BY THOMAS FUNKHOUSER, MICHAEL KAZHDAN, PATRICK MIN, AND PHILIP SHILANE

# Shape-Based Retr Analysis of 3D M

*Just like Google searches online text documents, emerging systems now search repositories of 3D surface models with queries describing geometric properties.*

The number of 3D geometric models available in online repositories is growing dramatically. Examples include: the Protein Data Bank [1], which stores the 3D atomic coordinates for 29,000 protein molecules; the National Design Repository [9], which stores 3D computer-aided design (CAD) models for tens of thousands of mechanical parts; and the Princeton Shape Database [5], which stores polygonal surface models for 36,000 everyday objects crawled from the Web. Since graphics hardware is getting faster and 3D scanning hardware cheaper, there is every reason to believe that demand for and supply of 3D models will continue to increase into the future, leading to an

# ieval and
# odels

online environment in which 3D models are as plentiful as images, videos, and audio files today.

Large digital repositories of 3D models help create demand for search engines that are able to retrieve the data of interest and for data mining algorithms to discover relationships among them. Text annotation is almost always helpful, but content-based methods are often required to discover novel geometric relationships in the data. For example, a mechanical engineer might use a search engine to find a particular CAD model in a parts catalog based on its 3D shape characteristics; a

plicated than for other types of multimedia. The main difficulty is that 3D surfaces rarely have simple parameterizations. Since 3D surfaces can have arbitrary topologies, many useful methods for analyzing other media (such as Fourier analysis) have no obvious analogues for 3D surface models. Moreover, the dimensionality is higher, making searches for pose registration, feature correspondences, and model parameters more difficult, while the likelihood of model degeneracies is greater. As such, a specialized set of shape analysis methods is required for 3D data.

Here, we explore applications and technologies for shape-based retrieval and analysis of 3D models. We survey motivating applications (see the sidebar "Shape Analysis Applications"), review shape repre-



Figure 1. Harmonic shape descriptor. A robust, concise, rotation-invariant shape representation can be constructed from the amplitudes of spherical harmonic coefficients within every frequency (order) and spherical shell of a grid.
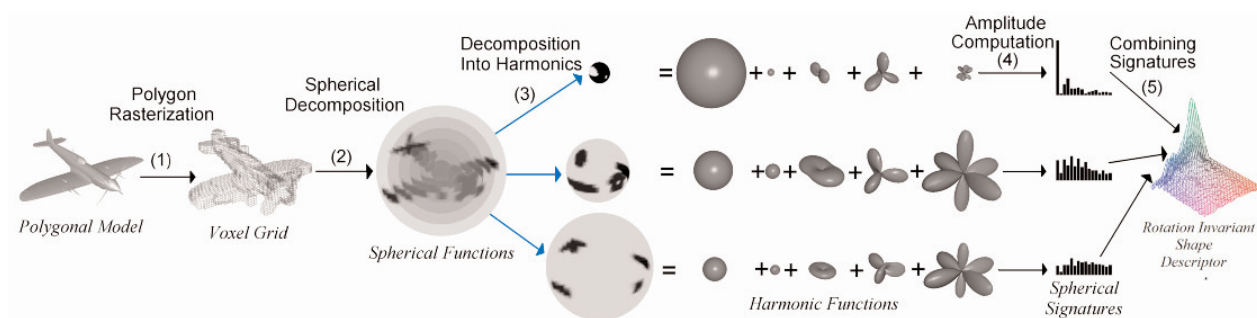
doctor might use an automatic classification system to aid diagnosis of a disease from the shapes of afflicted organs; and a paleontologist might use shape analysis to link similar 3D models scanned from animal skeletons of different species.

Such problems are common to other types of multimedia data, including sound, images, and video. However, databases of 3D models have several new and interesting characteristics that significantly affect shape-based retrieval and analysis algorithms. Unlike other multimedia data types, most 3D models do not depend on the configuration of sensors, emitters, or surrounding objects. As a result, they do not contain projections, reflections, shadows, or occlusions, greatly simplifying identification of matches among objects of the same type. For example, it is plausible to expect that the 3D model of a horse contains exactly four legs of roughly equal size. In contrast, any 2D image of the same horse may contain fewer than four legs (if some of them are occluded by, say, tall grass) or "extra legs" appear as the result of shadows on a barn or reflections in a nearby pond, or some of the legs appear smaller than others due to perspective distortions.

