group in this field has collected and classified their own 3D databases. In Section 4, we present our own prepared ground truth, which we use to experimentally compare the effectiveness of several feature-based methods for 3D similarity search. Recently, the carefully compiled *Princeton Shape Benchmark* was proposed by Shilane et al. [2004]. The benchmark consists of a train database, which is proposed for calibrating search algorithms, and a test database, which can then be used to compare different search engines against each other. This benchmark could eventually become the standard in evaluating and comparing retrieval performance of 3D retrieval algorithms in the future.

3. METHODS FOR CONTENT-BASED 3D RETRIEVAL

This section reviews recent methods for feature-based retrieval of 3D objects. In Section 3.1 an overview and a classification of the different methods discussed in this survey is given. In Sections 3.2 – 3.7 we give a detailed description of many individual methods, sorted according to our classification.

3.1 Overview and classification

Classifying methods for 3D description can be done along different criteria. A popular differentiation from the field of shape analysis is according to the following schema [Loncaric 1998]:

—Descriptors can be built based on the *surface* of an object, or based on *interior* properties. Curvature of the boundary is an example of the first type of descriptor, while measures for the distribution of object mass are of the second type of description.

—Depending on the type of resulting object descriptor, *numeric* methods produce a vector of scalar values representing the object, while *spatial* methods use other means, e.g., a sequence of primitive shapes approximating the original shape, or a graph representing object structure.

—*Preserving* descriptors preserve the complete object information, which allows the lossless reconstruction of the original object from the description. *Non-preserving* descriptors discard a certain amount of object information, usually retaining only some part of information that is considered the most important.

A descriptor differentiation more specific to 3D models can be done based on the type of model information focused on, e.g., geometry, color, texture, mass distri-

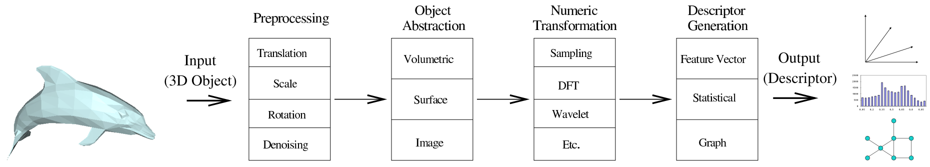


Fig. 5. 3D descriptor extraction process model.

bution, and material properties. Usually, geometry is regarded most important for 3D objects, and thus, all descriptors presented in this survey make use of geometry input only. This is also because geometry is always specified in models, while other characteristics are more application-dependent and cannot be assumed to be present in arbitrary 3D databases.

Furthermore, one could differentiate descriptors with respect to integrity constraints assumed for the models, e.g., solid shape property, consistent face orientation, or the input type assumed (polygon mesh, voxelization, CSG set, etc.). Most of the presented methods are flexible in that they allow for model inconsistencies and assume triangulations. Of course, the description flexibility depends on the model assumptions; additional information can be expected to yield more options for designing descriptors.

Recently, we proposed a new way to classify methods for 3D model retrieval [Bustos et al. 2005]. In this classification, the extraction of shape descriptors is be regarded as a multistage process (see Figure 5). In the process, a given 3D object, usually represented by a polygonal mesh, is first preprocessed to approximate the required invariance and robustness properties. Then, the object is abstracted 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 may take place, from the result of which finally the FVs are extracted.

We briefly sketch these basic steps in the following. Without losing generality, we assume that the 3D object is represented by a polygonal mesh.

(1) *Preprocessing.* If required by the descriptor, the 3D model is preprocessed for rotation (R), translation (T), and/or scaling (S) invariance.

(2) *Type of object abstraction.* There are three different types of object abstraction: *Surface*, *volumetric*, and *image*. Statistics of the curvature of the object surface is an example of a descriptor based directly on 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 characteristics 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. This forms image-based object abstractions.

(3) *Numerical transformation.* The main features of the polygonal mesh may be captured numerically using different methods. E.g., voxels grids and image arrays can be Wavelet transformed, or surfaces can be adaptively sampled. Other numerical transformations include spherical harmonics (SH), curve fitting, and the discrete Fourier transform (DFT). Such transforms yield a numerical representation of the underlying object.

(4) *Descriptor generation.* At this stage, the final descriptor is generated. It can belong to one of the next three classes:
   (a) *Feature vectors* (FV) 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.
   (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. Usually, the obtained histogram is represented as a FV, where each coordinate value correspond to a bin of the histogram.
   (c) The third category is better suited for structural 3D object shape description that can be represented in the form of a *graph* [Sundar et al. 2003; Hilaga et al. 2001]. A graph can more easily represent the structure of an object that is made up of or can be decomposed into several meaningful parts, such as the body and the limbs of objects modeling animals.

Table I shows the algorithms surveyed in this paper with their references, preprocessing steps employed, type of object abstraction considered, numeric transform applied, and descriptor type obtained.

For presentation in this survey, we have organized the descriptors to the following Subsections:

—Statistics (Section 3.2). Statistical descriptors reflect basic object properties like the number of vertices and polygons, the surface area, the volume, the bounding volume, and statistical moments. A variety of statistical descriptors are proposed in the literature for 3D retrieval. In some application domains, simple spatial extension or volumetric measures may already be enough to retrieve objects of interest.

—Extension based descriptors (Section 3.3). Extension based methods build object descriptors from features sampled along certain spatial directions from an objects center.

—Volume-based descriptors (Section 3.4). These methods derive object features from volumetric representations obtained by discretizing object surface into voxel grids, or by relying on the models being already given in volumetric representation.

—Surface geometry (Section 3.5). These descriptors focus on characteristics derived from model surface.

—Image based descriptors (Section 3.6). The 3D similarity problem may be reduced to an image similarity problem by comparing 2D projections rendered from the 3D models.

While this survey focuses on FV-based descriptors, we recognize there exists a rich body of work from computer vision and shape analysis which deals with advanced 3D shape descriptors relying on structural shape analysis and customized data structures and distance functions. In principle, these can also be used to implement similarity search algorithms for 3D objects. Therefore, in Section 3.7 we

Table I. Overview of the surveyed methods.

Descriptor Name	Sect.	Prepr.	Obj. abs.	Num. transf.	Type
Simple statistics	3.2.1	RTS	Volum.	None	FV
Parametrized stat.	3.2.2	RTS	Surface	Sampling	FV
Geometric 3D moments	3.2.3	RTS	Surface	Sampling	FV
Ray moments	3.2.3	RTS	Surface	Sampling	FV
Shape distr. (D2)	3.2.4	None	Surface	Sampling	Hist.
Cords based	3.2.5	RT	Surface	Sampling	Hist.
Ray based w. SH	3.3.1	RTS	Image	Sampl.+SH	FV
Shading w. SH	3.3.1	RTS	Image	Sampl.+SH	FV
Complex w. SH	3.3.1	RTS	Image	Sampl.+SH	FV
Ext. to ray based	3.3.2	RTS	Image	Sampl.+SH	FV
Shape histograms	3.4.2	RTS	Volum.	Sampling	Hist.
Rot. inv. point cloud	3.4.3	RTS	Volum.	Sampling	Hist.
Voxel	3.4.4	RTS	Volum.	None	Hist.
3DDFT	3.4.4	RTS	Volum.	3D DFT	FV
Voxelized volume	3.4.5	RTS	Volum.	Wavelet	FV
Volume	3.4.5	RTS	Volum.	None	FV
Cover sequence	3.4.5	RTS	Volum.	None	FV
Rot. inv. sph. harm.	3.4.6	TS	Volum.	Sampl.+SH	FV
Reflective symmetry	3.4.7	TS	Volum.	Sampling	FV
Weighted point sets	3.4.8	RTS	Volum.	None	Hist.
Surface normal direct.	3.5.1	None	Surface	None	Hist.
Shape spectrum	3.5.2	None	Surface	Curve fitting	Hist.
Ext. Gaussian image	3.5.3	R	Surface	None	Hist.
Shape based on 3DHT	3.5.4	None	Surface	Sampling	FV
Silhouette	3.6.1	RTS	Image	Sampl.+DFT	FV
Depth Buffer	3.6.2	RTS	Image	2D DFT	FV
Lightfield	3.6.3	TS	Image	DFT, Zernike	FV
Topological Matching	3.7.1	None	Surface	Sampling	Graph
Skeletonization	3.7.2	None	Volumetric	Dist. transf., clustering	Graph
Spin Image	3.7.3	None	Surface	Binning	2D Hist.

exemplarily recall 3D matching approaches based on topological graphs, skeleton graphs, and a customized data structure built for each point in a 3D image (or model).

Figure 6 summarizes the chosen organization of the methods surveyed in this paper. The remainder of this section follows this organization.

3.2 Statistical 3D descriptors

3.2.1 *Simple statistics.* Bounding volume, object orientation and object volume density descriptors are probably the most basic shape descriptors, and are widely used in the CAD domain. In Paquet et al. [2000] the authors review several possible simple shape descriptors. The bounding volume (BV) is given by the volume of the minimal rectangular box that encloses a 3D object. The orientation of this bounding box is usually specified parallel to either the coordinate frame, or parallel to the principal axes of the respective object. Also, the occupancy fraction of the object within its bounding volume gives information on how “solid” respectively “rectangular” the object is. Having determined the principal axes, it is also possible to integrate orientation information in the description, relating the principal axes

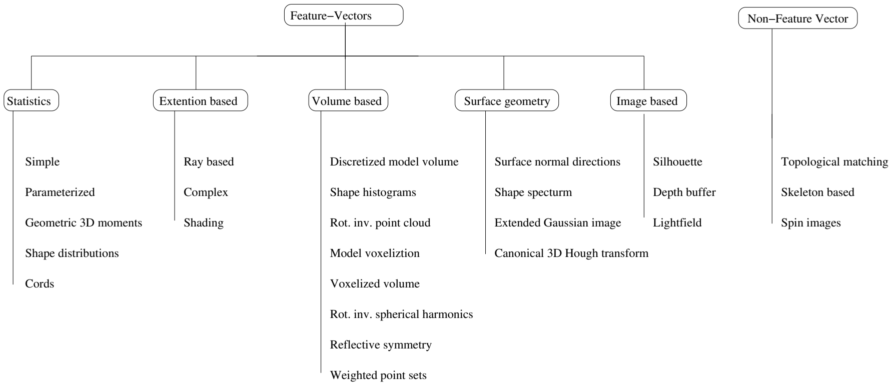


Fig. 6. Organization of 3D retrieval methods in this survey.

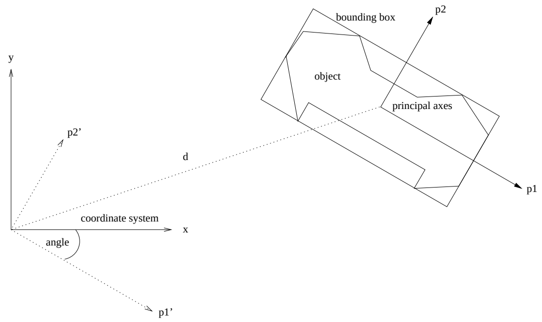


Fig. 7. Principal axes-based bounding volume and orientation of an object with respect to the original coordinate system (2D illustration).

to the given world coordinates of the object. Here, it is proposed to consider the distance between the bounding volume's center from the origin of the coordinate system, as well as the angle enclosed between the principal axes and the coordinate system. If only the bounding volume is considered, this descriptor is invariant with respect to translation. If the bounding volume is determined edge-parallel the object's principal axes, it is also approximately invariant with respect to rotation. In both variants, the bounding volume descriptor is not invariant with respect to the object's scale. Figure 7 illustrates.

3.2.2 *Parameterized statistics.* Ohbuchi et al. [2002] propose a statistical feature vector which is composed of three measures taken from the partitioning of a model into "slices" orthogonal to its three principal axes. The FV consists of $3 * 3 * (n - 1)$ components, where n is the number of equally-sized bins along the principal axes. A sampling window is moved along the axes that considers the average measures from consecutive pairs of adjacent slides, obtaining $n - 1$ values on each principal axis for each of the three proposed measures (see Figure 8). The measures used are the moment of inertia of the surface points, the average distance of surface points

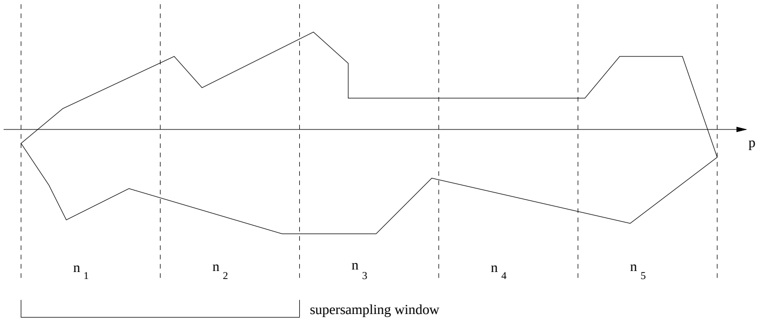


Fig. 8. Discretization of a model into 5 equally-sized “slices”, yielding 4 descriptor components.

from the principal axis, and as the variance in this distance. Selection of object points for PCA and statistical measure calculation is done by randomly sampling a number of points from the object’s faces (assuming a polygonal mesh), keeping the number of points in each face proportional to its area. For retrieval, the authors experiment with the standard Euclidean distance, as well as with a custom distance called “elastic distance”, which allows for some shift in the bins to be compared [Ohbuchi et al. 2002]. Both metrics are shown to produce similar results. The authors conduct experiments on a VRML object database and conclude that their descriptor is suited well for objects that possess rotational symmetry, like, e.g., chess figures. A sensitivity analysis indicates that there exists some optimal choice for the number of analysis windows, given a number of total sampling points.

