

Content Based 3D Shape Retrieval

Remco C. Veltkamp

Department of Information and Computing Sciences, Utrecht University, The Netherlands

Johan W.H. Tangelder

Centre for Mathematics and Computer Science, Amsterdam, The Netherlands

Recent developments in techniques for modeling, digitizing and visualizing 3D shapes has led to an explosion in the number of available 3D models on the Internet and in domain-specific databases. Currently, large archives recording cultural heritage models like the David of Michelangelo [11], and the Minerva of Arezzo [7] are obtained by laser scanning projects. Also archives containing domain-specific shape models are now accessible on the Internet. Examples are the National Design Repository, an online repository of CAD models, and the Protein Data Bank, an online archive of structural data of biological macromolecules. These developments has led to the design of 3D shape retrieval systems that, given a query object, retrieve similar 3D objects. In visualization, 3D shapes are often represented as a surface, in particular polygonal meshes, for example in VRML format. Often these models contain holes, intersecting polygons, are not manifold, and do not enclose a volume unambiguously. By contrast, 3D volume models, such as solid models produced by CAD systems, or voxels models, enclose a volume properly. Unlike text documents, 3D models are not easily retrieved. Attempting to find a 3D model using textual annotation and a conventional text-based search engine would not work in many cases. The annotations added by human beings depend on language, culture, age, sex, and other factors. They may be too limited or ambiguous. In contrast, content based 3D shape retrieval methods, that use shape properties of the 3D models to search for similar models, work better than text based methods [14].

In content based shape retrieval, the following terminology is often used. Shape matching is the process of determining how similar two shapes are. This is often done by computing a distance. A complementary process is indexing. Here, indexing is understood to be the process of building a datastructure to speed up the search. Note that also “indexing” is also used as a term for the identification of features in models, or multimedia documents in general. Retrieval is the process of searching and delivering the query results. Matching and indexing are often part of the retrieval process.

Recently, a lot of researchers have investigated the specific problem of content based 3D shape retrieval. Also, an extensive amount of literature can be found in the related fields of computer vision, object recognition, geometric modeling, computer-aided design and engineering. Survey papers to this literature have been provided by Besl and Jain [1], Loncaric [12], Campbell and Flynn [3] and Mamic and Bennamoun [13]. Iyer et al. [8] provide an extensive overview of 3D shape searching techniques especially relevant for CAD and engineering. The survey by Cardone et al. [4] primarily focuses on shape similarity methods that are suitable to compare CAD models in the context of product design and manufacturing applications. Shilane et al. [18] compare 12 shape matching methods with respect to processing time, storage requirements and discriminative power.

In addition to these surveys, this chapter evaluates 3D shape retrieval methods with respect to the following requirements: (1) shape representation requirements, (2) properties of dissimilarity measures, (3) efficiency, (4) discrimination abilities, (5) ability to perform partial matching, (6) robustness, and (7) necessity of pose normalization.

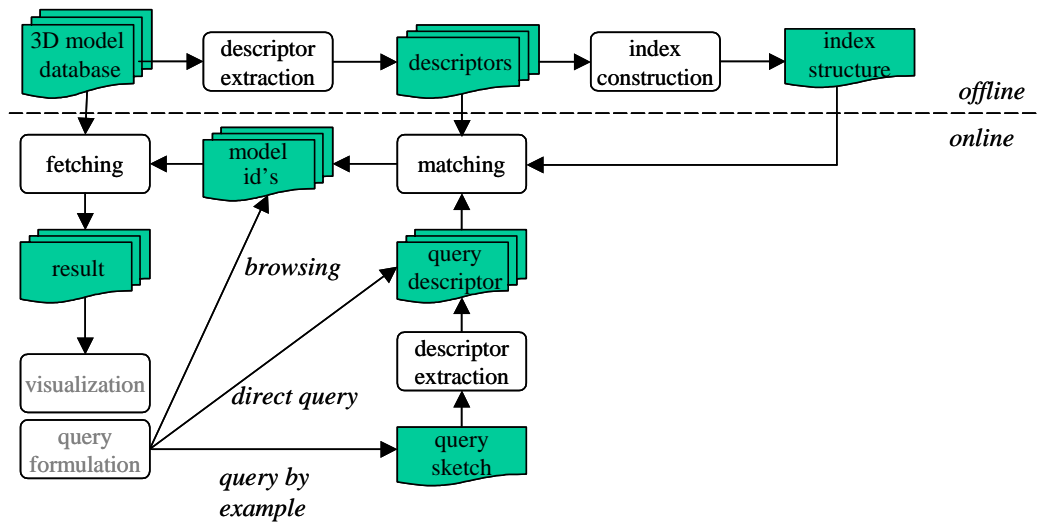


Figure 1: Conceptual framework for shape retrieval.