These problems are vexing when analyzing images but absent from most 3D models. In other respects, representing and processing 3D models is more com-

sentations, present a case study for a shape-based search engine for retrieval of 3D models crawled from the Web, and provide references to related projects and Web sites. Our aim is to introduce the problems in 3D shape analysis and provide a road map for possible solution methods.

## SHAPE ANALYSIS METHODS

Methods for computer-aided shape analysis and retrieval are being pursued in several fields, including computer vision, computational geometry, and computer graphics. Most of this work has focused on specific data types (such as CAD and range scans). General-purpose methods have also been described. Surveys can be found in [7, 12].

The primary challenge in building a shape-based retrieval and analysis system is to find a computational representation of shape (shape descriptors) for which an index can be built, similarity queries answered efficiently, and/or interesting features computed robustly. Shape-based descriptors should generally be concise; efficient to compute, compare, and search; insensitive to noise, blur, cracks, and small extra features; independent of 3D object representa-

# SHAPE ANALYSIS APPLICATIONS

3D shape analysis is important in many application domains, including molecular biology, chemistry, forensics, mechanical CAD, military target recognition, medicine, paleontology, computer graphics, computer vision, entertainment, and art. The related challenges (see the figure) are similar for each and can be characterized by the following basic problem types:

*Registration.* Given two 3D models, align them optimally; for example, an archeologist might want to align 3D scans of excavated pots in order to visualize their similarities and differences;

*Matching.* Given two 3D models, compute the geometric similarity between them; for example, a doctor might want to compare a pair of MRI scans to characterize the differences between tumors scanned in different patients or the same tumor scanned in a single patient at different times;

*Retrieval.* Given a database of 3D models and a geometric query, find the models that best match the query; for example, a kitchen designer might search a catalog for furniture that best matches a set of geometric constraints imposed by an oddly shaped room;

*Recognition.* Given a database of 3D models and a query model, either find the query model in the database or determine it is not there; for example, a criminal investigator might search a database of 3D face images scanned from known criminals to identify the ones that best match surveillance camera images;

*Verification.* Given a 3D model and a specification, determine whether they match to within some tolerance; for example, a machinist might want to detect defects in manufactured parts passing a laser scanner on a conveyer belt;

*Clustering.* Given a database of 3D models, automatically partition them into a set of classes; for example, a paleontologist might derive insight into the process of evolution by clustering 3D models of fossils scanned from different species;

*Feature detection.* Given a 3D model, find geometric features of interest on its surface; for example, an art historian might want to detect marks from a particular type of chisel on

the scanned surface of a famous statue;

*Classification.* Given a set of model class specifications and a query model, determine the class to which the query model belongs; for example, a molecular biologist might classify new protein structures based on geometric comparisons to others with known function;
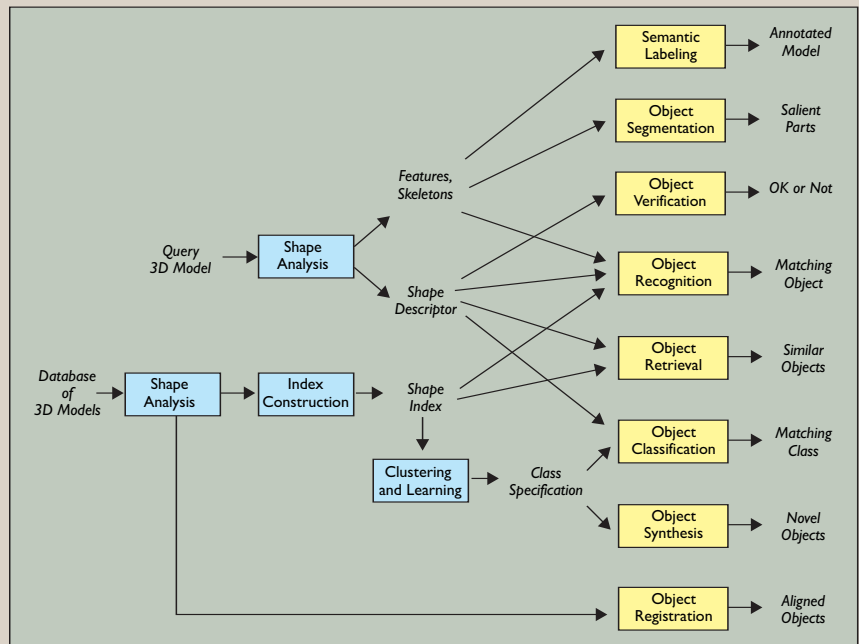