3.2.3 *Geometric 3D moments.* The usage of moments as a means of description has a tradition in image retrieval and classification. Thus, moments have been used in some of the first attempts to define feature vectors also for 3D object retrieval. Statistical moments μ are scalar values that describe a distribution f. Parametrized by their order, moments represent a spectrum from coarse-level to detailed information of the given distribution [Paquet et al. 2000]. In the case of 3D objects, an object may be regarded as a distribution $f(x, y, z) \in \mathbb{R}^3$, and the moment $\mu_{i,j,k}$ of order $n = i + j + k$ in continuous form can be given as:

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

As is well known, the complete (infinite) set of moments uniquely describes a distribution and vice versa. In its discrete form, objects are taken as finite point sets P in 3D, and the moment formula becomes $\mu_{ijk} = \sum_{p=1}^{|P|} {x_p}^i {y_p}^j {z_p}^k$. Because moments are not invariant with respect to translation, rotation, and scale of the considered distribution, appropriate normalization should be applied before moment calculation. When given as a polygon mesh, candidates for input to moment calculation are the mesh vertices, the centers of mass of triangles, or other object points sampled by some scheme. A FV can then be constructed by concatenating several moments, e.g., all moments of order up to some n.

Studies that employ moments as descriptors for 3D retrieval include Vranić and

Saupe [2001a], where moments are calculated for object points sampled uniformly with a ray-based scheme (see Section 3.3.1), and Paquet et al. [2000], where moments are calculated from the centers of mass (centroids) of all object faces (see Section 3.2.5). Vranić and Saupe [2001a] compare the retrieval performance of ray-based with centroid-based moments, and conclude that the former are more effective. Another publication that proposed the usage of moments for 3D retrieval is Elad et al. [2002]. Here, the authors uniformly sample a certain number of points from the object's surface for moment calculation. Special to their analysis is the usage of relevance feedback to adjust the distance function employed on their moment-based descriptor. While in most systems a static distance function is employed, here it is proposed to interactively adapt the metric. A user performs an initial query using a feature vector of several moments under the Euclidean norm. She marks relevant and irrelevant objects in a prefix of the complete ranking. Then, via solving a quadratic optimization problem, weights are calculated that reflect the feedback so that in the new ranking using the weighted Euclidean distance, relevant and irrelevant objects (according to the user input) are discriminated by a fixed distance threshold. The user is allowed to iterate through this process, until a satisfactory end result is obtained. The authors conclude that this process is suited to improve search effectiveness.

3.2.4 *Shape distribution.* Osada et al. [2002] propose 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). Depending on the shape function employed, shape distributions possess rigid transformation invariance, robustness against small model distortions, independence of object representation, and 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 certain fixed point 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 formed by triples of random surface points, or the cube root of the volume of the tetrahedron between four random points on the surface. Where necessary, a normalization step is to be applied for differences in scale.

Generally, the analytic computation of distributions is not feasible. Thus, the authors perform random point sampling of an object, and construct a histogram to represent a shape distribution. Retrieval experiments yielded that the best results were achieved using the *D2* distance function (distance between pairs of points on the surface, see also figure 9) and using the L_1 norm of the probability density histograms, which were normalized by aligning the mean of each two histograms to be compared.

Shape distributions for 3D retrieval have further been explored in Ip et al. [2002], Ip et al. [2003], and Ohbuchi et al. [2003].

3.2.5 *Cords-based descriptor.* Paquet et al. [2000] present a descriptor that combines information about the spatial extent and orientation of a 3D object. The

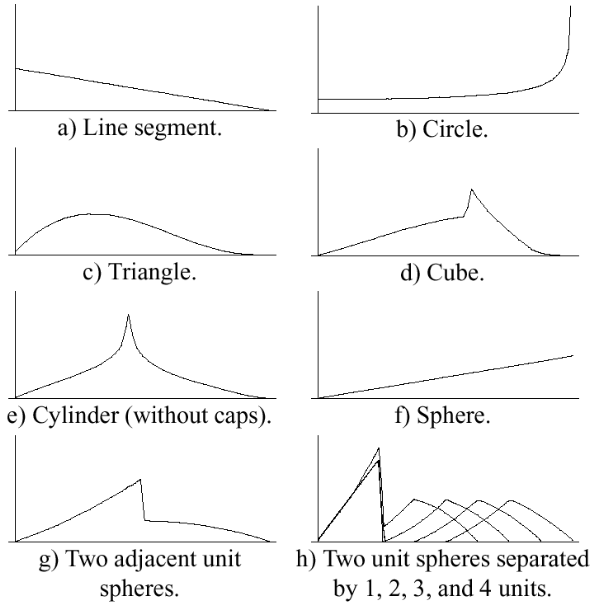


Fig. 9. D2 distance histograms for some example objects. (Figure taken from Osada et al. [2002] (© 2002 ACM Press). Copyright is held by the owner.)

authors define a "cord" as a vector that runs from an object's center of mass to the centroid of a face of the object. For all object faces, such a cord is constructed. The descriptor consists of three histograms, namely for the angles between the cords and the object's first two principal axes, and for the distribution of the cord length, measuring spatial extension. The three histograms are normalized by the number of cords. Using the principal axes as reference, the descriptor is 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.

3.3 Extension-based descriptors

3.3.1 *Ray-based sampling with spherical harmonics representation.* Vranić and Saupe [2001a;2002] propose a descriptor framework that is based on taking samples from a PCA-normalized 3D object by probing the polygonal mesh along regularly spaced directional unit vectors $\mathbf{u}_{ij}$ as defined and visualized in Figure 10. The samples can be regarded as values of a function on a sphere ($||\mathbf{u}_{ij}|| = 1$). The so-called *ray-based* feature vector measures the extent of the object from its center of gravity O in directions $\mathbf{u}_{ij}$. The extent $r(\mathbf{u}_{ij}) = ||P(\mathbf{u}_{ij}) - O||$ in direction $\mathbf{u}_{ij}$ is determined by finding the furthest intersection point $P(\mathbf{u}_{ij})$ between the mesh and the ray emitted from the origin O in the direction $\mathbf{u}_{ij}$. If the mesh is not intersected by the ray, then the extent is set to zero, $r(\mathbf{u}_{ij}) = 0$. The number of samples, $4B^2$ (Figure 10), should be sufficiently large (e.g., $B \geq 32$) so that sufficient information about the object may be captured. The obtained samples can be considered as

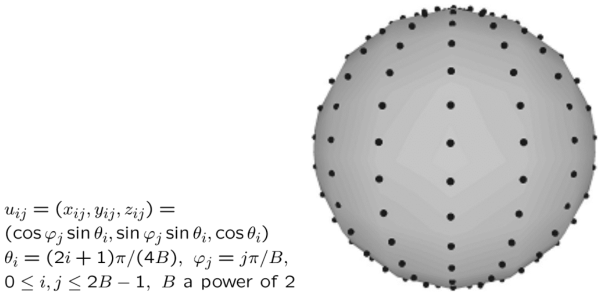

$u_{ij} = (x_{ij}, y_{ij}, z_{ij}) =$
$(\cos\varphi_j \sin\theta_i, \sin\varphi_j \sin\theta_i, \cos\theta_i)$
$\theta_i = (2i+1)\pi/(4B),\ \varphi_j = j\pi/B,$
$0 \leq i, j \leq 2B-1,\ B$ a power of 2

Fig. 10. Determining ray directions u by uniformly varying spherical angular coordinates θ_i and φ_j.

components of a feature vector in the spatial domain. A similar FV called "Sphere Projection" was considered by Leifman et al. [2003], which includes a number of empirical studies, showing good performance with respect to to a ground truth database of VRML models collected from the Internet.

Nonetheless, such a descriptor consists of a large dimensionality. In order to characterize many samples of a function on a sphere by just a few parameters, *spherical harmonics* [Healy et al. 2003] are proposed as a suitable tool. The magnitudes of complex coefficients, which are obtained by applying the fast Fourier transform on the sphere (SFFT) to the samples, are regarded as vector components. Thus, the ray-based feature vector is represented in the spectral domain, where each vector component is formed by taking into account all original samples. Having in mind that the extent function is a real-valued function, the magnitudes of the obtained coefficients are pairwise equal. Therefore, vector components are formed by using magnitudes of non-symmetric coefficients. Also, an embedded multi-resolution representation of the feature can easily be provided. A useful property of the ray-based FV with spherical harmonic representation is invariance with respect to rotation around the z-axis (when the samples are taken as depicted in Figure 10). The inverse SFFT can be applied to a number of the spherical harmonic coefficients to reconstruct an approximation of the underlying object at different levels, see Figure 11. Besides considering the extent as a feature aimed at describing 3D-shape, the authors consider a rendered perspective projection of the object on an enclosing sphere. The scalar product $x(\mathbf{u}) = |\mathbf{u} \cdot \mathbf{n}(\mathbf{u})|$, where $\mathbf{n}(\mathbf{u})$ is the normal vector of the polygon that contains the point $O + r(\mathbf{u})\mathbf{u}$ (if $r(\mathbf{u}) > 0$), can be regarded as information about shading at the point (θ, φ) on the enclosing sphere. A *shading-based* FV is generated analogously to the ray-based FV, by sampling the shading function, applying the SFFT, and taking the magnitudes of low-frequency coefficients as vector components. In extension to using either $r(u)$ or $x(u)$, also the combination of both measures in a complex function $y(u) = r(u) + i \cdot x(u)$ 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 the ray-based and the shading-based FVs.

3.3.2 *Extensions for ray-based sampling.* Vranić [2003] further explores an improvement of the ray-based methods described above. Particularly, the author

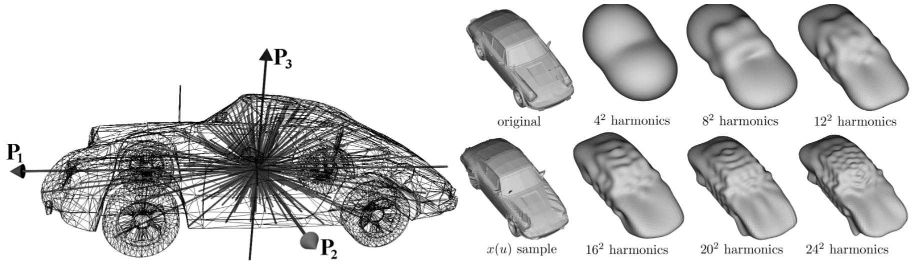


Fig. 11. Ray based feature vector (left). The right illustration shows the back-transform of the ray-based samples from frequency to spatial domain.

proposes to not restrict the sampling at each ray to the last intersection point with the mesh, but also to consider "interior" intersection points of the ray with model surface. This is implemented by using concentric spheres centered at the model origin and with uniformly varying radii, and associating all intersection points between rays and the mesh to the closest sphere each. For each ray and each sphere radius, the largest distance between the origin and the intersection points associated with the respective ray and radius is set as the sampling value, if such a point exists (otherwise, the respective sampling value is set to zero). The author thereby obtains samples of functions on multiple concentric spheres. He defines two FVs by applying the spherical Fourier transform on these samples, and extracting FV components from either the energy contained in certain low frequency bands (*RH1* FV) as done in the approach by Funkhouser et al. [2003] and described in Section 3.4.6, or from the magnitudes of certain low frequency Fourier coefficients (*RH2* FV). While RH2 relies on the PCA for pose estimation and includes orientation information, RH1 is rotation invariant by definition, discarding orientation information. The author experimentally evaluates the retrieval quality of these two descriptors against the ray-based FV in spherical harmonics representation described above, and against the FV defined by Funkhouser et al. [2003]. From the results the author concludes that (1) RH1 outperforms the implicitly rotation invariant FV based on the spherical harmonics representation of a model voxelization (see Section 3.4.6), implying that the SFT is effective in filtering high frequency noise and (2) that RH2 and the ray based FV, both relying on PCA, outperform the other two FVs, implying that including orientation information using the PCA in FVs may positively affect object retrieval on average. As a further conclusion, the author states that RH2 performs slightly better that the ray-based FV, implying that considering interior model information can increase retrieval effectiveness.

3.4 Volume-based descriptors

3.4.1 *Discretized model volume.* A class encompassing several 3D descriptors that are all derived from some form of volumetric discretization of the models is reviewed in the following. Here, the basic idea is to construct a feature vector from a model by partitioning the space in which it resides, and then aggregating the model content that is located in the respective partitioning segments to form the components of feature vectors. Unless otherwise stated, these descriptors rely

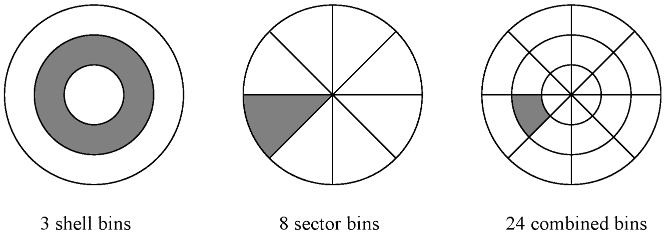


Fig. 12. Shells and sectors as basic space decompositions for shape histograms. In each of the 2D examples, a single bin is marked.

on model normalization, usually with PCA methods, to approximately provide comparability between the spatial partitions of all models.

3.4.2 *Shape histograms.* Ankerst et al. [1999a] studied classification and similarity search of 3D objects modeled as point clouds. They describe 3D object shapes as histograms of point fractions that fall into partitions of the enclosing object space under different partitioning models. One decomposition is the *shell model*, which partitions the space into shells concentric to the object's center of mass, keeping radii intervals constant. The *sector model* decomposition uses equally-sized segments obtained by forming Voronoi partitions around rays emitted from the model origin and pointing to the vertices of an enclosing regular polyhedron. Finally, a *combined model* uses the intersection of shells and sectors, see Figure 12 for an illustration. While the shell model is inherently rotation invariant, the sector and the combined models rely on rotational object normalization. The authors propose the quadratic form distance for similarity estimation in order to reflect cross-similarities between histogram bins. Experiments are conducted in a molecular classification setup, and good discrimination capabilities are reported for the high-dimensional sector (122-dim) and combination (240-dim) models, respectively.

