Content-Based 3D Object Retrieval

Benjamin Bustos, Daniel Keim, Dietmar Saupe, Tobias Schreck

Abstract

3D objects are an important multimedia data type with many applications in domains such as Computer Aided Design, Simulation, Visualization, and Entertainment. Advancements in production, acquisition, and dissemination technology contribute to growing repositories of 3D objects. Consequently, there is a demand for advanced searching and indexing techniques to make effective and efficient use of such large repositories. Methods for automatically extracting descriptors from 3D objects are a key approach to this end. In this paper, we survey techniques for searching for similar content in databases of 3D objects. We address the basic concepts for extraction of 3D object descriptors which in turn can be used for searching and indexing. We sketch the wealth of different descriptors by two recently proposed schemes, and discuss methods for benchmarking the qualitative performance of 3D retrieval systems.

Keywords: 3D objects, database retrieval, similarity search, feature extraction, benchmarking, indexing.

1 Introduction

3D objects are an important multimedia data type with many application possibilities. 3D models can represent complex information, and the problem of content-based searching in large 3D object repositories arises in a number of practical fields. Example application domains include computer aided design/computer aided manufacturing (CAD/CAM), virtual reality (VR), medicine, molecular biology, military applications, entertainment, and so on.

In content-based searching and organization, the problem is to define appropriate similarity measures to automatically assess the similarity between any pair of 3D objects based on a suitable notion of similarity. The existence of such similarity measures is an important precondition for implementing effective search algorithms, which allow to query a repository of 3D objects for specific content a user is interested in, facilitating the re-usage of existing 3D content. Also, similarity metrics allow the organization of 3D repositories by means of representing large object collections by a limited number of cluster prototypes, or to visualize the content of large databases by appropriate 2D mappings.

Advanced automatic applications such as the classification of shapes in industrial screening and object recognition applications are supported by a similarity notion. For example, in medicine the detection of similar organ deformations can be used for diagnostic purposes. 3D object databases are also used to support CAD tools which have many applications in industrial design and manufacturing, and the re-usage of standard parts can lead to a reduction of production costs.

Over the last years, a range of methods for implementing similarity notions for 3D objects have been proposed. In Section 2 of this paper, we introduce important basic concepts of retrieval-oriented 3D database systems. In Section 3, we present a systematic overview over methods for characterizing 3D objects with descriptors suited for content-based 3D retrieval. Two interesting, exemplary 3D descriptors from recent research are recalled, and methods for the benchmarking of competing retrieval methods are discussed. Finally, in Section 4 we conclude the paper by a summary and an outline of open problems.

2 Basic Concepts for 3D Database Retrieval

A common characteristic of all applications in multimedia databases (and in particular for 3D object databases) is that a query searches for *similar objects* instead of performing an exact search, as in traditional relational databases. Multimedia objects cannot be meaningfully queried in the classical sense (exact search) because the probability that two multimedia objects are identical is very low, unless they are digital copies from the same source. Instead, a query in a multimedia database system usually requests a number of objects most similar to a given query object or to a manually entered query specification. Therefore, one of the most important tasks in a multimedia retrieval system is to implement effective and efficient *similarity search algorithms*.

Typically, the multimedia data are modeled as objects in a *metric* or *vector space*, where a *distance function* must be defined to compute the similarity between two objects. Thus, the similarity search problem is reduced to a search for close objects in the metric or vector space. Two common similarity queries are the *range query* (which returns all the objects within some given distance ε to the query) and the *k nearest neighbors query* (which returns the k closest objects to the query).

The primary goal in 3D similarity search is to design algorithms with the ability to effectively and efficiently execute similarity queries in 3D databases. Effectiveness is related with the ability to retrieve similar 3D objects while holding back non-similar ones, and efficiency is related with the cost of the search, measured e.g., in CPU or I/O time. But, first of all one needs to define how the similarity between 3D objects is computed. For this purpose, up to now the most widely used approach is to compare the *global geometric similarity* between two 3D objects.