3D shape retrieval aspects

At a conceptual level, a typical 3D shape retrieval framework as illustrated by fig. 1 consists of a database with an index structure created offline, and an online query engine. Each 3D model has to be identified with a shape descriptor, providing a compact overall description of the shape. To efficiently search a large collection online, an indexing data structure and searching algorithm should be available. The online query engine computes the query descriptor, and models similar to the query model are retrieved by matching descriptors to the query descriptor from the index structure of the database. The similarity between two descriptors is quantified by a dissimilarity measure. Three approaches can be distinguished for providing a query object: (1) browsing to select a new query object from the obtained results, (2) a direct query by providing a query descriptor, (3) query by example by providing an existing 3D model, by creating a 3D shape query from scratch using a 3D tool, or by sketching 2D projections of the 3D model. Finally, the retrieved models can be visualized.

Shape representations

An important issue is the type of shape representation(s) that a shape retrieval system accepts. Most of the 3D models found on the World Wide Web are meshes defined in a file format supporting visual appearance. Currently, the most common format used for this purpose is the Virtual Reality Modeling Language (VRML) format. Since these models have been designed for visualization, they often contain only geometry and appearance attributes. Often, they are represented by “polygon soups”, consisting of unorganized sets of polygons. Also, in general these models are not “watertight” meshes, i.e. they do not enclose a volume. By contrast, for volume models retrieval methods depending on a properly defined volume can be applied.

Measuring similarity

In order to measure how similar two objects are, it is necessary to compute distances between pairs of descriptors using a dissimilarity measure. Although the term similarity is often used, dissimilarity better corresponds to the notion of distance: small distance means small

dissimilarity, and large similarity. A dissimilarity measure can be formalized by a function defined on pairs of descriptors indicating the degree of their resemblance. Formally speaking, a dissimilarity measure d on a set S is a non-negative valued function $d: S \times S \rightarrow \mathbf{R}^+ \cup \{0\}$. Function d may have metric properties: identity, positivity, symmetry, and triangle inequality. Additionally, transformation invariance is often a useful property.

The identity property says that a shape is completely similar to itself, while the positivity property claims that different shapes are never completely similar. This property is very strong for a high-level shape descriptor, and is often not satisfied. However, this is not a severe drawback, if the loss of uniqueness depends on negligible details. Symmetry is not always wanted. Indeed, human perception does not always find that shape x is equally similar to shape y , as y is to x . In particular, a variant x of prototype y , is often found to be more similar to y than vice versa. The triangle inequality can be applied to make retrieval more efficient [21]. However, dissimilarity measures for partial matching, giving a small distance if a part of one matches a part of the other, do not obey the triangle inequality. Transformation invariance has to be satisfied, if the comparison and the extraction process of shape descriptors have to be independent of the location, orientation and/or scale of the object in its Cartesian coordinate system.

Efficiency

For large shape collections, it is inefficient to sequentially match all objects in the database with the query object. Because retrieval should be fast, efficient indexing search structures are needed to support efficient retrieval. Since for “query by example” the shape descriptor is computed online, it is reasonable to require that the shape descriptor computation is fast enough for interactive querying.

Discriminative power

A shape descriptor should capture properties that discriminate objects well. However, the judgment of the similarity of the shapes of two 3D objects is somewhat subjective, depending on the user preference or the application at hand. E.g. for CAD models often topological properties such as the numbers of holes in a model are more important than minor differences in shape. On the contrary, if a user searches for models looking roughly similar, then the existence of a small hole in the model may be of no importance to the user.

Partial matching

In contrast to global shape matching, partial matching finds a shape of which a part is similar to a part of another shape. Partial matching can be applied if 3D shape models are not complete, e.g. for objects obtained by laser scanning from one or two directions only. Another application is the search inside “3D scenes” containing an instance of the query object. Also, this feature can potentially give the user flexibility towards the matching problem, if parts of interest of an object can be selected or weighted by the user.

