A Survey of Content Based 3D Shape Retrieval Methods

Johan W.H. Tangelder and Remco C. Veltkamp
Institute of Information and Computing Sciences, Utrecht University
hanst@cs.uu.nl, Remco.Veltkamp@cs.uu.nl

Abstract

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. This has led to the development of 3D shape retrieval systems that, given a query object, retrieve similar 3D objects. For 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. On the contrary, 3D volume models, such as solid models produced by CAD systems, or voxels models, enclose a volume properly. This paper surveys the literature on methods for content based 3D retrieval, taking into account the applicability to surface models as well as to volume models. The methods are evaluated with respect to several requirements of content based 3D shape retrieval, such as: (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. Finally, the advantages and limits of the several approaches in content based 3D shape retrieval are discussed.

1. Introduction

The advancement of modeling, digitizing and visualizing techniques for 3D shapes has led to an increasing amount of 3D models, both on the Internet and in domain-specific databases. This has led to the development of the first experimental search engines for 3D shapes, such as the 3D model search engine at Princeton university [2, 57], the 3D model retrieval system at the National Taiwan University [1, 17], the Ogden IV system at the National Institute of Multimedia Education, Japan [62, 77], the 3D retrieval engine at Utrecht University [4, 78], and the 3D model similarity search engine at the University of Konstanz [3, 84].

Laser scanning has been applied to obtain archives recording cultural heritage like the Digital Michelangelo Project [25, 48], and the Stanford Digital Formae Urbis Romae Project [75]. Furthermore, archives containing domain-specific shape models are now accessible by the Internet. Examples are the National Design Repository, an online repository of CAD models [59, 68], and the Protein Data Bank, an online archive of structural data of biological macromolecules [10, 80].

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 [58].

Matching is the process of determining how similar two shapes are. This is often done by computing a distance. A complementary process is indexing. In this paper, indexing is understood as the process of building a datastructure to speed up the search. Note that the term indexing is also often used 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 and geometric modelling. Survey papers to this literature have been provided by Besl and Jain [11], Loncaric [50] and Campbell and Flynn [16]. For an overview of 2D shape matching methods we refer the reader to the paper by Veltkamp [82]. Unfortunately, most 2D methods do not generalize directly to 3D model matching. Work in progress by Iyer et al. [40] provides an extensive overview of 3D shape searching techniques. Atmosukarto and Naval [6] describe a number of 3D model retrieval systems and methods, but do not provide a categorization and evaluation.

In contrast, this paper evaluates 3D shape retrieval methods with respect to several requirements on content based 3D shape retrieval, such as: (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

0-7695-2075-8/04 $20.00 © 2004 **IEEE**

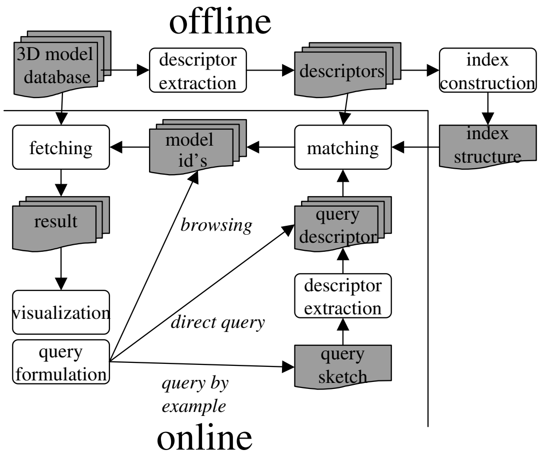

Figure 1. Conceptual framework for shape retrieval.

normalization. In section 2 we discuss several aspects of 3D shape retrieval. The literature on 3D shape matching methods is discussed in section 3 and evaluated in section 4.

2. 3D shape retrieval aspects

In this section we discuss several issues related to 3D shape retrieval.

2.1. 3D shape retrieval framework

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 to provide 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 or by creating a 3D shape query from scratch using a 3D tool or sketching 2D projections of the 3D model. Finally, the retrieved models can be visualized.

2.2. 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. In particular, 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.