One way to compute global geometric similarity is by direct geometric matching, measuring how “costly” it is to transform a given 3D object into another one. The cost associated with the transformation process serves as the metric for similarity. However, directly comparing all 3D objects from a database with a query object may be a prohibitively time consuming process, because 3D objects can be represented in many different formats and may exhibit widely varying complexity.

The *descriptor-based* approach is another way to compute the similarity between 3D objects. In this approach, numerical descriptors (also known as *feature vectors*) are extracted from the 3D objects and are used for indexing and retrieval purposes. A 3D feature vector usually characterizes the global geometry of a 3D object, and can be efficiently compared to other feature vectors to identify similar shapes and to discard dissimilar ones. For additional information on this aproach, see Figure 1.

The feature vector approach

A *metric space* is a pair $(\mathbb{X}, \delta)$ where $\mathbb{X}$ represents the universe of valid objects and $\delta : \mathbb{X} \times \mathbb{X} \to \mathbb{R}^+$ is a function between pairs of objects, that returns a positive real value (the *distance* between objects in the space) and hold the properties of a metric (strict positiveness, symmetry, and the triangle inequality). A *vector space* $\mathbb{R}^d$ is a particular type of metric space, composed by d-tuples of real numbers called *vectors*. That is, if $x \in \mathbb{R}^d$ then $x = (x_1, \ldots, x_d),\ x_i \in \mathbb{R}, 1 \leq i \leq d$.
A widely used family of distance functions for vector spaces is the *Minkowski distance* (L_p), which is defined as

$$L_p(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}, \ p \geq 1.$$

To model multimedia data as a vector space, a *transformation function*, which is highly dependent on the multimedia data type, must be used. This function extracts important features from the multimedia objects and maps these values into d-dimensional *feature vectors*. Usually, the dimensionality d of the resulting feature vector is a parameter of the transformation function: By using higher values of d it is possible to obtain a better (finer) representation of the multimedia object. However, in practical applications there is usually a *saturation point*, i.e., adding more dimensions after reaching the saturation point does not improve considerably the quality of the representation of the object. For most applications, the transformation is *irreversible*, i.e., it is not possible to reconstruct the original multimedia object from its feature vector.

Figure 1: Side-Box on feature vectors.

Apart from global geometric similarity, the notion of *local* or *partial similarity* may be important for some specific application domains. In this case, the problem is to find similarities in parts or sections of the 3D objects, or even to find complementary parts between solid object segments (e.g., in protein docking). Although this is an important research field in 3D databases, it is still not clear how to design fast segmentation methods that lead to suited 3D object partitions, which could be compared pairwise. Another approach to define the similarity of 3D objects is based on comparing the topology of the 3D objects, which can be done for example by comparing the skeletons derived from solid objects.

3 Content-based Retrieval With Descriptors

3.1 Feature Extraction Model

Candidate features for 3D description depend on the specific format in which the models in a considered database are given. A property explicitly coded in most representations is geometry, and consequentially, 3D descriptors usually rely on geometry information only. The extraction of shape descriptors generally can be regarded as a multistage process like illustrated in Figure 2 [BKS+05]. In this process, a given 3D object is first preprocessed to achieve required invariance and robustness properties. Then, the object is transformed so that its character is either of surface type, or volumetric, or captured by one or several 2D images. Then, a numerical analysis of the shape takes place, from the result of which finally the descriptor is formed.

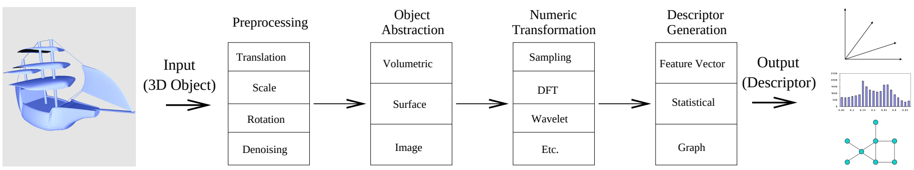


Figure 2: 3D FV extraction process model.