3.4.3 *Rotation invariant point cloud descriptor.* Kato et al. [2000] present a descriptor that relies on PCA registration but at the same time is invariant to rotations of 90 degrees along the principal axes. To construct the descriptor, an object is placed and oriented into the canonical coordinate frame using PCA, and scaled to fit into a unit cube with origin at the center of mass of the object and perpendicular to the principal axes. The unit cube is then partitioned into $7 \times 7 \times 7$ equally sized cube cells, and for each cell, the frequency of points regularly sampled from the object surface and which lie in the respective cell, is determined. To reduce the size of the descriptor, which until now consists of 343 values, all grid cells are associated with one of 21 equivalence classes based on their location in the grid. To this end, all cells that coincide when performing arbitrary rotations of 90 degrees along the principal axes are grouped together in one of the classes (see Figure 13 for an illustration). For each equivalence class, the frequency data contained in the cells belonging to the respective equivalence class is aggregated, and the final descriptor of dimensionality 21 is obtained. The authors presented retrieval performance results on a 3D database, on which $7 \times 7 \times 7$ was found to be the best grid dimensionality, but state that in general the optimal size of the

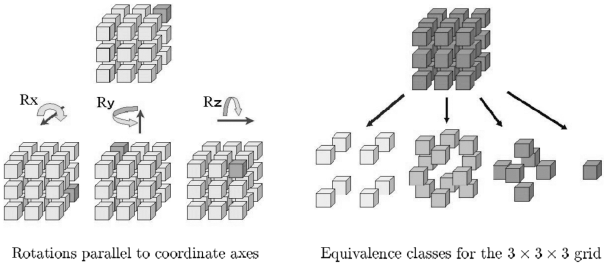

Rx
Ry
Rz
Rotations parallel to coordinate axes
Equivalence classes for the 3 × 3 × 3 grid

Fig. 13. Aggregating object geometry in equivalence classes defined on a $3 \times 3 \times 3$ grid. (Figure taken from Kato et al. [2000] (© 2000 IEEE). Copyright is held by the owner.)

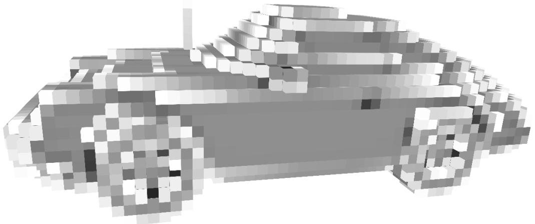


Fig. 14. The voxel-based feature vector compares occupancy fractions of voxelized models in the spatial or frequency domain.

descriptor depends on the specific database characteristics.

3.4.4 *Model voxelization.* Vranić and Saupe [2001b] present a FV based on the rasterization of a model into a voxel grid structure, and experimentally evaluate the representation of this FV in both the spatial and the frequency domain. The voxel descriptor is obtained by first subdividing the bounding cube of an object (after PCA-based rotation 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 of the model. In order to compute the value of S_{ijk}, each model triangle T_i $(i = 1, \ldots, m)$ is subdivided into l_i^2 $(l \in \mathbb{N})$ coincident triangles, where the value of l_i^2 is proportional to the area of T. The value of S_{T_i}/l_i^2 (S_{T_i} is the area of triangle T_i) is the attribute of the centers of gravity of the triangles obtained by the subdivision. Finally, the value of S_{ijk} is approximated by summing up attributes of centroids lying in the corresponding voxel cell. The object's voxel cell occupancies constitute the FV of dimension n^3. For similarity estimation with this FV, a metric can be defined in the spatial domain (voxel), or after a 3D Fourier-transform in the frequency domain (3DDFT). Then, magnitudes of certain k low-frequency coefficients are used for description, enabling multi-resolution search.

Using octrees for 3D similarity search was also recently proposed by Leifman et al. [2003], where the similarity of two objects is given by the sum of occupancy differences for each non-empty cell pair of the voxel structure. The authors report good retrieval capabilities of this descriptor.

3.4.5 *Voxelized volume.* In the preceding section, an object was considered as a collection of 2D-polygons, i.e., as a surface in 3D. This approach is the most general applying to unstructured "polygon soups". In the case of polygons giving rise to a watertight model, one may want to use the enclosed volume to derive shape descriptors. Such schemes require an additional preprocessing step after pose normalization, namely the computation of a 3D bitmap that specifies the inside/outside relation of each voxel with respect to the enclosed volume of the polygonal model. Several methods for similarity estimation based on voxelized volume data of normalized models have been proposed. Paquet et al. [2000] and Paquet and Rioux [2000] propose a descriptor that characterizes voxelized models by statistical moments calculated at different levels of resolution of the voxel grid, and where the different resolutions are obtained by applying the Daubechies-4 wavelet transform on the three-dimensional grid. Keim [1999] describes a similarity measure based on the amount of intersection between the volumes of two voxelized 3D objects.

Novotni and Klein [2001b] proposed to use the minimum of the symmetric volume differences between two solid objects obtained when considering different object alignments based on principal axes, in order to measure volume similarity. The authors also give a technique that supports the efficient calculation of symmetric volume differences based on the discretization of volumes into a grid. Sánchez-Cruz and Bribiesca [2003] report a scheme for optimum voxel-based transformation from one object into another one, which can be employed as a measure of object dissimilarity.

Another volume based FV is presented in Heczko et al. [2002]. In order to remove the topological requirement of a watertight model the volume of a given 3D-model specified by a collection of polygons in defined in a different way. Each polygon contributes a (signed) volume given by the tetrahedron that is formed by considering the center of mass of all polygons as a vertex for a polyhedron with the given polygon as a base face. The sign is chosen according to the normal vector for the polygon given by the model. The space surrounding the 3D models is partitioned into sectors similar to the method in Section 3.4.2 and in each sector the (signed) volumes of the intersection with a generated polyhedra is accumulated and gives one component of the FV. The partitioning scheme is as follows. 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 is obtained. For similarity search either the volumes occupied in each segment, or a number of k first coefficients after a Fourier transform is considered. Figure 15 illustrates the idea in a 2D sketch. Experimental results with this descriptor are presented in Section 4. It performs rather poorly, which may be attributed to the fact that the used retrieval database does not guarantee consistent orientation of the polygons.

Kriegel et al. [2003] present another approach for describing voxelized objects. The *cover sequence model* approximates a voxelized 3D object using a sequence of *grid primitives* (called *covers*), which are basically large parallelepipeds. The quality of a cover sequence is measured as the symmetric volume difference between the original voxelized object and the sequence of grid primitives. The sequence is

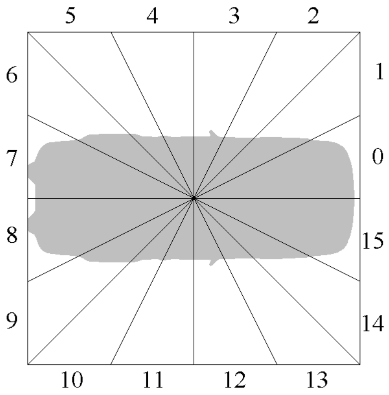


Fig. 15. Volume based feature vector.

described as the set of unions or differences of the covers, and then each cover of the sequence contributes with six values for the final descriptor (three values for describing the position of the cover, and three values for describing the extension of the cover). The main problem is the ordering of the covers in the sequence: Two voxelized objects that are similar may produce features that are distant away, depending on the ordering of the covers. To overcome this problem, the authors propose to use sets of FVs (one FV for each cover) to describe the 3D objects, and to compute a similarity measure between two sets of FVs that ensures that the optimal sequence of covers, the one that produces the minimal distance between the two objects, will be always considered.

3.4.6 *Rotation invariant spherical harmonics descriptor.* Funkhouser et al. [2003] propose a descriptor based also on the spherical harmonics representation of object samples. The main difference between this approach and the one reported in Section 3.3.1, apart from the sampling function chosen, is that by descriptor design it provides rotation invariance without requiring pose estimation. This is possible since the energy in each frequency band of the spherical transform is rotation invariant [Healy et al. 2003].

Input to their transform is the binary voxelization of a polygon mesh into a grid with dimension $2R * 2R * 2R$, where each occupied voxel indicates the intersection of the mesh with the respective voxel. To construct the voxelization, the object's center of mass is translated into grid position (R, R, R) (grid origin), and the object is scaled so that the average distance of occupied voxels to the center of mass amounts to $\frac{R}{2}$, that is $\frac{1}{4}$ of the grids edge length. By using this scale instead of scaling it so that the bounding cube fits into the grid, it is possible to lose some object geometry in the description. On the other hand, sensitivity with respect to outliers is expected to be reduced. The $8R^3$ voxels give rise to a binary function on the corresponding cube, which is written in spherical coordinates as $f_r(\theta, \phi)$ with the origin $(r = 0)$ placed at the cube center. The binary function is sampled for radii $r = 1, 2, \ldots, R$ and sufficiently many angles θ, ϕ to allow computation of the spherical harmonics representation of the spherical functions f_r. The feature vector consists of low frequency band energies of the functions $f_r, r = 1, \ldots, R$. By

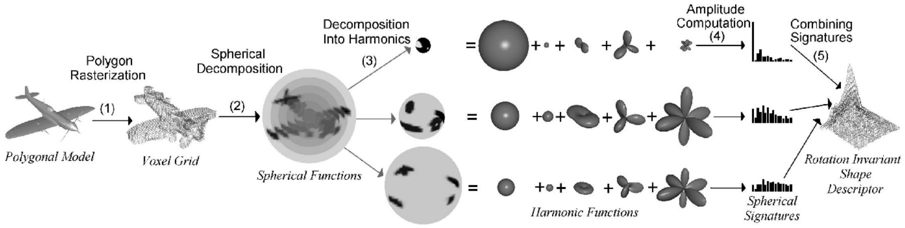


Fig. 16. Descriptor extraction process for the harmonics 3D descriptor. (Figure taken from Funkhouser et al. [2003] (© 2003 ACM Press). Copyright is held by the owner.)

construction it is invariant with respect to rotation about the center of mass of the object.

The authors presented experimental results conducted on a database of 1,890 models that were manually classified, comparing the harmonics 3D descriptor with the shape histogram (see Section 3.4.2), shape distribution D2 (see Section 3.2.4), EGI (see Section 3.5.3), and a moment based (see Section 3.2.3) descriptor. The experiments indicated that their descriptor consistently outperformed the other descriptors, which among other arguments was attributed to discarding rotation information.

A generalization of this approach considering the full volumetric model information was introduced in Novotni and Klein [2003;2004]. The authors form rotational-invariant descriptors from 3D Zernike moments obtained from appropriately voxelized mesh models. The authors present the mathematical framework and discuss implementation specifics. From analysis and experiments, they concluded that by considering the integral of the volumetric information in extension to just sampling it on concentric spheres, retrieval performance may be improved, and at the same time a more compact descriptor is obtained.

3.4.7 *Reflective symmetry descriptor.* Kazhdan et al. [2003] present a descriptor that is based on global object symmetry. The method is based on a function on the unit sphere that measures reflective symmetry between two parts of an object lying on the opposite sides of a cutting plane. The cutting plane contains the center of gravity of the object, while the normal vector is determined by a point on the unit sphere. The main idea of the approach is that the shape of an object may be characterized by using an appropriate measure of symmetry and by sampling the corresponding function at sufficiently many points. Briefly, for a given cutting plane, the reflective symmetry is computed using a function f on concentric spheres. The function f is defined by sampling a voxel-based representation of the object. The voxel attributes are defined using an exponentially decaying Euclidean distance transform. The reflective symmetry measure describes the proportion of f that is symmetric with respect to the given plane and the proportion of f that is anti-symmetric. The symmetric proportion is obtained by projecting the function f onto the space π of functions invariant under reflection about the given plane and by computing the L_2 distance between the original function and the projection. The anti-symmetric proportion is calculated in a similar manner, by projecting f onto

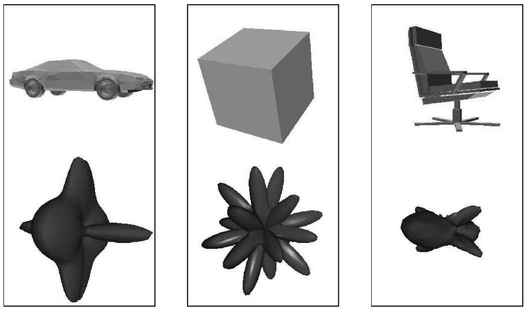


Fig. 17. Visualization of the reflective symmetry measure (lower row) for certain objects (upper row). The extension in direction u indicates the degree of symmetry with respect to the symmetry plane perpendicular to u. (Figure taken from Kazhdan et al. [2003] (© 2003 Springer-Verlag) with kind permission of Springer Science and Business Media. Copyright is held by the owner.)

the space $\pi\perp$ that is orthogonal to π and by computing the L_2 distance between f and the projection. Since the reflective symmetry measure is determined by analyzing the whole object, a sample value of the function on the unit sphere gives information about global symmetry with respect to a given plane.

For 3D retrieval, the similarity between objects is estimated by the L_∞ norm between their reflective symmetry descriptions. Analytically and experimentally, the authors show the main properties of this approach to be stability against high-frequency noise, scale invariance and robustness against the level of detail of the object representation. Considering retrieval power, the algorithm is experimentally shown to be "orthogonal" to some extent to other descriptors, in the sense that overall retrieval power is increased when combining it with other descriptors. Figure 17 shows an example of reflective symmetry measures for several objects.