2.3. 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 corresponds to the notion of distance: small distances 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 $d : S \times S \rightarrow \mathbb{R}^+ \cup \{0\}$. Function d may have some of the following properties:

i. *Identity*: For all $x \in S, d(x, x) = 0$.
ii. *Positivity*: For all $x \neq y$ in $S, d(x, y) > 0$.
iii. *Symmetry*: For all $x, y \in S, d(x, y) = d(y, x)$.
iv. *Triangle inequality*:
For all $x, y, z \in S, d(x, z) \leq d(x, y) + d(y, z)$.
v. *Transformation invariance*: For a chosen transformation group G, for all $x, y \in S, g \in G, d(g(x), g(y)) = d(x, y)$.

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 more similar to y then vice versa [81].

Dissimilarity measures for partial matching, giving a small distance $d(x, y)$ if a part of x matches a part of y, 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 place, orientation and scale of the object in its Cartesian coordinate system. If we want that a dissimilarity measure is not affected by any transformation on x, then we may use as alternative formulation for (v): *Transformation invariance*: For a chosen transformation group G, for all $x, y \in S, g \in G, d(g(x), y) = d(x, y)$.

When all the properties (i)-(iv) hold, the dissimilarity measure is called a *metric*. Other combinations are possible: a pseudo-metric is a dissimilarity measure that obeys (i), (iii) and (iv) while a semi-metric obeys only (i), (ii) and (iii). If a dissimilarity measure is a pseudo-metric, the triangle inequality can be applied to make retrieval more efficient [7, 83].

2.4. 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.

2.5. Discriminative power

A shape descriptor should capture properties that discriminate objects well. However, the judgement 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 solid modeling applications often topology properties such as the numbers of holes in a model are more important than minor differences in shapes. On the contrary, if a user searches for models looking visually similar the existence of a small hole in the model, may be of no importance to the user.

2.6. 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 for “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.

2.7. Robustness

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. Also, if a model is given in multiple levels-of-detail, representations of different levels should not differ significantly from the original model.

2.8. 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 rotation and translation, it may be necessary to place the 3D models into a canonical coordinate system. This should be the same for a translated, rotated or scaled copy of the model.

A natural choice is to first translate the center to the origin. For volume models it is natural to translate the center of mass to the origin. But for meshes this is in general not possible, because they have not to enclose a volume. For meshes it is an alternative to translate the center of mass of all the faces to the origin. For example the Principal Component Analysis (PCA) method computes for each model the principal axes of inertia e_1, e_2 and e_3 and their eigenvalues λ_1, λ_2 and λ_3, and make the necessary conditions to get right-handed coordinate systems. These principal axes define an orthogonal coordinate system (e_1, e_2, e_3), with $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Next, the polyhedral model is rotated around the origin such that the coordinate system (e_x, e_y, e_z) coincides with the coordinate system(e_1, e_2, e_3).

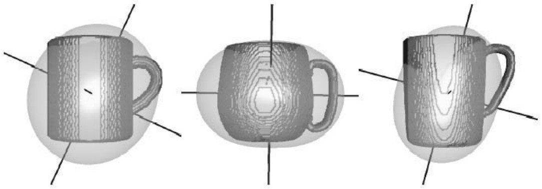

Figure 2. Similar mugs oriented by principal axes in different ways [30].

The PCA algorithm for pose estimation is fairly simple and efficient. However, if the eigenvalues are equal, principal axes may 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). Fig. 2 illustrates the problem.

3. Shape matching methods

In this section we discuss 3D shape matching methods. We divide shape matching methods in three broad categories: (1) feature based methods, (2) graph based methods and (3) other methods. Fig. 3 illustrates a more detailed categorization of shape matching methods. Note, that the classes of these methods are not completely disjoined. For instance, a graph-based shape descriptor, in some way, describes also the global feature distribution. By this point of view the taxonomy should be a graph.

3.1. 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.