*Segmentation.* Partition a given 3D model into its salient parts; for example, an animator might want assistance in building an articulated "skeleton" model to be used for animating a human character model acquired through a full-body laser range scanner;
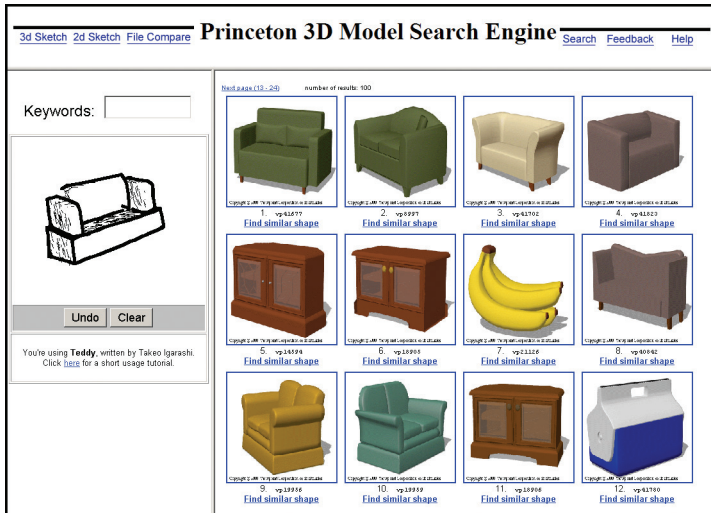
*Semantic labeling.* Infer semantic meaning regarding the purpose and function of a given 3D model; for example, a mesh-processing system might want to recognize important semantic features of a 3D model in order to avoid destroying them when the model is simplified or compressed; and

*Synthesis.* Automatically synthesize new examples typical of a given model class specification; for example, by examining 3D surfaces of several office chairs, a computer program might be able to develop a parameterized model for the geometric arrangements of their parts, then help automate construction of new chairs by assembling parts automatically. **C**

tion, tessellation, topology, or genus; invariant to transformations (such as scales, translations, rotations, mirrors, or articulations); and representative of key shape features.

Most of all, they should enable an efficient algorithm to compute the shape properties of interest to a particular application domain. Unfortunately, it is

unlikely that any single shape descriptor will satisfy all these properties, as there are usually trade-offs between computational expense and discrimination power. The spectrum of shape descriptors ranges from those that are simple to compute (but perhaps not very discriminating) to those that require expensive computations (but provide sophisticated shape analy-

sis). They are sometimes integrated into a retrieval system organized as a pipeline of conservative filters, where the early ones quickly cull irrelevant 3D models from consideration, and the later ones provide more discriminating shape analysis for smaller subsets of the remaining candidates.

At the low end of the spectrum, shapes can be represented by their statistical properties. The simplest approach describes a shape as a feature vector where the axes of a multidimensional feature space encode global geometric properties (such as genus, algebraic moments, circularity, eccentricity, and compactness) [3]. Other methods utilize histograms of geometric statistics (such as the distances between points and the angles between lines and planes). Yet others characterize statistics of frequency decompositions for shapes for, say, storing the amplitudes of spherical harmonic coefficients to provide invariance to rotations (see Figure 1). These statistical methods are generally robust, concise, quick to compute, indexable, and effective for characterizing large-scale shape features. But they also often provide invariance to a limited range of transformations (such rigid body transformations), are not invertible, and fail to capture subtle semantic details that might be valuable for discriminating among slightly different classes of shapes (such as doors with keyholes vs. doors without keyholes).