3.4.8 *Point sets methods.* The *weighted point set* method [Tangelder and Veltkamp 2003] compares two 3D objects represented by polyhedral meshes. A shape signature of the 3D object is defined as a set of points that consists of weighted salient points from the object. The first step of the algorithm is to place the 3D object into a canonical coordinate system, which is established by means of the PCA, and to partition the object's bounding cube into a rectangular voxel grid. Then, for each nonempty grid cell one representative vertex as well as an associated weight are determined by either one out of three methods explored by the authors. In the first proposed method, for each nonempty cell one of the contained points is selected based on the Gaussian curvature at the respective point, and associating the Gaussian curvature with that point. The two other proposed methods average over the vertices of a cell, and associate either a measure for the normal variation, or the unit weight. The latter method is given in order to be able to support meshes with inconsistent polygon orientation, because then curvature and normal variation cannot be determined meaningfully. A variation of the Earth's Mover Distance (EMD)

[Rubner et al. 1998] [1], the so-called *proportional transportation distance* (PTD), is introduced as the similarity function to compare two weighted point sets. The PTD is described as a linear programming problem that can be solved, for example, using the simplex algorithm. The authors state that the PTD is a pseudo-metric, which makes it suitable to use for indexing purposes (in contrast, the EMD does not obey the triangle inequality). Experiments were performed which indicate competitive retrieval performance, while no clear winner could be identified among the three proposed weighing methods.

Another approach that matches sets of points was introduced in Shamir et al. [2003]. There, the point sets to be matched are obtained for each 3D model by decomposing the model into a coarse-to-fine hierarchy of an elementary shape (spheres). The point sets therefore consists of sphere radii and associated centers, and can be matched by a custom coarse-to-fine algorithm involving exhaustive search on a coarse level, and graph matching techniques on finer levels in the multiresolution representation.

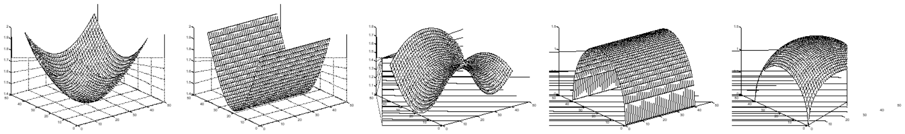
3.5 Surface-geometry based descriptors

In this Section, we present 3D descriptors that are based on object surface measures. These surface measures include surface curvature measures, as well as the distribution of surface normal vectors.

3.5.1 *Surface normal directions.* Paquet and Rioux [2000] consider histograms of the angles enclosed between the first two principal axes each and the face normals of all object polygons. Straightforward, it is possible to construct either one unifying histogram, two separate histograms for the distribution with respect to the each of the two first principal axes, or a bivariate histogram which reflects the dependency between the angles. Intuitively, the bidimensional distribution contains the most information. Still, such histograms are sensitive to the level of detail by which the model is represented. An illustrating example is given by the authors. Considering two pyramids where the sides of one of them is formed by inclined planes, and for the other is formed by a stairway-like makeup. Obviously, the angular distributions of the two pyramids will differ tremendously, while their global shape might be quite similar.

3.5.2 *Surface curvature.* Zaharia and Prêteux [2001] present a descriptor for 3D retrieval proposed within the MPEG-7 framework for multimedia content description. The descriptor reflects curvature properties of 3D objects. The *shape spectrum FV* is defined as the distribution of the *shape index* for points on the surface of a 3D object, which is a function of the two principal curvatures. The shape index is a scaled version of the angular coordinate of a polar representation of the principal curvature vector, and it is invariant with respect to rotation, translation and scale by construction. Figure 18 illustrates some elementary shapes with their corresponding shape index values. 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

[1] The basic idea of the earth movers distance is to measure the distance between two histograms by solving the transportation problem of converting one histogram into the other.



a. Spherical cup (0.0) b. Rut (0.25) c. Minimal Saddle (0.5) d. Ridge (0.75) e. Spherical cap (1.0)

Fig. 18. Shape index values for some elementary shapes. (Figure taken from Zaharia and Prêteux [2001] (© 2001 SPIE). Copyright is held by the owner.).

to all mesh faces based on the respective face and 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 the relative area of the sum of such problematic surface regions is accumulated in two additional attributes named *planar surface* and *singular surface*, respectively. These attributes together with the shape index histogram form the final descriptor. Experiments conducted by the authors with this descriptor on several 3D databases quantitatively show good retrieval results.

Surface curvature as a description for 3D models was also considered by Shum et al. [1996]. In this work, the models were resampled by fitting a regularly tessellated spherical polygon mesh onto the model. Then, the curvature was determined for each vertex of the fitted spherical mesh, based on the vertex' neighbor nodes. Finally, the similarity measure between two models was obtained by minimizing an l_p norm between their curvature maps over all rotations of the map, thereby supporting rotation invariance.

3.5.3 *Extended Gaussian Image.* The distribution of the normals of the polygons that form a 3D object can be used to describe its global shape. One way to represent this distribution is using the *Extended Gaussian Image* (EGI) [Horn 1984; Ip and Wong 2002]. The EGI is a mapping from the 3D object to the Gaussian sphere. To compute the EGI of a 3D object, the normal vectors of all polygons of the 3D objects are mapped onto the respective point of the Gaussian sphere that has the same normal as the polygon. To build a descriptor from this mapping, the Gaussian sphere is partitioned into $R \times C$ cells (by using R different longitudes and $C-1$ different latitudes), where each cell corresponds to a range of normal orientations. The number of mapped normals on cell c_{ij} gives the value of this cell. All cell's values are mapped to a $R \times C$ matrix, which is called the signature of the 3D object. The similarity between two object signatures a and b is given by $sim(a,b) = \sum_{i=1}^{R} \sum_{j=1}^{C} \left(|a_{ij} - b_{ij}| \,/\, |a_{ij} + b_{ij}| \right)$ [Ip and Wong 2002]. The EGI is scale and translation invariant, but it requires rotational normalization. Retrieval performance studies were performed in Kazhdan et al. [2003] and Funkhouser et al. [2003]. Also, its performance was evaluated in recognition of aligned human head models in Ip and Wong [2002]. There is also a complex version of the EGI (CEGI) [Kang and Ikeuchi 1993], which associates a complex weight to each cell of the EGI.

3.5.4 *Shape descriptor based on the 3D Hough transform.* Zaharia and Prêteux [2002] presents a descriptor based on the 3D Hough transform, the so-called *canon-*

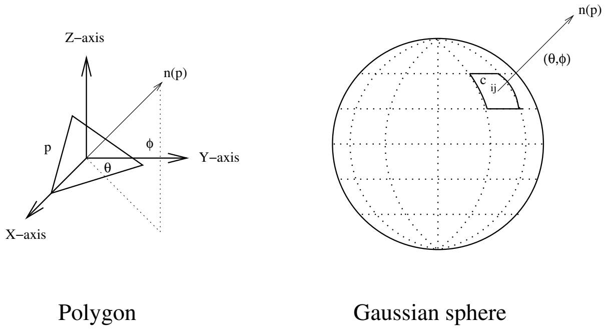


Fig. 19. Mapping from object normals to the Gaussian sphere.

ical 3D Hough transform descriptor (C3DHTD). The basic idea of the 3D Hough transform is to accumulate three-dimensional points within a set of planes. These planes are determined by parametrizing the space using spherical coordinates (e.g., a distances from the origin, b azimuth angles, and c elevation angles, thus obtaining $a \cdot b \cdot c$ planes). Each triangle t of the object contributes to each plane p with a weight equal to the projection area of t on p, but only if the scalar product between the normals of t and p is higher than a given threshold.

Rotation invariance for this descriptor is approximated by normalizing the 3D object with PCA, determining its principal axes, and using its center of gravity as the origin of the coordinate system for the Hough transform. However, it is argued that PCA may fail to provide the correct orientation of a 3D object by just labeling the principal axes according to the eigenvalues (in ascending or descending order). For this reason, the 48 possible Hough transforms (one for each possible PCA-based coordinate system) are aggregated into the descriptor.

The direct concatenation of 48 descriptors would lead to a high complexity in terms of descriptor size and matching computation time. To solve this problem, the authors propose to partition the unit sphere by projecting the vertices of any regular polyhedron. This partitioning schema is then used as parametrization of the space. Then, they show how to derive all Hough transforms from just one of them, which is then called the *generating transform.* The similarity measure between two canonical 3DHT descriptors from objects q and r, h_q and h_r respectively, is defined as $d(h_q, h_r) = \min_{1\leq i\leq 48}\left\{||h_q - h_r^i||\right\}$, where the set $\{h_r^i\}$ corresponds to one of the 48 possible 3D Hough transforms of object r, and $||\cdot||$ denotes the L_1 or L_2 norm. Retrieval experiments were conducted, contrasting the proposed descriptor with the shape spectrum (cf. Section 3.5.2) and EGI (cf. Section 3.5.3) descriptors, attributing best performance to the C3DHT descriptor.

3.6 Image-based descriptors

In the real world, spatial objects, apart from means of physical interaction, are recognized by humans in the way they are visually perceived. Therefore, a natural approach is to consider 2D projections of spatial objects for similarity estimation. Thereby, the problem of 3D retrieval is reduced to one in two dimensions, where techniques from image retrieval can be applied. One advantage of image-based

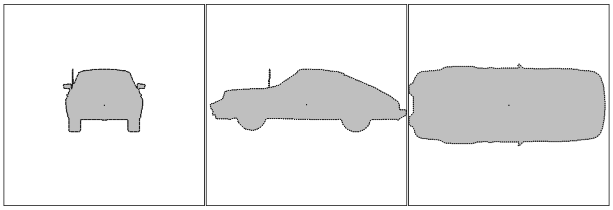


Fig. 20. Silhouettes of a 3D models. Note that, from left to right, the viewing direction is parallel to the first, second, and third principal axes of the model. Equidistant sampling points are marked along the contour.

retrieval methods over most of the other descriptors is that it is straightforward to design query interfaces where a user supplies a 2D sketch which is then input to the search algorithm [Funkhouser et al. 2003; Löffler 2000]. The problem of rotational invariance can again be solved by either rotational normalization preprocessing, by using rotational-invariant features, or by matching over many different alignments simultaneously.

3.6.1 *Description with silhouettes.* A method called *silhouette* descriptor [Heczko et al. 2002] characterizes 3D objects in terms of their silhouettes that are obtained by parallel projections. The objects are first PCA-normalized 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 descriptors of the three resulting contours. To obtain the descriptors, a silhouette contour is scanned by placing equally-spaced, sequential points onto the contour. The sequence of centroid-distances of the (ordered) contour points is Fourier transformed, and magnitudes of a number of low-frequency Fourier coefficients contributes to the feature vector. Via PCA preprocessing, the Silhouette descriptor is pose and scale invariant. Figure 20 illustrates the contour images of a car object.

Experimental results on the retrieval effectiveness of this descriptor were published in Vranic [2004] and some results are also given in Section 4 of this survey. Song and Golshani [2002] also address the usage of projected images for 3D retrieval. The authors propose to render object images from certain directions, and to employ various distance functions on resulting image pairs, e.g., based on circularity measures from the projections, or distances between vectors of magnitudes after Fourier transform. Further work on image-based retrieval methods has been reported in Ansary et al. [2004], Löffler [2000], and Cyr and Kimia [2004].

3.6.2 *Description with depth information.* Another image-based descriptor was proposed in Heczko et al. [2002], and further discussed in Vranic [2004]. The so-called *depth buffer* descriptor starts with the same setup as the silhouette descriptor: the model is oriented 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 orthogonal distance from the viewing plane (i.e., sides of the unit cube) of the object. These images correspond to the concept of z- or depth-buffers in computer graph-

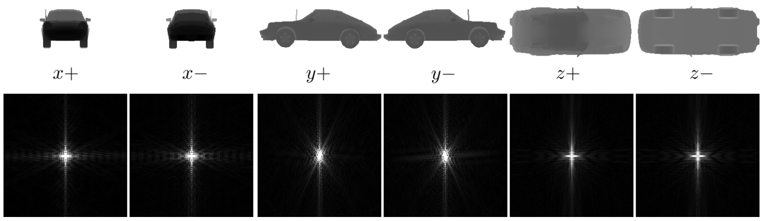


Fig. 21. Depth buffer based feature vector. The first row of images shows the depth buffers of a car model. Darker pixels indicate that the distance between view plane and object is smaller than at brighter pixels. The second row shows coefficient magnitudes of the 2D Fourier transform of the six images.

ics. After rendering, the six images are transformed using the standard 2D discrete Fourier transform, and the magnitudes of certain k first low-frequency coefficients of each image contribute to the depth buffer feature vector of dimensionality $6k$. An illustration of this method is given in Figure 21.

From our own experimental results (see Vranic [2004] as well as Section 4.2 in this paper) we conclude that the depth buffer has good retrieval capability and is able to outperform other descriptors on our benchmarking database.

Figure 21 shows the depth buffer renderings of a car object, as well as a visualization of the respective Fourier transforms.

3.6.3 *Lightfield descriptor.* Chen et al. [2003] proposed a descriptor based on images from many different viewing directions. The authors define the *LightField* descriptor as certain image features extracted from a set of silhouettes that are obtained from parallel projections of a 3D object. A camera system is defined, where a camera is located on each of the vertices of a dodecahedron which is centered at the object's center, completely surrounding the object. The cameras' viewing directions point towards the center of the dodecahedron and the camera up-vector uniquely defined. Considering parallel projections, due to symmetries at most 10 unique silhouettes result from one such camera system. The similarity between two objects is then defined as the minimum of the sum of distances between all corresponding image pairs when rotating one camera system relative to the other, covering all 60 possible alignments of the camera systems. To further support rotation invariance of this method, the authors consider not just one, but ten images per dodecahedron vertex obtained by uniformly varying all camera positions in the neighborhood of the vertex. For a full object to object comparison run, this leads to 5460 rotations of one camera system in order to determine the final distance.