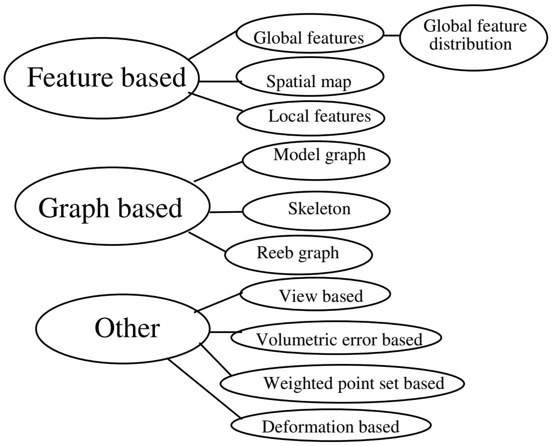

Figure 3. Taxonomy of shape matching methods.

The value of d can easily be a few hundred. 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. Using the Euclidean distance, matching feature descriptors can be done efficiently in practice by searching in multiple 1D spaces to solve the approximate k nearest neighbor problem as shown by Indyk and Motwani [36]. In contrast with the feature based methods from the first three categories, local feature based methods describe for a number of surface points the 3D shape around the point. For this purpose, for each surface point a descriptor is used instead of a single descriptor.

3.1.1. Global feature based similarity

Global features characterize the global shape of a 3D model. Examples of these features are the 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.

Zhang and Chen [88] describe methods to compute global features such as volume, area, statistical moments, and Fourier transform coefficients efficiently.

Paquet et al. [67] apply bounding boxes, cords-based, moments-based and wavelets-based descriptors for 3D shape matching.

Corney et al. [21] introduce convex-hull based indices like hull crumpliness (the ratio of the object surface area and the surface area of its convex hull), hull packing (the percentage of the convex hull volume not occupied by the object), and hull compactness (the ratio of the cubed surface area of the hull and the squared volume of the convex hull).

Kazhdan et al. [42] describe a reflective symmetry descriptor as a 2D function associating a measure of reflective symmetry to every plane (specified by 2 parameters) through the model's centroid. Every function value provides a measure of global shape, where peaks correspond to the planes near reflective symmetry, and valleys correspond to the planes of near anti-symmetry. Their experimental results show that the combination of the reflective symmetry descriptor with existing methods provides better results.

Since only global features are used to characterize the overall shape of the objects, these methods are not very discriminative about object details, but their implementation is straightforward. Therefore, these methods can be used as an active filter, after which more detailed comparisons can be made, or they can be used in combination with other methods to improve results.

Global feature methods are able to support user feedback as illustrated by the following research. Zhang and Chen [89] applied features such as volume-surface ratio, moment invariants and Fourier transform coefficients for 3D shape retrieval. They improve the retrieval performance by an active learning phase in which a human annotator assigns attributes such as airplane, car, body, and so on to a number of sample models. Elad et al. [28] use a moments-based classifier and a weighted Euclidean distance measure. Their method supports iterative and interactive database searching where the user can improve the weights of the distance measure by marking relevant search results.

3.1.2. Global feature distribution based similarity

The concept of global feature based similarity has been refined recently by comparing distributions of global features instead of the global features directly.

Osada et al. [66] introduce and compare shape distributions, which measure properties based on distance, angle, area and volume measurements between random surface points. They evaluate the similarity between the objects using a pseudo-metric that measures distances between distributions. In their experiments the D2 shape distribution measuring distances between random surface points is most effective.

Ohbuchi et al. [64] investigate shape histograms that are discretely parameterized along the principal axes of inertia of the model. The shape descriptor consists of three shape histograms: (1) the moment of inertia about the axis, (2) the average distance from the surface to the axis, and (3) the variance of the distance from the surface to the axis. Their experiments show that the axis-parameterized shape features work only well for shapes having some form of rotational symmetry.