In the *preprocessing* stage, approximate rotation, translation, and scale invariance is aimed at, as well as improving the robustness of the description extraction with respect to noise. Ideally, an arbitrary combination of translation, rotation, scaling operations applied to one object should not affect its similarity measure with respect to another object, even in the presence of (moderate, noisy) perturbations of the models. Invariance with respect to anisotropic scaling may also be desirable. In some applications, even certain allowable shape deformations as, e.g., in articulated bodies, should be taken into account as an invariance requirement for a shape descriptor. Besides relying on preprocessing to provide these invariances, designing descriptors providing certain invariances by definition (i.e., in the numerical transform stage of the descriptor generation) is an option.

The next stage *abstracts* the model to one out of different key characteristics which can be seen in 3D shape. It may be regarded as an infinitely thin *surface* with precisely defined properties of differentiability. Alternatively, it may be seen as a thickened surface that occupies some portion of volume in 3D space, or as the boundary of a solid. The transformation of a mesh into one of the latter forms is typical for *volumetric* abstractions. A third way to capture the character of a mesh would be to project it onto one or several image planes producing renderings, corresponding depth maps, silhouettes, and so on, from which descriptors can be derived. In the *numerical transformation* stage, certain main features of the models in one of the three abstraction types are captured numerically using one of various methods. Voxel grids and image arrays can be Fourier or Wavelet transformed, and surfaces can be adaptively sampled. This yields a numerical representation of the underlying object, not necessarily allowing reconstruction of the object.

In the final stage, a *descriptor* is generated. The descriptor type itself can be classified into *feature vectors* (FVs), or *statistical*, or *graph-based* descriptors. The first two methods capture object features either in a vector of reals or statistical summarizations, and defining distance functions for them is straightforward. Graph-based descriptions are more complex in nature, and are especially useful for representing structural properties when object features can be segmented in a meaningful and robust way. On the other hand, for graph-based representations, often custom distance functions have to be developed.

Other classifications for shape description and analysis methods are possible, see for example the surveys of Tangelder and Veltcamp [TV04] or Ramani et al [IJL$^+$05]. The methods in the feature vector class are efficient, robust, easy to implement, and provide some of the most common and best approaches. This does not imply, however, that statistical or graph-based methods cannot be recommended - most of these methods have their particular strengths and may well be the ideal candidate for a specific application.

3.2 Desired Properties of the Retrieval

Several desirable properties can be identified for an efficient and effective 3D search system. Efficiency refers to the consumption of resources needed for storage and retrieval of the multimedia objects and is typically measured by system response times or storage utilization. Effectiveness typically relates to the quality of the answer objects returned by the search system, and is often assessed by metrics known from Information Retrieval. Quality of the answers measures the degree of relevance of answers with respect to the query object. An effective retrieval system is supposed to return the most relevant objects from the database on the first positions of the *k*-NN query, and to hold back irrelevant objects from this ranking.

Effectiveness and efficiency in a FV-based search system are determined primarily by the implemented FVs. Regarding efficiency, we demand the FV descriptors to be efficiently extracted from the objects and efficiently encoded, possibly by a representation providing the embedded multi-resolution property. Fast extraction makes it possible to perform database inserts on the fly, where FVs are calculated for any new object to be inserted in real time. Efficiency of representations requires the vectors to consume minimal space in terms of number of vector components and number of bits used to encode the component values. Short FVs reduce the amount of disk/memory space required to store the FVs, and speed up distance calculations and access to the vectors. Specifically, the performance of vector space index structures deteriorates quickly if the dimensionality of the indexed data grows [BBK01]. Often, there is a typical tradeoff between resolution (size) of the FVs, and the provided discrimination power, in that higher dimensionality leads to better retrieval precision. Therefore, the embedded multi-resolution property is desirable. FVs with this property encode progressively more object information inside a given FV, meaning that by considering subsets of dimensions in embedded multi resolution FVs allows to chose the level of detail of the object description. For additional information on efficiency aspects, c.f. Figure 3.