The image metric employed to compare each image pair is the l_1 norm over a vector of coefficients composed of 35 Zernike moments and 10 Fourier coefficients extracted from the rendered silhouettes. For on-line retrieval purposes, this rather expensive algorithm is accelerated by a multi stage filter-and-refinement process. This process gradually increases the number of rotations, images and vector components as well as the component quantization accuracy that are considered in each refinement iteration, discarding all the objects that exhibit a distance greater than

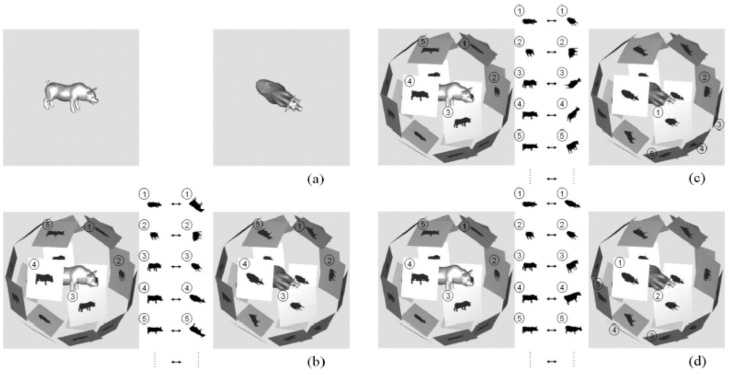


Fig. 22. The LightField descriptor determines similarity between 3D objects by the maximum similarity when aligning sets of projections obtained from an array of cameras surrounding the object. (Figure taken from Chen et al. [2003] (© 2003 Blackwell Publishing). Copyright is held by the owner.)

the mean distance between the query object and all objects of the database. From experiments conducted by the authors as well as those presented in Shilane et al. [2004], it can be concluded that the retrieval quality of this method is excellent, and that it outperforms a range of other methods presented.

3.7 Non-feature vector matching techniques

Up to now, we have reviewed 3D descriptors based on rather fast and easy to extract vectors of real-values measures (feature vectors) defined on model characteristics such as spatial extent, surface curvature, 2D projections and so on. While the feature vector approach is practical for application on large databases of 3D objects, other paradigms for object description and matching exist, originating from computer vision and shape analysis. While very powerful, methods from these fields usually are computationally more complex, may lead to representations other than real vectors, and demand for customized distance functions.

Graphs are a natural choice to capture model topology, but they often involve complex extraction algorithms and matching strategies. Graphs have been derived from model surface [Hilaga et al. 2001; McWherter et al. 2001; McWherter et al. 2001] and volumetric [Sundar et al. 2003] properties of 3D models. Similarity calculation can proceed on the graphs themselves via customized graph matching approaches [Hilaga et al. 2001; Bespalov et al. 2003] or via numeric descriptions of the graphs obtained, e.g., using spectral theory [McWherter et al. 2001], or combinations of both methods. In Sections 3.7.1 and 3.7.2, we exemplarily recall two graph-based techniques, noting that a growing body of work exists in this area [Bespalov et al. 2003; Biasotti et al. 2003].

Besides graphs and feature vectors, customized numeric data structures have been proposed for 3D description and retrieval, such as the *Spin Images* recalled in Section 3.7.3.

While all of these methods introduce interesting matching concepts, their application to large databases of general 3D objects raises problems due to complexity issues, or certain restrictions imposed on the types of 3D models supported.

3.7.1 *Topological matching.* Hilaga et al. [2001] present an approach to describe the topology of 3D objects by a graph structure and show how to use it for matching and retrieval. The algorithm is based on constructing so-called Reeb graphs from the models which can be interpreted as information about the skeletal structure of an object. The basic idea is to partition the object into connected portions by analyzing a function μ that is defined over the entire object's surface. Informally, the Reeb graph generated from a 3D object is made up of nodes that represent portions of the object for which μ assumes values ranging in certain value intervals. Parent-child relationships between nodes represent adjacent intervals of these function values for the contained object parts. For computing the similarity of two objects, it is proposed to compare the topology of the objects respective Reeb graphs, as well as similarities between the mesh properties of the model parts that are associated with corresponding graph nodes.

Defining a suited function μ is critical to the construction of graphs suited for object analysis and matching. E.g., the height function $h(x, y, z) = z$ that returns the height of a surface at position (x, y) is suited to analyze terrain data, where orientation is well-defined. To be rotation, translation and scale invariant, Hilaga et al. [2001] propose to use the appropriately normalized sum of geodesic distances between a unique central point and all other points of the model surface as the function μ. Intuitively, if for a point p of the objects surface, $\mu(p)$ is relatively low, p is expected to be closer to the "center" of the object, while points on the object's periphery would possess higher function values. To construct the final descriptor for an object, the range of possible function values is discretized into a number of bins. For each bin, the restriction of the object to the parts containing μ-values in the respective bin, topologically connected subparts are identified and aggregated into a node of the Reeb graph each. By merging the nodes belonging to adjacent bins, a given Reeb graph is recursively condensed into coarser Reeb graphs, and the so-called multi-resolution Reeb graph (MRG) is obtained. The authors give details on computing the descriptor and a coarse-to-fine MRG matching strategy. Sensitivity and retrieval experiments are reported, indicating that the descriptor is useful for retrieving topologically similar objects according to human notion. Figure 23 schematically illustrates construction of a Reeb graph, and visualizes the geodesic distance function on two similar, but deformed objects.

3.7.2 *Skeleton-based object matching.* Skeletons derived from solid objects can be regarded as intuitive object descriptions. They are able to capture important information about the structure of objects with applications in, e.g., object analysis, compression, or animation. In order to use skeletons for 3D object retrieval, suitable skeletonization algorithms and skeleton similarity functions have to be defined. In a recent paper, Sundar et al. [2003] presented a framework for this task. To obtain a thin skeleton, the authors proposed to first apply a thinning algorithm on the voxelization of a solid object. The method reduces the model voxels to those voxels that are important for object reconstruction, as determined by a heuristic that relates the distance transform value of each voxel with the mean of the distance transform values of the voxels among its 26-neighbors [Gagvani and Silver 1999].

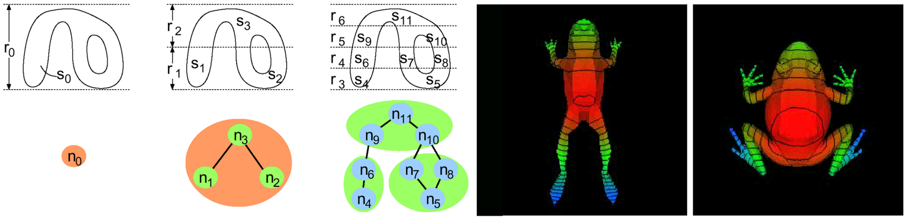


Fig. 23. Left is shown the construction of several Reeb graphs by recursively refining μ-intervals. The right image visualizes the aggregated, normalized geodesic distance function evaluated on two topologically similar objects. (Figure taken from [Hilaga et al. 2001] (© 2001 ACM Press). Copyright is held by the owner.)

In a second step, the remaining voxels are clustered, and a minimum spanning tree is constructed connecting the voxel clusters. Clustering and connecting proceed subject to the condition of not violating the object boundary. The resulting tree may be converted to a directed acyclic graph (DAG) by directing edges guided by the distance transform values of the voxels within a cluster. It may also be converted to a uniquely rooted tree [Siddiqi et al. 1998]. Having obtained a DAG, to each node in the DAG a *topological signature vector* (TSV) is associated as a node label, which is formed by sums of eigenvalues of the adjacency matrices of all subtrees rooted at the considered node. The TSV is used to encode structural information about the subgraph rooted at the respective node. As another node label, measures for the distribution of distance transform values of the respective cluster members are considered. These node labels constitute the input to a distance function that measures similarity between individual nodes in skeletal graphs. The final matching of two skeletal graphs is performed by establishing a set of node-to-node correspondences between the graphs based on a greedy, recursive bipartite graph matching algorithm [Shokoufandeh and Dickinson 2001]. A final measure for the dissimilarity between two skeletal graphs may be obtained from the quality of the node-correspondences as determined by their node label distance.

The authors demonstrated the capabilities of their framework by a number of matches obtained from querying a test database (see Figure 24 for an example). They emphasize the method's suitability for matching articulated objects and also the potential for finding partial matches between objects. The approach requires several parameters to be set, e.g., the threshold levels for thinning and clustering.

3.7.3 *Spin-images.* A 3D descriptor using sets of so-called *spin-images* to characterize 3D objects was proposed by Johnson and Hebert [1999] and, regarding the matching process, modified in a study by de Alarcón et al. [2002]. The descriptor is rotation and translation invariant by design. It requires a set of points on the model surface and associated normal vectors (that is, an oriented point set O) as input. The basic idea is to generate a set of two-dimensional histograms of the object geometry in the neighborhood of selected points, and to use these descriptions to search for point-to-point correspondences between two models. It can also be used to search for correspondences between a model and a whole 3D scene. Via refinement steps a final measure for similarity between parts of an object, or two

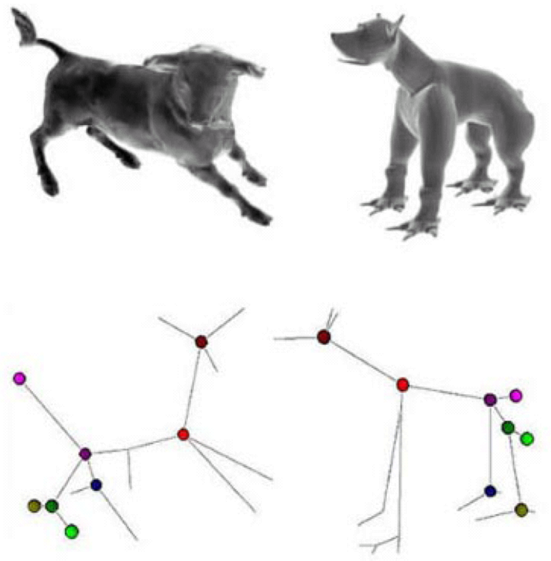


Fig. 24. A pair of mutually best-matching objects from a database of about 100 models. (Figure taken from Sundar et al. [2003] (© 2003 IEEE). Copyright is held by the owner.)

objects as a hole can be generated. We recall the description generation algorithm in the following.

For each $o_i \in O$, O for example chosen as the centers of mass of (oriented) triangles of a model mesh, a so-called spin-image S_i is generated by accumulating the output of a mapping $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ in a two–dimensional index, the spin-image. These spin-images describe object geometry in the neighborhood of o_i. The set of spin images gives an object description by a set of descriptions each local to one point o_i from the object. Given an object point o_i and its associated normal vector n_i, the mapping is performed by building a 2dimensional histogram from distances $\alpha_{i,j}$ and $\beta_{i,j}$. $\alpha_{i,j}$ is defined as the distance of all points $o_j \in O$, $i \neq j$ to the line L extending n_i to infinity. $\beta_{i,j}$ is defined as the distance of all points $o_j \in O$, $i \neq j$ to the plane through o_i with normal vector n_i. The pair of distance distributions $(\alpha_{ij}, \beta_{ij})$ for a point o_i is then discretized into a two-dimensional histogram S_i, where for each distance bin the number of points o_j that belong to the respective bin is recorded. Note that this mapping is equal to discretizing radius and elevation components of points o_j in a cylindrical coordinate system given by origin o_i and normal vector n_i. Via thresholding the relevant neighborhood around o_i is controlled; alternatively, one may consider the complete object as the neighborhood. The authors suggest to apply bilinear filtering on the spin-images, in order to reduce the impact of noise. Scaling invariance is provided by normalizing the distance range to unit length. Figures 25 and 26 illustrate the spin images generation process.

Due to potentially high storage and computation overhead when cross-comparing all spin-images of two objects, and also the presence of redundant information

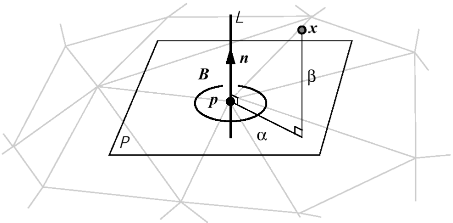


Fig. 25. Building a spin image as a histogram of distances α and β of points in some neighborhood with respect to basis point p. (Figure taken from Johnson and Hebert [1998] (© 1998 Elsevier) with permission from Elsevier. Copyright is held by the owner.)

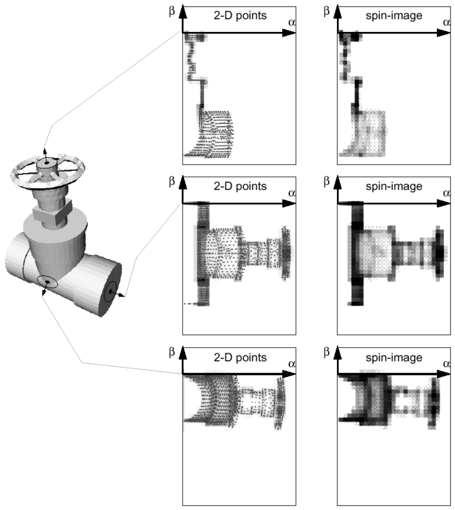


Fig. 26. Selected spin images generated from a 3D model. (Figure taken from Johnson [1997]. Copyright is held by the author.)

among close or symmetrically related spin images, Johnson and Hebert [1999] suggest to perform compression on the set of an object's spin images using dimensionality reduction. de Alarcón et al. [2002] propose a two-step data reduction process by first clustering a spin-image set using the self-organizing map (SOM) algorithm to group similar spin images, followed by application of a clustering algorithm. This technique is suited to reduce the number of required descriptor comparisons by checking only the spin image prototypes. Also, Assfalg et al. [2004] suggested spin image post-processing techniques that help to reduce the number of spin images used to describe each object. Particularly, spin images were interpreted as grey-scale images, which could be efficiently described by a low-dimensional region-based description scheme from the content-based image retrieval (CBIR) domain. Orthogonally, fuzzy clustering was proposed to reduce the number of spin images to a smaller number of prototypes, onto which a sum of cluster distance function was suggested.