Ip et al. [37] investigate the application of shape distributions in the context of CAD and solid modeling. They refined Osada's D2 shape distribution function by classifying

2 random points as 1) IN distances if the line segment connecting the points lies complete inside the model, 2) OUT distances if the line segment connecting the points lies complete outside the model, 3) MIXED distances if the line segment connecting the points lies passes both inside and outside the model. Their dissimilarity measure is a weighted distance measure comparing D2, IN, OUT and MIXED distributions. Since their method requires that a line segment can be classified as lying inside or outside the model it is required that the model defines a volume properly. Therefore it can be applied to volume models, but not to polygonal soups. Recently, Ip et al. [38] extend this approach with a technique to automatically categorize a large model database, given a categorization on a number of training examples from the database.

Ohbuchi et al. [63], investigate another extension of the D2 shape distribution function, called the Absolute Angle-Distance histogram, parameterized by a parameter denoting the distance between two random points and by a parameter denoting the angle between the surfaces on which two random points are located. The latter parameter is actually computed as an inner product of the surface normal vectors. In their evaluation experiment this shape distribution function outperformed the D2 distribution function at about 1.5 times higher computational costs. Ohbuchi et al. [65] improved this method further by a multi-resolution approach computing a number of alpha-shapes at different scales, and computing for each alpha-shape their Absolute Angle-Distance descriptor. Their experimental results show that this approach outperforms the Angle-Distance descriptor at the cost of high processing time needed to compute the alpha-shapes.

Shape distributions distinguish models in broad categories very well: aircraft, boats, people, animals, etc. However, they perform often poorly when having to discriminate between shapes that have similar gross shape properties but vastly different detailed shape properties.

3.1.3. Spatial map based similarity

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.

Ankerst et al. [5] use shape histograms as a means of analyzing the similarity of 3D molecular surfaces. The histograms are not built from volume elements but from uniformly distributed surface points taken from the molecular surfaces. The shape histograms are defined on concentric shells and sectors around a model's centroid and compare shapes using a quadratic form distance measure to compare the histograms taking into account the distances between the shape histogram bins.

Vranić et al. [85] describe a surface by associating to each ray from the origin, the value equal to the distance to the last point of intersection of the model with the ray and compute spherical harmonics for this spherical extent function. Spherical harmonics form a Fourier basis on a sphere much like the familiar sine and cosine do on a line or a circle. Their method requires pose normalization to provide rotational invariance. Also, Yu et al. [86] propose a descriptor similar to a spherical extent function and a descriptor counting the number of intersections of a ray from the origin with the model. In both cases the dissimilarity between two shapes is computed by the Euclidean distance of the Fourier transforms of the descriptors of the shapes. Their method requires pose normalization to provide rotational invariance.

Kazhdan et al. [43] present a general approach based on spherical harmonics to transform rotation dependent shape descriptors into rotation independent ones. Their method is applicable to a shape descriptor which is defined as either a collection of spherical functions or as a function on a voxel grid. In the latter case a collection of spherical functions is obtained from the function on the voxel grid by restricting the grid to concentric spheres. From the collection of spherical functions they compute a rotation invariant descriptor by (1) decomposing the function into its spherical harmonics, (2) summing the harmonics within each frequency, and computing the L_2-norm for each frequency component. The resulting shape descriptor is a 2D histogram indexed by radius and frequency, which is invariant to rotations about the center of the mass. This approach offers an alternative for pose normalization, because their method obtains rotation invariant shape descriptors. Their experimental results show indeed that in general the performance of the obtained rotation independent shape descriptors is better than the corresponding normalized descriptors. Their experiments include the ray-based spherical harmonic descriptor proposed by Vranić et al. [85]. Finally, note that their approach generalizes the method to compute voxel-based spherical harmonics shape descriptor, described by Funkhouser et al. [30], which is defined as a binary function on the voxel grid, where the value at each voxel is given by the negatively exponentiated Euclidean Distance Transform of the surface of a 3D model.