At the high end of the spectrum, shape can be represented by model-based structures that capture the geometric relationships among the key parts of an object [11]. For example, hierarchical structures can be used as templates for which their fit to a shape provides useful information for recognizing or classifying

articulated shapes. Alternatively, a shape can be decomposed into its parts automatically through algorithms that decompose its surface along concave seams, compute an approximation to its medial axis, or approximate its volume with a set of simple primitives (such as covering it with ellipsoids). These methods are generally good at capturing the topology of an object but are almost always time-consuming to compute, overly sensitive to small features, and difficult to match and/or index for retrieval applications.

In the middle of the spectrum are a plethora of approaches, including view-based representations that describe the shape of a 3D object by "how it looks" in a set of 2D projections from different views [2] and point-based methods that describe the local shape around a multitude of sample points [4]. Each method involves its own benefits and costs associated with different application domains. Comparing these shape representations is difficult [10], especially since the quality of the results is subjective. However, lower-level statistical shape descriptors are generally more successful in applications involving recognition, matching, and retrieval, while higher-level model-based shape representations are better suited for segmentation, semantic labeling, and synthesis applications.

## RETRIEVING 3D MODELS

Perhaps the most important application of 3D shape analysis is the retrieval of 3D models available on the Web. Several such search engines have recently been introduced for indexing 3D data for computer graphics, molecular biology, and mechanical CAD. One of the earliest was the Princeton 3D Model Search Engine, which went online in 2001 and today indexes 36,000 computer graphics models crawled from the Web. It was developed to help people looking for 3D polygonal surface models (such as cars and humans)—in other words, a Google for 3D models.

The Princeton search engine is available as a Web page (shape.cs.princeton.edu) in which the user enters queries, and the system retrieves the best-matching 3D models from a Web-crawled repository, presenting them as a ranked set of thumbnail images (see Figure 2). If the user clicks on any returned thumbnail image, the system produces a pop-up Web page with detailed information and close-up images of the associated 3D model, along with a link for downloading the 3D model file from its original Web address.

Since different types of objects are best described

overlapping, and intersecting polygons) and widely varying shapes within the same class of objects. Yet matches must be found within a second or so to provide an interactive response to every query. Thus, robustness, transformation invariances, and speed are the primary concerns when choosing a shape representation for this application domain.

The search engine uses a statistical shape descriptor representing the shape of each 3D model by a 2D feature vector describing "how much shape" resides within each spherical harmonic frequency and each distance from the model's center of mass, as in Figure 1 [6]. This representation is insensitive to mesh degeneracies, since the 3D function from which spherical harmonic coefficients are computed depends only on distance from the surface, not on the tessellation of the mesh. The representation is also invariant to the model's orientation; thus, comparison of the descriptors can be performed without aligning the models a priori or searching over all possible rotations. Third, it provides the important theoretical guarantee that the Euclidean distance between two descriptors provides a lower bound on the Euclidean distance between the corresponding 3D models. The guarantee means there are no false negatives when the descriptor is used for narrowing down a set of candidate matches to be examined later in greater detail. Finally, it enables efficient indexing based on nearest-neighbor search structures. Empirically, for queries to the search engine, the system takes less than one second to find the 16 best matches in a database with 36,000 3D structures. It retrieves better matches than many larger and computationally more expensive descriptors.

The search engine has deployed these query interfaces and retrieval methods [8], so far receiving 770,000 queries from 160,000 different hosts and 130 different countries and been slashdotted twice. Thus, it is now possible to characterize how the system is used in vivo. As you might expect, the majority of queries (65%) include only text keywords; text is the most familiar type of query for most users and the quickest to enter. However, one-third of the queries have incorporated some sort of shape, most often as "find similar shape" (13% of all queries) and 2D sketches (16% of all queries).

While it is difficult to say which query types users

through different query types, the system supports many different query interfaces. Perhaps the simplest is a text box (see Figure 2 upper left) in which the user enters keywords (such as "car"), and the system retrieves 3D models with associated text containing them (such as matching words in the file name, the referring Web page, and the scene graph node names). Alternatively, the user can sketch 2D outlines of the desired object, possibly as seen from multiple viewpoints; the system then returns 3D models whose projections match the sketches. Finally, the user can specify a query with a 3D shape, either by uploading an existing 3D model, sketching a coarse 3D model from scratch, as in Figure 2, or clicking on the "Find Similar Shapes" link under any previously retrieved thumbnail image. The system then matches the shape of the 3D query to the models in the repository and returns the best matches.