Robustness and sensitivity

It is often desirable that a shape descriptor is insensitive to noise and small extra features, and robust against arbitrary topological degeneracies, e.g. if it is obtained by laser scanning. Therefore, small changes in a shape should result in small changes in the shape descriptor. On the other hand, if large changes in the shape of the object result in very small changes in the shape descriptor, then the shape descriptor is considered not sensitive. Poor sensitivity will lead to poor discriminative abilities. Also, if a model is given in multiple levels-of-detail, representations of different levels should not differ significantly from the original model.

Pose normalization

In the absence of prior knowledge, 3D models have arbitrary scale, orientation, and position in the 3D space. Because not all dissimilarity measures are invariant under scale, translation, or rotation, one or more of the normalization procedures described below may be necessary. The normalization procedure depends on the center of mass, which is defined as the center of its surface points.

To normalize a 3D model for scale, the average distance of the points on its surface to the center of mass should be scaled to a constant. Note that normalizing a 3D model by scaling its bounding box is sensitive to outliers. To normalize for translation the center of mass is translated to the origin.

To normalize a 3D model for rotation usually the principal component analysis (PCA) method is applied. It aligns the principal axes to the x -, y -, and z -axes of a canonical coordinate system by an affine transformation based on a set of surface points, e.g. the set of vertices of a 3D model. After translation of the center of mass to the origin, a rotation is applied so that the largest variance of the transformed points is along the x -axis. Then a rotation around the x -axis is carried out such that the maximal spread in the yz -plane occurs along the y -axis.

A problem is that differing sizes of triangles are not taken into account which may cause very different results for models that are identical except for finer triangle resolution in some parts of the model. To address this issue, one may use appropriately chosen vertex weights. The PCA has been generalized to the continuous PCA (CPCA) such that all of the (infinitely many) points on the mesh surface are equally relevant to compute the transformation aligning the principal axes to a canonical coordinate system [22].

Another problem with the PCA algorithm is that due to the lack of information about the direction of the principal axes, either the positive or the negative axes are moved to the x -, y - and z -axes. This results in four valid configurations that align the principal axes (four configurations are not valid because they do not define a proper coordinate system).

The PCA algorithm and its variants for pose estimation are fairly simple and efficient. However, if the eigenvalues are equal, principal axes may still switch, without affecting the eigenvalues. Similar eigenvalues may imply an almost symmetrical mass distribution around an axis (e.g. nearly cylindrical shapes) or around the center of mass (e.g. nearly spherical shapes).

Shape matching methods

Based on the representation of the shape descriptor we divide shape matching methods in three broad categories: (1) feature based methods, (2) graph based methods and (3) geometry based methods. Fig. 2 shows a more detailed categorization of shape matching methods. Note that the classes of these methods are not completely disjoint. For instance, a graph-based shape descriptor, may be extended to describe shape properties not related to topology. In these cases we categorized the method by the most characteristic aspect of its representation. Recently, a number of approaches to combine multiple shape matching methods have been introduced. Also, a number of techniques have been introduced that improve shape matching in general.

Feature based methods

In the context of 3D shape matching, features denote geometric and topological properties of 3D shapes. So, 3D shapes can be discriminated by measuring and comparing their features. Feature based methods can be divided into four categories according to the type of shape features used: (1) global features, (2) global feature distributions, (3) spatial maps, and (4) local features. Feature based methods from the first three categories represent features of a shape using a single descriptor consisting of a d -dimensional vector of values, where the dimension d is fixed for all shapes. The value of d can easily be a few hundred.

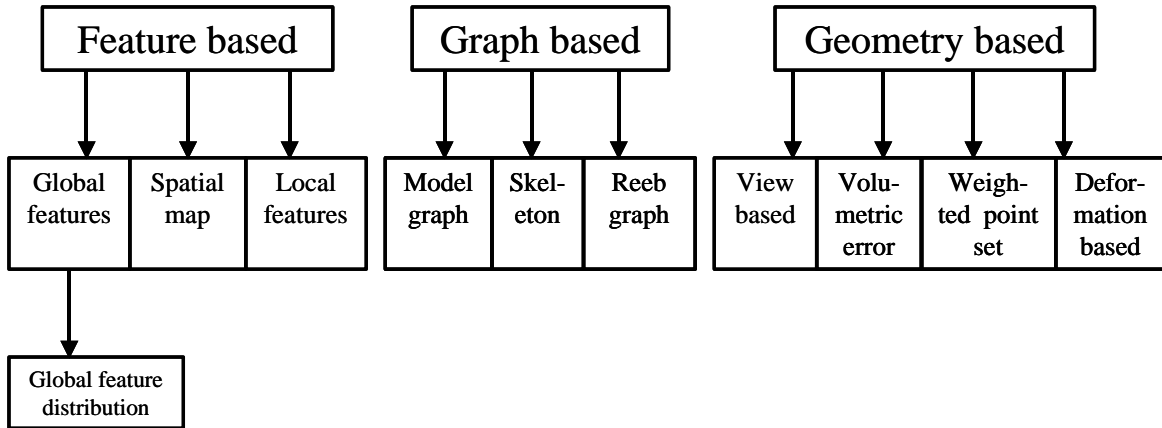


Figure 2: Taxonomy of shape matching methods.