Regarding system effectiveness, it is desirable to have descriptors which provide sufficient discrimination power as well as certain invariance properties as required by the application. Discrimination power requires that an appropriate distance function defined in FV space effectively captures the similarity relationships present in object space by distances in FV space. Also, the descriptors should be robust with respect to small changes in the input 3D objects. Depending on the application, certain invariances of the search may be desired, meaning that distances in FV space should be invariant with respect to certain object transformations which are considered leaving the similarity relationships unchanged. Robustness is another effectiveness criterion often demanded, implying that small variations in the multimedia objects, e.g., caused by noise, should not dramatically alter the resulting distance between the objects in FV space.

3.3 An Image- and a Graph-Based Descriptor

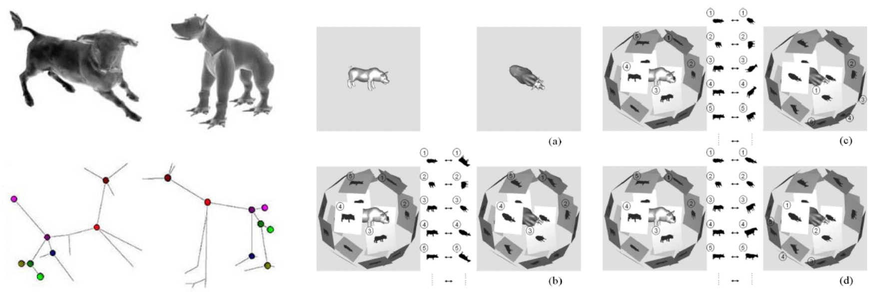
As recent surveys indicate [TV04, IJL$^+$05, BKS$^+$05], there is a wealth of different approaches to describe 3D shape for usage in a retrieval system. The situation is comparable to content-based image retrieval (CBIR), where also, many different descriptors have been proposed over the recent years. We can mention that many of the 3D descriptors currently in existence are heuristically introduced, motivated by techniques and practices from Computer Graphics (e.g., projection-based descriptors), Geometry (e.g., descriptors based on surface curvature statistics), or Signal Processing (e.g., descriptors representing object samples in the frequency domain). Usually, it is a priori unclear which of the potentially many different features should be preferred for addressing the general 3D retrieval problem. Each of the descriptors captures specific model information, and their suitability for effective retrieval needs to be experimentally evaluated. We here review two exemplary 3D descriptors which have been recently proposed [SSGD03, CTSO03], a selection intended to give the reader a feeling of the types of approaches used for shape matching.

Skeletons derived from solid objects can be regarded as intuitive object descriptions, possibly capturing important structural object information. For 3D object retrieval, suitable skeletonization algorithms and similarity functions defined on

Index structures for efficient retrieval

A naïve method to answer range and *k* nearest neighbors queries is to perform a sequential scan of the database, comparing each multimedia object directly against the query. However, this method may be too slow for real-world applications. An *index structure* can be used to filter out irrelevant objects during the similarity search without comparing them against the query, thus avoiding the sequential scan.
Several index structures have been proposed for metric and vector spaces. *Metric access methods* [CNBYM01] are index structures that use the metric properties of the distance function (especially the triangle inequality) to filter out zones of the space. *Spatial access methods* [BBK01] are index structures especially designed for vector spaces which, together with the metric properties of the distance function, use geometric information to discard points from the space. Usually, these indices are hierarchical data structures that use a balanced tree to index the database.

Figure 3: Side-Box on index structures.