4. COMPARISON BETWEEN 3D DESCRIPTORS

Comparing the surveyed descriptors is a difficult task, since the amount of technical details given in the original literature did vary between different descriptors, and most of the authors did employ individually compiled benchmarks when empirically evaluating retrieval precision. We recognize it is a tremendous task to (re-)produce an objective analytic and experimental comparison of the wealth of 3D retrieval methods, which is beyond our resources. What we provide here is a limited comparison of key features of the surveyed algorithms, as well as an experimental comparison of the retrieval performance measured against our own benchmark, of a subset of algorithms of which we possess implementations.

4.1 Qualitative comparison

We first summarize the surveyed methods using the following main algorithm characteristics:

—*Proposed dimensionality* gives either the dimensionality that was found to perform best if experiments were performed, or recommended values from the literature if available. It is generally agreed that the optimal dimensionality depends on the experimental setup, that is, the choice of database and ground truth for retrieval experiments.

—*Invariance* indicates the provided invariance (Rotation (R), Translation (T), Scale (S)), and how they are achieved (implicit or by object normalization).

—*Required object representation* gives the assumed object representation. Typically the algorithms work on triangulations, but some assume point clouds or voxelizations.

—*Consistency requirements* states the properties required in addition to geometry information, mainly topology and orientation of objects.

—The *proposed metric* mentions the distance function used in retrieval experiments or recommended by the respective authors (if applicable). If an interval is given, good retrieval results are reported throughout the interval.

Table II shows a qualitative comparison between the FVs surveyed in Section 3. Where more than one descriptor was proposed in a descriptor class (e.g., for statistical moments), we chose the descriptor that was technically described best in the literature, according to our view. We omitted descriptors from the table, in case the technical description in the original sources were insufficient for this comparison. In general, much technical detail is also encapsulated in the implementation of object preprocessing such as mesh normalization, surface point resampling, choosing voxel grid resolutions, etc. Also, parametrization of the numeric transform methods often employed, such as Fourier and Spherical Harmonics transform, which in itself may have an impact on retrieval performance of the methods, would have to be considered. Such effects are not reflected in the comparison table.

Table II. Comparison table between 3D descriptors.

Descriptor	Proposed Dim.	Invariance	Req. Obj. Represent.	Req. Obj. Consistency	Proposed Metric
Parametrized Statistics 3.2.2	$64 \times 3 \times 3$	RTS via PCA	Triangulation	-	L_2, elastic matching
Shape distribution 3.2.4	1024 bins	RT implicit, S via hist. norm.	Triangulation	-	L_1
Shape histograms 3.4.2	122–240 bins	RTS via PCA	Point cloud	-	Quadratic form
Rot. inv. point cloud 3.4.3	21	RTS via PCA	Triangulation	-	Not specified
Voxel 3.4.4	172	RTS via PCA	Triangulation	-	L_1
Volume 3.4.5	486	RTS via PCA	Triangulation	Orientation	L_1
Cords 3.2.5	120	RT via PCA, S via hist. norm.	Triangulation	-	L_1
Ray-based sampling 3.3.1	91–169	RTS via PCA	Triangulation	-	L_1, L_2
Rot. inv. sph. harm. 3.4.6	512	TS via norm., R implicit	Triangulation	-	L_2
Reflective symmetry 3.4.7	-	TS via norm., R assumed	Triangulation	-	L_∞
Surf. normal properties 3.5.1	n.a.	RT via PCA, S implicit	Triangulation	Orientation	n.a.
Shape spectrum 3.5.2	10–100	RTS implicit	Triangulation	Orientation	L_1, L_2
Ext. Gaussian image 3.5.3	200	TS implicit, R assumed	Triangulation	Orientation	Histogram metric
Canonical 3DHT 3.5.4	2560	RTS via PCA	Triangulation	Orientation	L_1, L_2
Weighted point sets 3.4.8	25×3 cells in signature	RTS via PCA	Triangulation	Orientation (some of the variants)	Solution to transport problem
Silhouette 3.6.1	375	RTS via PCA	Triangulation	-	L_1
Depth buffer 3.6.2	366	RTS via PCA	Triangulation	-	L_1
Lightfield 3.6.3	45 per image	RTS implicit	Triangulation	-	Multistage matching
Topological matching 3.7.1	n.a.	RTS implicit, non-structural deformation	Triangulation	Non-disconnected objects	Custom graph matching
Skeletonization 3.7.2	n.a.	RTS implicit	Volumetric	Volume	Custom graph matching
Sping Image 3.7.3	n.a.	RTS implicit	Point Cloud	-	Correlation

Table III. Description of the classified set of our 3D object database.

Class id	Description	# of models	Class id	Description	# of models
1	ants	6	29	submarines	5
2	rabbits	4	30	warships	5
3	cows	7	31	beds	7
4	dogs	4	32	chairs	24
5	fish-like	13	33	office chairs	6
6	bees	5	34	sofas	4
7	CPUs	4	35	benches	3
8	keyboards	8	36	couches	11
9	cans	4	37	axes	4
10	bottles	14	38	glasses	7
11	bowls	4	39	knives	3
12	pots	4	40	screws	3
13	cups	8	41	spoons	3
14	wine glasses	9	42	tables	6
15	teapots	4	43	skulls	3
16	biplanes	5	44	human heads	8
17	helicopters	9	45	human masks	4
18	missiles	16	46	books	4
19	jet planes	18	47	watches	2
20	fighter jet planes	26	48	sand clocks	4
21	propeller planes	10	49	swords	25
22	other planes	4	50	barrels	3
23	zeppelins	6	51	birches	4
24	motorcycles	5	52	flower pots	9
25	sport cars	6	53	trees	11
26	cars	23	54	weeds	9
27	Formula-1 cars	9	55	human bodies	56
28	galleons	4			

4.2 Experimental comparison

The database used for our experiments contains 1,838 3D objects that we collected from the Internet [2]. From this set, 472 objects were manually classified by shape similarity into 55 different model classes. The rest of the objects were left as "unclassified". Each classified object of each model class was used as a query object. The objects belonging to the same model class, excluding the query, were taken as the relevant objects.

Table III gives a complete description of the classified objects of the database. The first column indicates the class identification number. The second column describes the 3D class models. The last column lists the number of objects per model class.

We implemented 16 different types of FVs to perform experiments, which includes: statistical FVs (3D moments), geometry based FVs (principal curvature, shape distribution, ray-based, ray-based with spherical harmonics, shading, complex valued shading, cords-based, segment volume occupation, voxel based, 3DDFT, rotation invariant spherical harmonics), image based FVs (depth buffer, silhouette), and other approaches (rotation invariant point cloud descriptor).

[2] Konstanz 3D Model Search Engine. `http://merkur01.inf.uni-konstanz.de/CCCC/`

4.3 Computational complexity of descriptors

Firstly, we compared the computational complexity of 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 for query examples submitted by a user to 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. We made the experience 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 very compute intensive. In general, PCA object preprocessing only constituted a minor fraction of total extraction cost, as on average the PCA cost was only 3.59 seconds for the complete database of 1,838 objects (1.96 milliseconds per object on average).

Figure 27 shows the average extraction time as a function of the dimensionality of a descriptor. We did not include in this chart those descriptors that posses the multiresolution 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 milliseconds). 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).

4.4 Effectiveness comparison between descriptors

We use *precision versus recall figures* [Baeza-Yates and Ribeiro-Neto 1999] for comparing the effectiveness of the search algorithms. *Precision* is the fraction of the retrieved objects which are relevant to a given query, and *recall* is the fraction of the relevant objects which have been retrieved from the database. That is, if R is the set of relevant objects to the query, A is the set of objects retrieved, and R_A is the set of relevant objects in the result set, then

$$Precision = \frac{|R_A|}{|A|} \text{ and } Recall = \frac{|R_A|}{|R|}.$$

All our precision versus recall figures are based on the eleven standard recall levels (0%, 10%, ..., 100%), 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 calculate the *R-precision* [Baeza-Yates and Ribeiro-Neto 1999] for each query, which is defined by the precision when retrieving only the first N objects. The R-precision gives a single number to rate the performance of a retrieval algorithm. This measure is similar to the *Bull-Eye Percentage (BEP)* score adopted as an evaluation standard by MPEG-7. The BEP is also a single value measure and equal to recall when

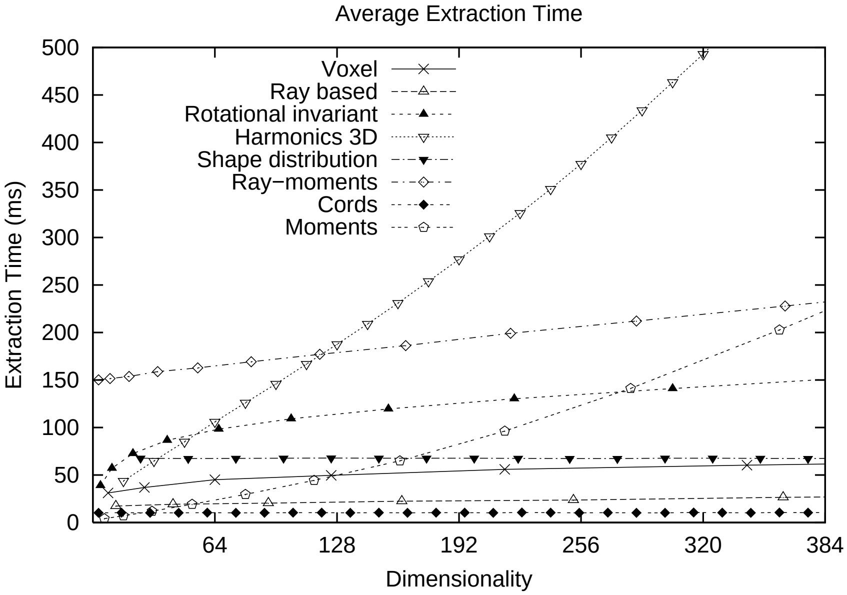


Fig. 27. Average extraction time for some of the descriptors while varying its dimensionality.

retrieving the first $2N$ objects.

We tested all these FVs using different levels of resolution, from a few dimensions up to 512, and we used the Manhattan (L_1) distance as the similarity function between vectors (we also tested L_2 and L_{max}, but we consistently obtained the best effectiveness scores using L_1). Table IV shows the best R-precision values obtained with all the FVs in descending order. The first column lists the different descriptors. The second column indicates the best dimensionality (in terms of effectiveness) of the FV. The last column lists the average R-precision values obtained for each FV with their best dimensionality.

The best overall FV among our set of implemented methods was the depth buffer, with an average R-precision of 0.32. The difference in effectiveness between the best and the worst performing FV (depth buffer and principal curvature, respectively) was significant. However, the difference in effectiveness between "similar performing" FVs was small, specially when comparing the most effective descriptors. This implies that in practice these best FVs should be suited about equally well for retrieval of general polygonal objects. As a contrast, the effectiveness difference between the worst and the best descriptor was significant (up to a factor of 3). We observed that descriptors that rely on consistent polygon orientation like shape spectrum or volume exhibited low retrieval rates, as consistent orientation is not guaranteed for many of the models retrieved from the Internet. Also, the geometrical moment-based descriptors seem to offer only limited discrimination capabilities. Figures 28 and 29 show the precision vs. recall figures for all the implemented descriptors (first eight and last eight descriptors according to Table IV, respectively).

Figures 30 and 31 (first eight and last eight descriptors, respectively) show the effect of the descriptor dimensionality on the overall effectiveness. The figure shows

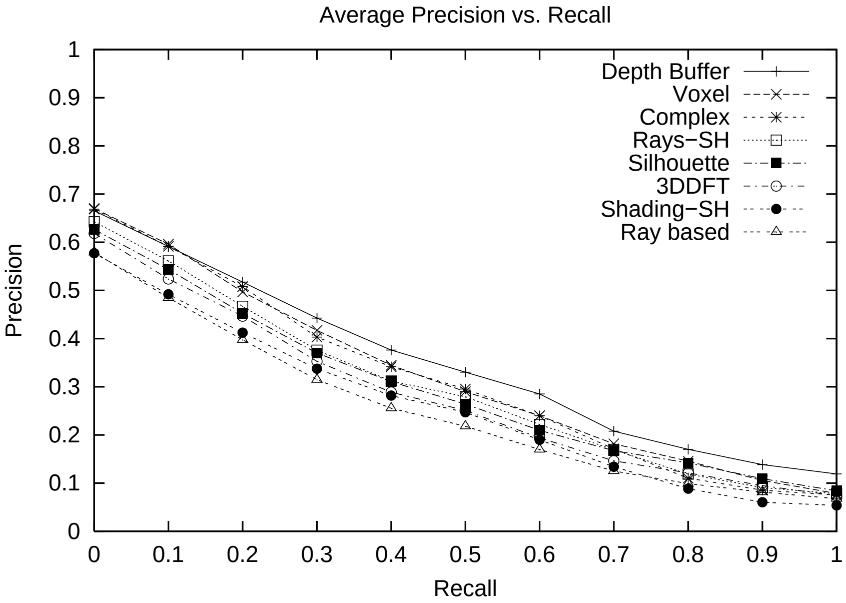


Fig. 28. Average precision vs. recall with best dimensionality, first eight descriptors according to Table IV.