Each of these query types requires its own indexing and retrieval strategy. Here, we consider the 3D shape queries (see [5] for details on the others) where the problem is: Given a repository of 3D models, build an index that can be searched efficiently to find the closest matches to an arbitrary 3D query model. The problem is difficult because computer graphics models found on the Web can appear in any coordinate system, with arbitrary degeneracies (disconnected,

A mechanical engineer might use a search engine to find a particular CAD model in a parts catalog based on its 3D shape characteristics.

deployed to perform robust feature detection, partial shape matching, part decomposition, and eventually complete semantic labeling. 3D models will then provide not only raw 3D geometry and surface attributes but smarts regarding their composition, how they move, and how they are used—all keys to making 3D data a more useful part of the information revolution. **C**

## REFERENCES
1. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., and Bourne, P. The Protein Data Bank. *Nucleic Acids Research 28* (2000), 235–242.
2. Chen, D.-Y., Ouhyoung, M., Tian, X.-P., and Shen, Y.-T. On visual similarity-based 3D model retrieval. *Comput. Graphics Forum 22,* 3 (Sept. 2003), 223–232.
3. Corney, J., Rea, H., Clark, D., Pritchard, J., Breaks, M., and MacLeod, R. Coarse filters for shape matching. *IEEE Comput. Graphics Applic. 22,* 3 (May/June 2003), 65–74.
4. Frome, A., Huber, D., Kolluri, R., Bulow, T., and Malik, J. Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision* (Prague, May 11–14). Springer-Verlag, 2004, 224–237.
5. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D. A search engine for 3D models. *Transact. Graphics 22,* 1 (Jan. 2003), 83–105.
6. Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of the Symposium on Geometry Processing* (Aachen, Germany, June 2003), 156–164.
7. Loncaric, S. A survey of shape analysis techniques. *Pattern Recognition 31,* 8 (Aug. 1998), 983–1001.
8. Min, P., Halderman, J., Kazhdan, M., and Funkhouser, T. Early experiences with a 3D model search engine. In *Proceedings of the Eighth International Conference on 3D Web Technology* (St. Malo, France, Mar. 9–12). ACM Press, New York, 2003, 7–18.
9. Regli, W. and Gaines, D. A repository of designs for process and assembly planning. *Computer Aided Design 29,* 12 (Dec. 1997), 895–905.
10. Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. The Princeton shape benchmark. In *Proceedings of Shape Modeling International* (Genoa, Italy, June 7–9). IEEE, 2004, 167–178.
11. Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. Skeleton-based shape matching and retrieval. In *Proceedings of Shape Modeling International* (Seoul, Korea, May 12–15). IEEE, 2003, 130–142.
12. Tangelder, J. and Veltkamp, R. A survey of content-based 3D shape retrieval methods. In *Proceedings of Shape Modeling International* (Genoa, Italy, June 7–9). IEEE, 2004, 145–156.

find most effective, we have found that text and shape queries augment each other in controlled user studies [5]. People tend to be most effective when starting out with a text query (to retrieve an example from a desired class of objects), then using "find similar shape" to find other models of the same type that were not well-annotated. Overall, users seem to find the search engine a useful tool for quickly identifying and retrieving 3D models on the Web.

## CONCLUSION
Shape-based retrieval and analysis tools are proving useful in a number of application domains, most notably computer graphics, mechanical computer-aided design, and molecular biology. Looking forward, these tools will be increasingly important, as 3D data acquisition hardware becomes more of a commodity, and more people begin to make and use 3D models in their everyday lives. It will then be easier to find and retrieve an existing model made by someone else than it will be to make a new one from scratch yourself. New methods will soon be

**THOMAS FUNKHOUSER** (funk@cs.princeton.edu) is an associate professor in the Department of Computer Science at Princeton University, Princeton, NJ.
**MICHAEL KAZHDAN** (misha@cs.jhu.edu) is an assistant professor in the Department of Computer Science at John Hopkins University, Baltimore, MD.
**PATRICK MIN** (min@cs.princeton.edu) is an assistant professor in the Department of Computer Science at John Cabot University, Rome, Italy.
**PHILIP SHILANE** (pshilane@cs.princeton.edu) is a graduate student in the Department of Computer Science at Princeton University, Princeton, NJ.