(a) Skeleten-based 3D retrieval (b) Image-based 3D retrieval

Figure 4: (a) A pair of mutually best-matching objects from a 3D database, using graph-based shape description (Figure taken from [SSGD03]) (b) The LightField descriptor determines similarity between 3D objects by the maximum similarity when aligning sets of 2D projections obtained from an array of cameras surrounding the object (Figure taken from [CTSO03]).

skeletons have to be devised. In [SSGD03], the authors proposed to use skeletons obtained by connecting clusters of object voxels left after an appropriate thinning of the model voxels has taken place. The thinning method is based on the Euclidean distance transform of the voxel grid, expected to identify salient object voxels. Clusters of salient voxels are connected to form a skeleton graph, where the graph nodes are enriched by information on the underlying voxel clusters, as well as local topological properties of the skeleton. Together with an intelligent graph-matching scheme, it is then possible to calculate the (dis)similarity between any two 3D models for which skeletons have been determined (c.f. Figure 4 (a)). The authors note the method's suitability for matching articulated objects as well as the potential for finding partial matches between objects.

How intelligent retrieval of 3D models can successfully leverage 2D shape description approaches is demonstrated in [CTSO03]. The authors calculate the similarity between a pair of 3D models by comparing sets of 2D projections rendered from the model. To this end, a system of cameras distributed regularly on an imaginary sphere enclosing a 3D model is constructed. Each camera renders a 2D image of the model by means of parallel projection (c.f. Figure 4 (b)). Each projection is then described by image features extracted from the corresponding 2D silhouettes. The similarity between two objects is defined as the minimum of the sum of distances between all corresponding image pairs over the rotation of one camera system relative to the other. Together with an efficient multistage filtering approach considering increasingly more detail information from the silhouette descriptors, the system supports retrieval in large 3D databases, and provides implicit rotational invariance not requiring object orientation preprocessing. In benchmark-based precision-recall experiments, the system was shown to provide excellent retrieval performance [CTSO03].

3.4 Evaluating Retrieval Quality Using Benchmarks

To evaluate the retrieval quality of a search engine, several measures have been defined and proposed by the Information Retrieval community. Two well known effectiveness measures are *precision* and *recall*. Precision is the fraction of the retrieved objects which is relevant to a given query, and recall is the fraction of the relevant objects which has been retrieved from the database. Precision values at several recall levels may be used to produce *precision versus recall figures*, which are used for comparing the effectiveness of similarity search algorithms. In addition, another widely used effectiveness measure is the *R-precision* (also called first-tier precision), which is defined as the precision for retrieving N objects, where N is equal to the number of relevant objects to the query stored in the database. The R-precision gives a single number to rate the performance of a retrieval algorithm. This effectiveness measure is similar to the *bull eye percentage* (also called second-tier precision), defined as the recall for retrieving $2N$ objects from the database.

To compare different retrieval algorithms against each other using such evaluation measures, benchmark databases with reference queries and associated relevance information are needed. Among several 3D benchmarks proposed earlier, the well-known Princeton Shape Benchmark (PSB) [PSF04] is one of the most popular such benchmarks to date. It consists of a carefully compiled collection of about 1800 3D models harvested from the Internet. The models represent real-world objects such as e.g., vehicles, buildings, animals, or plants, and are classified according to function and shape on multiple levels of abstraction. Based on such benchmarks, experimental evaluation of 3D retrieval methods can take place. E.g., a thorough experimental effectiveness evaluation of several different 3D descriptors can be found in [BKS$^+$06]. This work showed that many of the proposed descriptors for 3D objects have good average effectiveness, and are well suited for 'general-purpose' 3D content represented by the benchmarks. Also, an international shape retrieval contest launched in 2006, SHREC, was initially built around the PSB benchmark. Please refer to Figure 5 for details on this contest.

The SHREC 3D retrieval contest

Following examples in other retrieval disciplines, also in the 3D field there is an initiative to establish an international shape retrieval contest. In 2006, chaired by Remco Veltkamp of the EC-funded Network of Excellence Aim@Shape, the *3D Shape Retrieval Contest* (SHREC) debuted at the IEEE International Conference on Shape Modeling and Applications. The initial contest was designed around the Princeton Shape Benchmark [PSF04], and in 2007 specialized toward problems involving e.g., watertight models, CAD content, and partial similarity retrieval tasks. SHREC is expected to become an objective forum for evaluating and comparing 3D retrieval algorithms, and to stimulate research on new, challenging aspects of 3D shape retrieval.