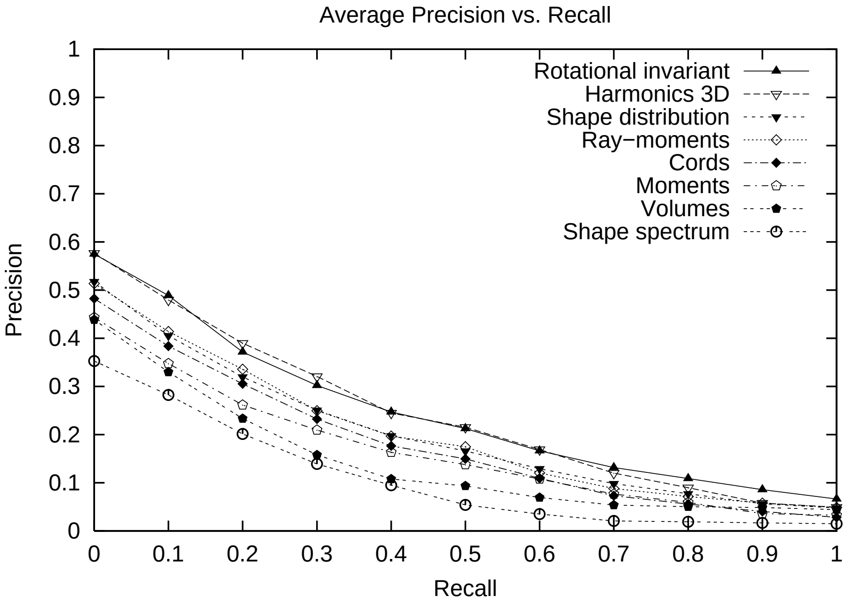


Fig. 29. Average precision vs. recall with best dimensionality, last eight descriptors according to Table IV.

Table IV. Average R-precision of the 3D descriptors.

Descriptor	Best dimensionality	Avg. R-precision
Depth buffer	366	0.3220
Voxel	343	0.3026
Complex valued shading	196	0.2974
Rays with spherical harmonics	105	0.2815
Silhouette	375	0.2736
3DDFT	365	0.2622
Shading	136	0.2386
Ray-based	42	0.2331
Rotation invariant point cloud	406	0.2265
Rotation invariant spherical harmonics	112	0.2219
Shape distribution	188	0.1930
Ray moments	363	0.1922
Cords-based	120	0.1728
3D moments	31	0.1648
Volume	486	0.1443
Principal curvature	432	0.1119

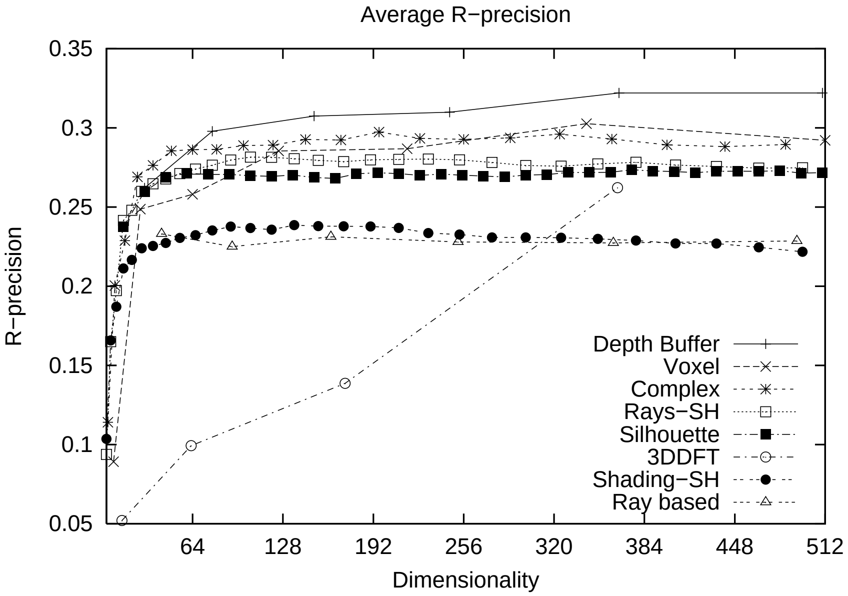


Fig. 30. Dimensionality vs. R-precision, first eight descriptors according to Table IV.

that the effectiveness of the FVs first increases with dimensionality, but the improvement rate diminishes quickly for roughly more than 64 dimensions for most FVs (except for 3DDFT). It is interesting to note that the saturation effect is reached for most descriptors at roughly the same dimensionality level. This is an unexpected result, considering that different FVs describe different characteristics of 3D objects.

We also performed some tests using the Princeton Shape Benchmark [Shilane et al. 2004], to contrast our experimental results with those obtained using a different 3D ground truth. In summary, we obtained the same results as with our

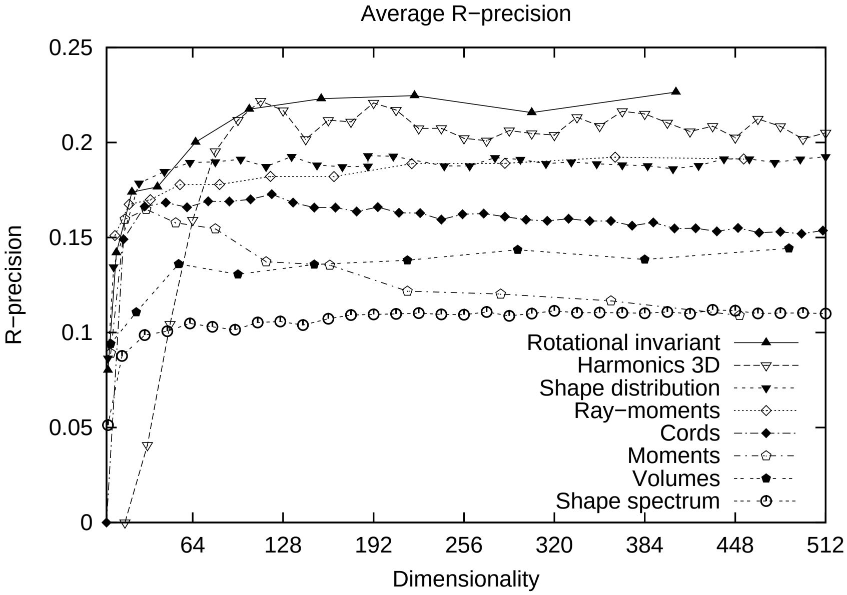


Fig. 31. Dimensionality vs. R-precision, last eight descriptors according to Table IV.

database with only minor differences (see Bustos et al. [2005] for details).

4.5 Analysis of the experimental results

From the experimental results obtained in our experiments and on the limited set of implemented descriptors, we conclude that the best descriptors on average are those based on projections (2D, ray based) of the original 3D object, e.g., depth buffer, silhouette, etc. Exceptions to this rule are the voxel FV and the 3DDFT FV, which are volumetric descriptors and also obtained a good experimental effectiveness. Surface based descriptors obtained in general low effectiveness values. All the implemented FVs showed good robustness with the respect to the level of detail of the 3D objects. The good retrieval quality of image-based descriptors from our experiments are in accordance with Chen et al. [2003], where an image-based descriptor embedded in an advanced multi-stage matching framework was shown to provide excellent retrieval results, and outperformed several other descriptors.

However, we also observed significant variance with respect to effectiveness of the retrieval when comparing the results for classes of objects. For different classes of objects, a different FV was usually the most effective one. Unfortunately, we could not find a strong correlation between geometric properties of the 3D object class and the best suited FV for that model class. A notable exception for this is the Shape Spectrum descriptor (the worst descriptor on average), which obtained the best retrieval effectiveness for the human model class. Shape spectrum was able to recognize different models of human bodies in different poses, something that was not possible for the rest of the implemented FVs. This can be attributed to the fact that the Shape Spectrum descriptor considers the distribution of local curvature on the 3D object, which does not vary considerably in similar 3D models with different poses (e.g., human bodies in different positions). Another observation that we made is, that model classes which are difficult to correctly orient using PCA (cf. Figure 4, first row) are best retrieved by FVs that are inherently rotational invariant, e.g., rotational invariant spherical harmonics.

Besides these specific exceptions, it is difficult to assess a priori which FV will have the best retrieval effectiveness for an unknown query object. On average, depth buffer or voxel will do pretty well, but one would like to always select the best FV given a query object. Therefore, it is hard to give a recommendation with respect to which FV to implement into a 3D similarity search system, when one wants to build one.

A promising approach to solve this problem is to resort to the usage of *combinations of feature vectors*. The idea is to not only use one but several FVs together, hence taking advantage of the particularities of each considered FV. A linear combination of FVs will not provide the optimal results, because if one of the considered FVs has a very bad effectiveness for the given query object, then it will "spoil" the final result. Dynamic weighting methods have been recently proposed [Bustos et al. 2004b; 2004a], which aim to avoid this problem, giving only a high weight to those FVs that are more promising to the query object. There, the goodness of a FV is estimated against a training (or reference) database prior to performing the weighted query against the actual database. The presented experimental results showed noticeable improvements in the overall effectiveness of the retrieval system by using dynamically generated combinations of FVs.

Regarding the nature of the surveyed FVs, we conclude that they are all proposed for usage on databases which do not restrict the type of objects contained. In practice, authors have used "general purpose" VRML models obtained from the Internet, representing a wide spectrum of objects. Of course, if the type of models to be supported can be anticipated in advance, it is possible to either perform benchmarks targeted at the specific models to select the best FVs to implement. On the other hand, if the relevant model features for the retrieval task are known, it should be possible to design custom descriptors. E.g., in CAD databases it might be possible to specify certain geometric features relevant for a construction process, so one could design descriptors exploiting such knowledge. Identifying application-specific requirements and designing descriptors that support them is an interesting future work with high commercial potential, as we expect.

5. CONCLUSIONS

This survey described a variety of recently proposed feature-based descriptors for 3D objects, and introduced a taxonomy to classify them. As we believe, the reported feature extraction methods present the first important achievements in the search for general-purpose, fast retrieval algorithms for 3D object databases. The feature vector approach maps 3D objects to a vector of real values, which in term can be used for distance calculation. Furthermore, it makes applicable the wide area of multimedia indexing techniques, which have been researched for a long time now [Böhm et al. 2001]. Retrieval systems may also profit from semi-interactive query enhancement methods, like relevance feedback [Elad et al. 2002], annotation information [Zhang and Chen 2001], or feature selection and combination techniques

[Bustos et al. 2004b].

While many sophisticated object analysis and matching methods exist in the domain of computational geometry and computer vision, those are usually tailored to specific recognition problems, and it is questionable whether these easily extent to the database retrieval problem. This is due to restrictions imposed on the objects and due to computational complexity issues. Extracting feature vectors from skeletal representations of the objects [Lou et al. 2004; Sundar et al. 2003] is an interesting approach, but to date its applicability to the database retrieval problem in terms of effectiveness and efficiency is unclear. Specifically, the robustness of such methods with respect to feature extraction parametrization has to be explored.

Considering the wealth of feature extraction methods proposed so far (still, new methods defining novel 3D features are proposed regularly), selecting the ones to use when building an actual 3D retrieval system is a difficult problem. A complete and fair comparison of all the main available methods seems not feasible, as it is currently more attractive for researchers to propose new methods, than to re-implement existing ones. But, considering computational complexity and object consistency requirements can provide guidance in order to select application-specific methods to implement.

In this survey, we compared the computational complexity of certain feature vectors, which are currently implemented in our own system. In practice, the normalization step and the descriptor computation cost is small, and almost all descriptors can be computed in less than a second for an object on average, and on a standard workstation. As the descriptor computation must be performed only once per object, this implies that the described descriptors can be used for real-world applications.

We also experimentally compared a wide variety of 3D FVs on a classified database of 3D objects, formed by models collected from the Internet, and we compared their retrieval performance using standard effectiveness measures from the information retrieval domain (precision vs. recall diagrams and the R-precision values). Our experimental comparison of 16 different 3D FVs shows that there is a number of them that have good average effectiveness and work well in most cases (e.g., depth buffer, voxel and complex FVs) for the types of 3D models one can find on the Internet today.

There remain important open problems in the research of content-based description and retrieval of 3D objects, some of which we sketch in the following.

Considering searching 3D objects from heterogeneous databases, where the objects may be arbitrarily scaled and oriented in their respective coordinate systems, scale and rotation invariant methods must either normalize the models, or employ descriptions that provide these invariances implicitly. Most methods to date advocate rotation and translation normalization based on Principal Components Analysis. As PCA may lead to counter-intuitive alignment results for certain 3D models [Funkhouser et al. 2003; Tangelder and Veltkamp 2003], extensions and alternatives to the PCA-based normalization need to be devised. Depending on the application domain, additional invariance may be desirable, e.g., invariance with respect to local deformations in geometry and topology, or invariance with respect to anisotropic scaling [Kazhdan et al. 2004].

Also, the current methods focus mainly on geometric aspects of 3D models. Left aside are other attributes which are present in many 3D databases: Color, material properties, and texture can be specified in many formats, like in the popular VRML format. More specialized formats from the CAD domain usually contain also structural object information and machining process information; This information might as well be exploited for 3D retrieval.

How to improve the efficiency of 3D search systems is also an open issue. The need for appropriate indexing techniques, considering the high dimensionality of the descriptors seems obvious. Moreover, if we consider the segmentation of objects as a possible approach for partial similarity search, then the original database with a few thousands of models can be transformed into a database with millions of objects, where efficiency considerations become mandatory.

ACKNOWLEDGMENTS

We thank the editors of ACM Computing Surveys and the anonymous referees for their helpful comments on the earlier version of this article. We also thank the authors and publishing houses that gave us permission to use some of the figures included in this article.

This work was partially funded by the German Research Foundation (DFG), Projects KE 740/6-1, KE 740/8-1, and SA 449/10-1 within the strategic research initiative SPP 1041 “Distributed Processing and Delivery of Digital Documents” (V3D2). It was also partially funded by the Information Society Technologies program of the European Commission, Future and Emerging Technologies under the IST-2001-33058 PANDA project (2001–2004). The first author is on leave from the Department of Computer Science, University of Chile.

REFERENCES

ANKERST, M., KASTENMÜLLER, G., KRIEGEL, H.-P., AND SEIDL, T. 1999a. 3D shape histograms for similarity search and classification in spatial databases. In *SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases.* Springer-Verlag, London, UK, 207–226.