The descriptor of a shape is a point in a high dimensional space, and two shapes are considered to be similar if they are close in this space. Retrieving the k best matches for a 3D query model is equivalent to solving the k -nearest neighbors problem in a high-dimensional space. Although this problem is known to be hard in the worst case, matching feature descriptors can be done efficiently in practice by searching to solve the “approximate k -nearest neighbor” problem.

Global features characterize the global shape of a 3D model. Examples of these features are the extended Gaussian image, statistical moments of the boundary or the volume of the model, volume-to-surface ratio, or the Fourier transform of the volume or the boundary of the shape.

Global feature distributions are a refinement of the concept of global feature based similarity. These methods compare global distributions of features, so-called shape distributions. Examples include distance, angle, area and volume measurements between random surface points [17]. The similarity between two objects is then some distances between distributions.

Spatial maps are representations that capture the spatial location of an object. The map entries correspond to physical locations or sections of the object, and are arranged in a manner that preserves the relative positions of the features in an object. Spatial maps are in general not invariant to rotations, except for specially designed maps. Therefore, typically a pose normalization is done first. A general approach to transform rotation dependent shape descriptors into rotation independent ones is based on spherical harmonics, often as a function on a voxel grid [10,22]. From a collection of spherical functions on a grid, a rotation invariant descriptor is computed by (1) decomposing the function into its spherical harmonics, (2) summing the harmonics within each frequency, and computing the distance for each frequency component. The resulting shape descriptor is a 2D histogram indexed by radius and frequency values.

Local feature based methods describe the 3D shape around a number of surface points. For this purpose, a descriptor for each surface point is used instead of a single descriptor. For example, ‘spin images’ are 2D histograms of the surface locations around a point [9]. Typically, local methods are suitable for recognizing models in a cluttered and occluded 3D scene.

Graph-based methods

In general, the feature based methods discussed in the previous section take into account only the pure geometry of the shape. In contrast, graph based methods attempt to extract a geometric meaning from a 3D shape using a graph showing how shape components are linked together. Graph based methods can be divided into three broad categories according to the type of graph used: (1) model graphs, (2) Reeb graphs, and (3) skeletons. Model graphs are extracted from

solid model representations [6] used by most CAD systems, while skeletal graph and Reeb graph approaches [2] are applicable to volume models including models of natural shapes. Efficient computation of existing graph metrics for general graphs is not trivial: computing the edit distance is NP-hard and computing the maximal common subgraph is even NP-complete.

Geometry based methods

The main idea of view based similarity methods is that two 3D models are similar, if they look similar from all viewing angles. A natural application of this paradigm is the implementation of query interfaces based on defining a query by one or more sketches showing the query from different views. The so-called lightfield descriptor [5] compares ten silhouettes of the 3D shape obtained from ten viewing angles distributed evenly on the viewing sphere. Each silhouette is a 2D image, encoded by its Zernike moments and Fourier descriptors. The dissimilarity of two shapes is found as the minimal dissimilarity obtained by rotating the viewing sphere of one lightfield descriptor relative to the other lightfield descriptor. Similar methods use depth buffers from various viewing directions.

Several other methods describe a geometry similarity approach to 3D shape matching based on calculating a volumetric error between two objects, e.g. [16]. To compute that efficiently, they are typically voxelized. Another approach is based on shape descriptors consisting of weighted 3D points. The points are typically samples, and the weights are the amount of local volume or curvature, see e.g. [19]. Shape deformation or evolution based similarity is mostly applied to 2D shapes. Because they often depend on a parameterization, they are difficult to generalize to 3D.