Novotni and Klein [61] present a method to compute 3D Zernike descriptors from voxelized models as natural extensions of spherical harmonics based descriptors. 3D Zernike descriptors capture object coherence in the radial direction as well as in the direction along a sphere. Both 3D Zernike descriptors and spherical harmonics based descriptors achieve rotation invariance. However, by sampling the space only in radial direction the latter descriptors do

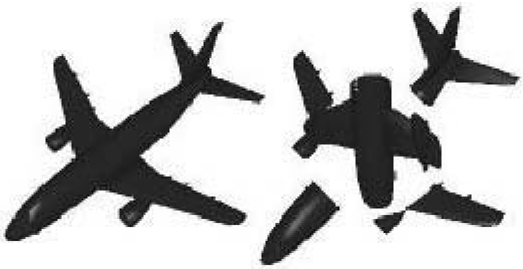

Figure 4. Spherical harmonics do not distinguish models that differ by a rotation to the interior part [43].

not capture object coherence in the radial direction, as illustrated by fig. 4.

The limited experiments comparing spherical harmonics and 3D Zernike moments performed by Novotni and Klein show similar results for a class of planes, but better results for the 3D Zernike descriptor for a class of chairs.

Vranić [84] expects that voxelization is not a good idea, because many fine details are lost in the voxel grid. Therefore, he compares his ray-based spherical harmonic method [85] and a variation of it using functions defined on concentric shells with the voxel-based spherical harmonics shape descriptor proposed by Funkhouser et al. [30]. Also, Vranić et al. [85] accomplish pose normalization using the so-called continuous PCA algorithm. In the paper it is claimed that the continuous PCA is better as the conventional PCA and better as the weighted PCA, which takes into account the differing sizes of the triangles of a mesh. In contrast with Kazhdan's experiments [43] the experiments by Vranić show that for ray-based spherical harmonics using the continuous PCA without voxelization is better than using rotation invariant shape descriptors obtained using voxelization. Perhaps, these results are opposite to Kazhdan results, because of the use of different methods to compute the PCA or the use of different databases or both.

Kriegel et al. [46, 47] investigate similarity for voxelized models. They obtain a spatial map by partitioning a voxel grid into disjoint cells which correspond to the histograms bins. They investigate three different spatial features associated with the grid cells: (1) volume features recording the fraction of voxels from the volume in each cell, (2) solid-angle features measuring the convexity of the volume boundary in each cell, (3) eigenvalue features estimating the eigenvalues obtained by the PCA applied to the voxels of the model in each cell [47], and a fourth method, using instead of grid cells, a more flexible partition of the voxels by cover sequence features, which approximate the model by unions and differences of cuboids, each containing a number of voxels [46]. Their experimental results show that the eigenvalue method and the cover sequence method outperform the volume and solid-angle feature method. Their method requires pose normalization to provide rotational invariance. Instead of representing a cover sequence with a single feature vector, Kriegel et al. [46] represent a cover sequence by a set of feature vectors. This approach allows an efficient comparison of two cover sequences, by comparing the two sets of feature vectors using a minimal matching distance.

The spatial map based approaches show good retrieval results. But a drawback of these methods is that partial matching is not supported, because they do not encode the relation between the features and parts of an object. Further, these methods provide no feedback to the user about why shapes match.

3.1.4. Local feature based similarity

Local feature based methods provide various approaches to take into account the surface shape in the neighbourhood of points on the boundary of the shape.

Shum et al. [74] use a spherical coordinate system to map the surface curvature of 3D objects to the unit sphere. By searching over a spherical rotation space a distance between two curvature distributions is computed and used as a measure for the similarity of two objects. Unfortunately, the method is limited to objects which contain no holes, i.e. have genus zero. Zaharia and Prêteux [87] describe the 3D Shape Spectrum Descriptor, which is defined as the histogram of shape index values, calculated over an entire mesh. The shape index, first introduced by Koenderink [44], is defined as a function of the two principal curvatures on continuous surfaces. They present a method to compute these shape indices for meshes, by fitting a quadric surface through the centroids of the faces of a mesh. Unfortunately, their method requires a non-trivial preprocessing phase for meshes that are not topologically correct or not orientable.

Chua and Jarvis [18] compute point signatures that accumulate surface information along a 3D curve in the neighbourhood of a point. Johnson and Herbert [41] apply spin images that are 2D histograms of the surface locations around a point. They apply spin images to recognize models in a cluttered 3D scene. Due to the complexity of their representation [18, 41] these methods are very difficult to apply to 3D shape matching. Also, it is not clear how to define a dissimilarity function that satisfies the triangle inequality.