ANKERST, M., KASTENMÜLLER, G., KRIEGEL, H.-P., AND SEIDL, T. 1999b. Nearest neighbor classification in 3D protein databases. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology.* AAAI Press, 34–43.

ANSARY, T. F., VANDEBORRE, J.-P., MAHMOUDI, S., AND DAOUDI, M. 2004. A bayesian framework for 3D models retrieval based on characteristic views. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission.* IEEE Computer Society, 139–146. Poster.

ASHLEY, J., FLICKNER, M., HAFNER, J., LEE, D., NIBLACK, W., AND PETKOVIC, D. 1995. The query by image content (QBIC) system. *SIGMOD Rec. 24,* 2, 475.

ASSFALG, J., BIMBO, A. D., AND PALA, P. 2004. Retrieval of 3D objects by visual similarity. In *MIR '04: Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval.* ACM Press, New York, NY, USA, 77–83.

BAEZA-YATES, R. A. AND RIBEIRO-NETO, B. 1999. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

BERCHTOLD, S., KEIM, D. A., AND KRIEGEL, H.-P. 1997. Using extended feature objects for partial similarity retrieval. *The VLDB Journal 6,* 4, 333–348.

BESPALOV, D., SHOKOUFANDEH, A., REGLI, W. C., AND SUN, W. 2003. Scale-space representation of 3D models and topological matching. In *SM '03: Proceedings of the 8th ACM Symposium on Solid Modeling and Applications.* ACM Press, New York, NY, USA, 208–215.

Biasotti, S., Marini, S., Mortara, M., Patanè, G., Spagnuolo, M., and Falcidieno, B. 2003. 3D shape matching through topological structures. In *Proceedings of the 11th International Conference on Discrete Geometry for Computer Imagery*. LNCS 2886. Springer, 194–203.

Böhm, C., Berchtold, S., and Keim, D. A. 2001. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv. 33,* 3, 322–373.

Bustos, B., Keim, D. A., Saupe, D., Schreck, T., and Vranić, D. 2004a. Automatic selection and combination of descriptors for effective 3D similarity search. In *Proceedings of the IEEE International Workshop on Multimedia Content-based Analysis and Retrieval*. IEEE Computer Society, 514–521.

Bustos, B., Keim, D. A., Saupe, D., Schreck, T., and Vranić, D. 2004b. Using entropy impurity for improved 3D object similarity search. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 1303–1306.

Bustos, B., Keim, D. A., Saupe, D., Schreck, T., and Vranić, D. 2005. An experimental effectiveness comparison of methods for 3D similarity search. In *Journal of Digital Libraries*. Springer-Verlag.

Campbell, R. J. and Flynn, P. J. 2001. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding 81,* 2, 166–210.

Chávez, E., Navarro, G., Baeza-Yates, R., and Marroquín, J. L. 2001. Searching in metric spaces. *ACM Comput. Surv. 33,* 3, 273–321.

Chen, D.-Y., Tian, X.-P., Shen, Y.-T., and Ouhyoung, M. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum 22,* 3, 223–232.

Cyr, C. M. and Kimia, B. B. 2004. A similarity-based aspect-graph approach to 3D object recognition. *Int. J. Comput. Vision 57,* 1, 5–22.

de Alarcón, P. A., Pascual-Montano, A. D., and Carazo, J. M. 2002. Spin images and neural networks for efficient content-based retrieval in 3D object databases. In *CIVR '02: Proceedings of the International Conference on Image and Video Retrieval*. Springer-Verlag, London, UK, 225–234.

Elad, M., Tal, A., and Ar, S. 2002. Content based retrieval of VRML objects: an iterative and interactive approach. In *Proceedings of the 6th Eurographics Workshop on Multimedia 2001*. Springer-Verlag New York, Inc., New York, NY, USA, 107–118.

Faloutsos, C. 1996. *Searching Multimedia Databases by Content*. Kluwer Academic Publishers, Norwell, MA, USA.

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

Gagvani, N. and Silver, D. 1999. Parameter-controlled volume thinning. *CVGIP: Graph. Models Image Process. 61,* 3, 149–164.

Geradts, Z., Hardy, H., Poortman, A., and Bijhold, J. 2001. Evaluation of contents-based image retrieval methods for a database of logos on drug tablets. In *Proceedings of SPIE Enabling Technologies for Law Enforcement and Security*. Vol. 4232. 553–562.

Healy, D. M., Rockmore, D. N., Kostelec, P. J., and Moore, S. S. B. 2003. FFTs for the 2-sphere - Improvements and variations. *Journal of Fourier Analysis and Applications 9,* 4, 341–385.

Heczko, M., Keim, D. A., Saupe, D., and Vranić, D. 2002. Methods for similarity search on 3D databases. *Datenbank-Spektrum 2,* 2, 54–63. In German.

Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, New York, NY, USA, 203–212.

Horn, B. 1984. Extended Gaussian image. *Proceedings of the IEEE 72,* 12, 1671–1686.

Ip, C. Y., Lapadat, D., Sieger, L., and Regli, W. C. 2002. Using shape distributions to compare solid models. In *SMA '02: Proceedings of the 7th ACM Symposium on Solid Modeling and Applications*. ACM Press, New York, NY, USA, 273–280.

Ip, C. Y., Regli, W. C., Sieger, L., and Shokoufandeh, A. 2003. Automated learning of model classifications. In *SM '03: Proceedings of the 8th ACM Symposium on Solid Modeling and Applications*. ACM Press, New York, NY, USA, 322–327.

Ip, H. and Wong, W. 2002. 3D head models retrieval based on hierarchical facial region similarity. In *Proceedings of the 15th International Conference on Vision Interface*. 314–319.

Johnson, A. E. 1997. Spin-images: A representation for 3-D surface matching. Ph.D. thesis, Robotics Institute, Carnegie Mellon University.

Johnson, A. E. and Hebert, M. 1998. Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing 16,* 9–10, 635–651.

Johnson, A. E. and Hebert, M. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell. 21,* 5, 433–449.

Kang, S. B. and Ikeuchi, K. 1993. The complex egi: A new representation for 3-d pose determination. *IEEE Trans. Pattern Anal. Mach. Intell. 15,* 7, 707–721.

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

Kazhdan, M., Chazelle, B., Dobkin, D., Funkhouser, T., and Rusinkiewicz, S. 2003. A reflective symmetry descriptor for 3D models. *Algorithmica 38,* 1, 201–225.

Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. 2004. Shape matching and anisotropy. *ACM Trans. Graph. 23,* 3, 623–629.

Keim, D. A. 1999. Efficient geometry-based similarity search of 3D spatial databases. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*. ACM Press, New York, NY, USA, 419–430.

Kriegel, H.-P., Brecheisen, S., Krüger, P., Pfeifle, M., and Schubert, M. 2003. Using sets of feature vectors for similarity search on voxelized CAD objects. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM Press, New York, NY, USA, 587–598.

Leifman, G., Katz, S., Tal, A., and Meir, R. 2003. Signatures of 3D models for retrieval. In *Proceedings of the 4th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics*. 159–163.

Löffler, J. 2000. Content-based retrieval of 3D models in distributed web databases by visual shape information. In *IV '00: Proceedings of the International Conference on Information Visualisation*. IEEE Computer Society, Washington, DC, USA, 82.

Loncaric, S. 1998. A survey of shape analysis techniques. *Pattern Recognition 31,* 8, 983–1001.

Lou, K., Prabhakar, S., and Ramani, K. 2004. Content-based three-dimensional engineering shape search. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*. IEEE Computer Society, Washington, DC, USA, 754–765.

McWherter, D., Peabody, M., Regli, W. C., and Shokoufandeh, A. 2001. Transformation invariant shape similarity comparison of solid models. In *ASME Design Engineering Technical Confs., 6th Design for Manufacturing Conf. (DETC 2001/DFM-21191)*.

McWherter, D., Peabody, M., Shokoufandeh, A. C., and Regli, W. 2001. Database techniques for archival of solid models. In *SMA '01: Proceedings of the 6th ACM Symposium on Solid Modeling and Applications*. ACM Press, New York, NY, USA, 78–87.

Ngu, A. H. H., Sheng, Q. Z., Huynh, D. Q., and Lei, R. 2001. Combining multi-visual features for efficient indexing in a large image database. *The VLDB Journal 9,* 4, 279–293.

Novotni, M. and Klein, R. 2001a. Geometric 3D comparison - An application. In *ECDL WS Generalized Documents*.

Novotni, M. and Klein, R. 2001b. A geometric approach to 3D object comparison. In *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*. IEEE Computer Society, Washington, DC, USA, 167–175.

Novotni, M. and Klein, R. 2003. 3D Zernike descriptors for content based shape retrieval. In *SM '03: Proceedings of the 8th ACM Symposium on Solid Modeling and Applications*. ACM Press, New York, NY, USA, 216–225.

Novotni, M. and Klein, R. 2004. Shape retrieval using 3d zernike descriptors. *Computer Aided Design 36,* 11, 1047–1062.

Ohbuchi, R., Minamitani, T., and Takei, T. 2003. Shape-similarity search of 3D models by using enhanced shape functions. In *TPCG '03: Proceedings of the Theory and Practice of Computer Graphics 2003*. IEEE Computer Society, Washington, DC, USA, 97.

Ohbuchi, R., Otagiri, T., Ibato, M., and Takei, T. 2002. Shape-similarity search of three-dimensional models using parameterized statistics. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*. IEEE Computer Society, Washington, DC, USA, 265–274.

Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. 2002. Shape distributions. *ACM Trans. Graph. 21,* 4, 807–832.

Paquet, E., Murching, A., Naveen, T., Tabatabai, A., and Rioux, M. 2000. Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication 16*, 103–122.

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

Puzicha, J., Buhmann, J. M., Rubner, Y., and Tomasi, C. 1999. Empirical evaluation of dissimilarity measures for color and texture. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*. IEEE Computer Society, Washington, DC, USA, 1165–1173.

Ronneberger, O., Burkhardt, H., and Schultz, E. 2002. 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 the 16th International Conference on Pattern Recognition*. Vol. 2. IEEE Computer Society, 290–295.

Rubner, Y., Tomasi, C., and Guibas, L. J. 1998. A metric for distributions with applications to image databases. In *ICCV '98: Proceedings of the 6th International Conference on Computer Vision*. IEEE Computer Society, Washington, DC, USA, 59–66.

Sánchez-Cruz, H. and Bribiesca, E. 2003. A method of optimum transformation of 3D objects used as a measure of shape dissimilarity. *Image and Vision Computing 21,* 11, 1027–1036.

Seidl, T. and Kriegel, H.-P. 1997. Efficient user-adaptable similarity search in large multimedia databases. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 506–515.

Shamir, A., Sharf, A., and Cohen-Or, D. 2003. Enhanced hierarchical shape matching for shape transformation. *International Journal of Shape Modeling 9,* 2.

Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. 2004. The Princeton shape benchmark. In *SMI '04: Proceedings of the Shape Modeling International 2004 (SMI'04)*. IEEE Computer Society, Washington, DC, USA, 167–178.

Shokoufandeh, A. and Dickinson, S. J. 2001. A unified framework for indexing and matching hierarchical shape structures. In *IWVF-4: Proceedings of the 4th International Workshop on Visual Form*. Springer-Verlag, London, UK, 67–84.

Shum, H.-Y., Hebert, M., and Ikeuchi:, K. 1996. On 3D shape similarity. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*. IEEE Computer Society, Washington, DC, USA, 526–531.

Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., and Zucker, S. W. 1998. Shock graphs and shape matching. In *ICCV '98: Proceedings of the 6th International Conference on Computer Vision*. IEEE Computer Society, Washington, DC, USA, 222–229.

Song, J.-J. and Golshani, F. 2002. 3D object retrieval by shape similarity. In *DEXA '02: Proceedings of the 13th International Conference on Database and Expert Systems Applications*. Springer-Verlag, London, UK, 851–860.

Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. J. 2003. Skeleton based shape matching and retrieval. In *SMI '03: Proceedings of the Shape Modeling International 2003*. IEEE Computer Society, Washington, DC, USA, 130–142.

Tangelder, J. and Veltkamp, R. 2003. Polyhedral model retrieval using weighted point sets. *International Journal of Image and Graphics 3,* 1, 209–229.

Teodoro, M. L., Phillips, G. N., and Kavraki, L. E. 2001. Molecular docking: A problem with thousands of degrees of freedom. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 960–966.

Vranić, D. 2003. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *Proceedings of the IEEE International Conference on Image Processing*. Vol. 3. IEEE, 757–760.

Vranic, D. 2004. 3D model retrieval. Ph.D. thesis, University of Leipzig, Germany.

Vranić, D. and Saupe, D. 2000. 3D model retrieval. In *Proceedings of the Spring Conference on Computer Graphics and its Applications*. Comenius University, 89–93.

Vranić, D. and Saupe, D. 2001a. 3D model retrieval with spherical harmonics and moments. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*. Springer-Verlag, London, UK, 392–397.

Vranić, D. and Saupe, D. 2001b. 3D shape descriptor based on 3D fourier transform. In *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services*. Comenius University, 271–274.

Vranić, D. and Saupe, D. 2002. Description of 3D-shape using a complex function on the sphere. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. 177–180.

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

Zaharia, T. and Prêteux, F. 2001. Three-dimensional shape-based retrieval within the MPEG-7 framework. In *Proceedings of the SPIE Conference on Nonlinear Image Processing and Pattern Analysis XII*. 133–145.

Zaharia, T. and Prêteux, F. 2002. Shape-based retrieval of 3D mesh models. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'02)*.

Zhang, C. and Chen, T. 2001. Indexing and retrieval of 3D models aided by active learning. In *MULTIMEDIA '01: Proceedings of the 9th ACM International Conference on Multimedia*. ACM Press, New York, NY, USA, 615–616.