Comparison

Twelve different shape descriptors have been compared using the Princeton Shape Benchmark [18]. The results show that shape matching algorithms do not perform equally well on all object types. For instance, extended Gaussian images are good at discriminating between man-made and natural objects, but not that good at making detailed class distinctions. They also find that the lightfield descriptor [5] is the most discriminating between the 12 shape descriptors tested, but at higher storage and computational costs than most other descriptors. [22] compares a number of shape matching methods using the Princeton shape benchmark, the publicly-available database at the University of Konstanz, and the MPEG-7 test set [15]. These experimental tests show that from the descriptors proposed by others, the spherical harmonics approach [10] significantly outperforms all other approaches. However, the light field descriptor was not included in these experiments. Furthermore, [22] investigates a number of hybrid shape matching methods obtained by combining shape matching methods. In this framework, similarity is measured using a weighted combination of the similarity of the separate methods, where the weight of each separate method is proportional to the dimension of its feature vector. Experiments show that for comparing overall performance, a combination of the depth buffer-based descriptor, a silhouette-based descriptor, and a spherical descriptor performs best.

For a more complete survey, see [20].

References

1. Besl P. J. and R. C. Jain. Three-dimensional object recognition. *Computing Surveys* 17(1), 1985, 75-145.
2. Biasotti S., S. Marini, M. Mortara, and G. Patané. An overview of properties and efficacy of topological skeletons in shape modelling. In: *Proc. Shape Modeling International*, 2003, pp. 245-254.

3. Campbell R. J. and P. J. Flynn. A Survey of Free-Form Object Representation and Recognition Techniques. *Computer Vision and Image Understanding* 81(2), 2001, 166-210.
4. Cardone A., S. K. Gupta, and M. Karnik. A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications. *Journal of Computing and Information Science in Engineering* 3, 2003, 109-118.
5. Chen D.-Y., X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On Visual Similarity Based 3D Model Retrieval. *Computer Graphics Forum (EG 2003 Proceedings)* 22(3), 2003.
6. Cicirello V., W. C. Regli. Machining Feature-based Comparisons of Mechanical Parts. In: *Proc. Solid Modeling 2001*, 176-185.
7. Fontana R., M. Greco, M. Materazzi, E. Pampaloni, L. Pezzati, C. Rocchini, and R. Scopigno. Three-dimensional modelling of statues: the Minerva of Arezzo. *Cultural Heritage Journal* 3(4), 2002, 325-331.
8. Iyer N., K. Lou, S. Janyanti, Y. Kalyanaraman, and K. Ramani. Three Dimensional Shape Searching: State-of-the-art Review and Future Trends. Accepted for publication in *Computer Aided Design*.
9. Johnson A. E., M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 1999, 635-651.
10. Kazhdan M., T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In: *Proc. Symposium on Geometry Processing*, 2003.
11. Levoy M., K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D Scanning of Large Statues. In: *Proc. SIGGRAPH 2000*, pp. 131-144.
12. Loncaric S. A Survey of Shape Analysis Techniques. *Pattern Recognition* 31(8), 1998, 983-1001.
13. Mamic G. and M. Bennamoun. Representation and Recognition of 3D Free-Form Objects. *Digital Signal Processing* 12, 2002, 47-67.
14. Min P., M. Kazhdan, and T. Funkhouser. A Comparison of Text and Shape Matching for Retrieval of Online 3D Models. In: *Proc. European Conference on Digital Libraries*, 2004, pp. 209-220.
15. MPEG-7 Video Group. Description of Core Experiments for Motion and Shape. Technical report, ISO/IEC N3397, 2000.
16. Novotni M, R. Klein. A Geometric Approach to 3D Object Comparison, *Proc. Shape Modeling International*, 2001, 154-166.
17. Osada R., T. Funkhouser, B. Chazelle, D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4), 2002, 807-832.
18. Shilane P., M. Kazhdan, P. Min, and T. Funkhouser. The Princeton Shape Benchmark. In: *Proc. Shape Modeling International*, 2004. pp. 157-166.
19. Tangelder J. W. H., R. C. Veltkamp. Polyhedral Model Retrieval Using Weighted Point Sets. *International Journal of Image and Graphics*, 3(1), 2003, 209-229.
20. Tangelder J. W. H., R. C. Veltkamp. A Survey of Content Based 3D Shape Retrieval Methods. Submitted to *Multimedia Tools and Applications*, 2005.
21. Vleugels J. and R. C. Veltkamp. Efficient Image Retrieval through Vantage Objects. *Pattern Recognition* 35(1), 2002, 69-80.
22. Vranić D. V. 3D Model Retrieval. Ph.D. thesis, University of Leipzig, 2004.