Körtgen et al. [45] apply 3D shape contexts for 3D shape retrieval and matching. 3D shape contexts are semi-local descriptions of object shape centered at points on the surface of the object, and are a natural extension of 2D shape contexts introduced by Belongie et al. [9] for recognition in 2D images. The shape context of a point **p**, is defined as a coarse histogram of the relative coordinates of the remaining surface points. The bins of the histogram are defined by the overlay of concentric shells around the centroid of the model and sectors emerging from the centroid. Matching consists of a local matching stage and a global matching stage. In the local matching stage, for points **p** the best matching point **q** is found on the other shape. In the global matching stage, correspondences between similar sample points on the two shapes are found.

Compared to the methods presented in the previous sections of this paper, matching 3D shape contexts is less efficient, efficient indexing is not straightforward, and the obtained dissimilarity measure does not obey the triangle inequality.

3.2. 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. For an extensive discussion of Reeb graphs and skeletons we refer the reader to the paper of Biasotti et al. [14].

Efficient computation of existing graph metrics for general graphs is not possible: computing the edit distance is NP-hard [90] and computing the maximal common subgraph [32] is even NP-complete. Polynomial solutions can be obtained for directed acyclic graphs such as shock graphs. Sebastian et al. [70] describe an approach to compute a pseudo-metric between shock graphs. It is obtained by exhaustively searching for the optimal deformation path between two 2D shapes, and using the cost of this path as a distance between two shapes. But the computation time of this method is too high for practical application, and it is not straightforwardly generalized to 3D.

3.2.1. Model graph based similarity

Model graph based similarity methods are applicable to 3D solid models as produced by CAD most systems. The most dominant solid modeling representation methods are Boundary Representation (B-rep) and Constructive Solid Geometry (CSG). A B-rep describes a model in terms of its vertices, edges and faces. By contrast to the facets in meshes, the faces of a B-rep may be represented as freeform surfaces. CSG describes the part in terms of a set of Boolean operations applied to primitive geometric entities such as cubes and cylinders. For an introduction to solid modeling representations we refer the reader to the book by Hoffmann [35]. For content based retrieval of solid models, researchers have investigated the application of graph-based datastructures storing engineering features (machining features, form features, etc.). Elinson et al. [29], and Cicirello and Regli [19] investigate the application of model dependency graphs storing machining features. These approaches compare the similarity of solid models by comparing their associated manufacturing plans.

McWerther et al. [53, 54, 55], and El-Mehalawi and Miller [26, 27] apply model signature graphs, that both model the topology of a shape model by a graph structure, and map a number of engineering features to a high-dimensional feature vector. Model signature graphs represent the faces from the B-Rep data structure of the solid model as nodes and the boundary curves between the faces as edges. This approach allows comparison of shape models by comparing their topology using graphs and comparing their other properties using feature vectors in the same way as feature based methods. Therefore, McWerther et al. [53, 54, 55] apply approximate graph comparison using the spectrum of the graph. The graph spectrum, which consists of the sorted eigenvalues of the adjacency matrix of the graph, is strongly related to the structure of the graph, and hence to the topology of the shape model. El-Mehalawi and Miller [26, 27] apply approximate graph comparison based on local clique matching.

The model graph based approaches are especially relevant for the CAD/CAM community, but are difficult to apply for models of natural shapes like humans and animals. To the best of our knowledge only Zuckerberger et al. [91], applied an approach similar to model graphs to content based retrieval suitable for natural shapes. They decompose the surface of a model into patches classified as similar to a sphere, a cylinder, a cone or a plane, and identify adjacent patches to build a graph representation of the model.