Figure 5: Side-Box on the SHREC contest.

4 Conclusions

In this paper, we gave an overview of practical approaches to similarity search in 3D object databases. In particular, feature-based techniques are widely used and offer several advantages. The extraction of features from 3D data is usually fast and easily parameterizable, and the metrics for vector spaces, as the Minkowski distances, can also be efficiently computed. Spatial access methods [BBK01] or metric access methods [CNBYM01] can be used to index the obtained feature vectors. All these advantages make the feature-based approach a good candidate for implementing a 3D object similarity search engine.

There are still many important open problems in the research area of content-based description and retrieval of 3D objects. For example, domain-specific model databases (e.g., CAD parts or models from visualization) may show specific requirements and restrictions that need to be taken into account to perform the similarity query (e.g., invariance with respect to local deformations in geometry and topology, or invariance with respect to anisotropic scaling). Thus, the similarity model used to perform the search must reflect these additional constraints or requirements.

Most of the retrieval methods developed to date restrict themselves on geometric aspects of 3D models. Conceptually, additional important object attributes such as color, material properties, and texture can be associated with 3D models. Depending on how the model was created, also structural object information or machining process information can be thought of. While these attributes offer additional information which could be exploited for content-based retrieval, the absence of a widely accepted, versatile and powerful 3D representation format makes research into multi-aspect 3D retrieval difficult in practice. For a discussion of the format problem, along other pressing challenges in managing growing amounts of 3D object data, please refer to the article by Sven Havemann and Dieter Fellner in the March 2007 issue of Computer Graphics & Applications [HF07].

The definition and effective implementation of partial similarity search notions among multimedia objects remains a big challenge. This problem is far more complex than the global geometry similarity search problem, because in partial similarity only a fraction of the 3D object is considered for the match. Even the concept of "match" in this context must be properly defined: We may want to look for similar parts, or for complementary parts.

Acknowledgments

This work was partially funded by the German Research Foundation (DFG) under Projects No. KE 740/6-1/8-1 and SA 449/10-1 within the strategic research initiative on "Distributed Processing and Delivery of Digital Documents" (V^3D^2). It was also partially funded by the Millennium Nucleus Center for Web Research, Grant P04-067-F, Mideplan, Chile (first author).

References

[BBK01] C. Böhm, S. Berchtold, and D. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.

[BKS+05] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranić. Feature-based similarity search in 3D object databases. *ACM Computing Surveys (CSUR)*, 37:345–387, 2005.

[BKS+06] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranic. An experimental effectiveness comparison of methods for 3D similarity search. *International Journal on Digital Libraries, Special Issue on Multimedia Contents and Management*, 6(1):39–54, 2006.

[CNBYM01] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Computing Surveys (CSUR)*, 33(3):273–321, 2001.

[CTSO03] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003.

[HF07] S. Havemann and D. Fellner. Seven research challenges of generalized 3d documents. *IEEE Computer Graphics and Applications*, 27(3):70–76, May/Jun 2007. (special issue on 3D documents).

[IJL+05] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37:509–530, 2005.

[PSF04] M. Kazhdan P. Shilane, P. Min and T. Funkhouser. The princeton shape benchmark. In *SMI '04: Proceedings of the Shape Modeling International 2004*, 2004.

[SSGD03] H. Sundar, Deborah Silver, Nikhil Gagvani, and Sven J. Dickinson. Skeleton based shape matching and retrieval. In *SMI '03: Proceedings of the Shape Modeling International 2003*, pages 130–142, Washington, DC, USA, 2003. IEEE Computer Society.

[TV04] J. Tangelder and R. Veltkamp. A survey of content based 3D shape retrieval methods. In *Proc. International Conference on Shape Modeling and Applications (SMI'04)*, pages 145–156. IEEE CS Press, 2004.