3.2.2. Skeleton based similarity

Sundar et al. [76] use as a shape descriptor a skeletal graph that encodes geometric and topological information. After voxelization of a shape, the skeletal points are obtained by a distance transform-based thinning algorithm developed by Gagvani [31] using a thinness parameter. The skeletal points are connected in an undirected acyclic shape graph by applying the Minimum Spanning Tree algorithm. Decreasing the thinness results in denser skeletal graphs. So, by using different values of the thinness parameter they obtain a hierarchical graph structure. Each node in the graph represents a segment of the original skeleton. With each node a geometrical signature vector is associated encoding the radial distribution about the segment. Also, with each node of the graph a topological signature vector is associated encoding the topology of the subtrees rooted at the node. This topological signature vector is defined recursively over the subgraphs of the node using eigenvalues of their adjacency matrices. Sundar et al. [76] match two shapes by approximate comparison of their hierarchical skeletal graphs using

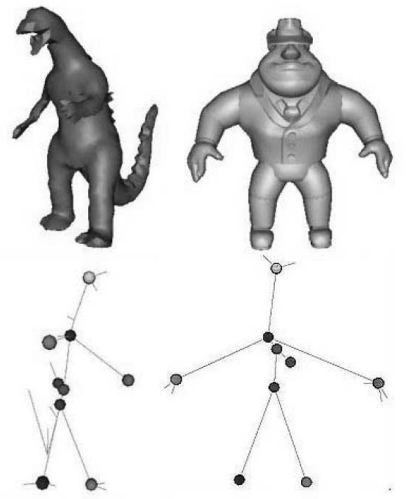

Figure 5. Skeletal graph matching with colors showing the node-to-node correspondence based upon the topology and radial distance about the edge [76].

a greedy algorithm finding the maximum cardinality, minimum weight matching in a bipartite graph. Efficient indexing of a skeletal graph is supported by storing for each node its topological signature vector. Since a topological signature vector has a fixed size, the size of its shape descriptor is a constant multiplied by the number of nodes of the skeleton. Fig. 5 illustrates their shape matching approach. Their method also supports matching of articulated objects, by taking into account only the topological signature vector as a descriptor, and partial matching, where a skeletal graph is allowed to match with a subgraph of another skeletal graph.

Iyer et al. [39, 51] use both global features and skeletal graphs to describe volume models, obtained by voxelizing solid models. They obtain a skeletal graph by a thinning algorithm iteratively eroding voxels until a one-voxel width skeleton is left. They match two shapes by an algorithm detecting graph/subgraph isomorphism using a decision-tree based approach developed by Messmer and Bunke [56]. Their algorithm indexes all the graphs in a database in the form of a decision tree using the various permutations of the adjacency matrix. Hence, this algorithm obtains a search time polynomial in the number of graph nodes, at the cost of exponential space requirements. Their results show the feasibility of their approach for relatively small volume models.

3.2.3. Reeb graph based similarity

Mathematically, the Reeb graph is defined as the quotient space of a shape S and a quotient function f. Biasotti et al. [15] compare Reeb graphs obtained by using different quotient functions f and highlight how the choice of f determines the final matching result. For instance, the integral geodetic distance as quotient function is especially suited for articulated objects, while the distance to the barycenter should be preferred if the aim is to distinguish different poses of an articulated object. Hilaga et al. [34] describe a topological matching method relevant especially for articulated objects. Their method uses Reeb graphs based on a quotient function defined by an integral geodesic distance. Bespalov et al. [12] investigate the application of Hilaga's method to solid models. They found that for solid models, minor changes in topology may result in significant differences in similarity. Since for solid models topological insensitivity is important, they conclude that the Reeb graph technique requires some improvements. Bespalov et al. [13] present preliminary research on a modification of Hilaga's method, which computes a scale-space decomposition of a shape, represented as a rooted undirected tree instead of a Reeb graph. This reduces the problem of comparing two 3D models to computing a matching among the corresponding rooted trees. In summary, Reeb graphs defined by a geodesic distance are suited for matching articulated objects, but they are sensitive to topological changes. Also, they cannot be applied to arbitrary meshes, because topological problems like missing faces disturb the computation of geodesic distances.

3.3. Other methods

Finally, we discuss a number of other methods that are applied for shape matching.

3.3.1. View based similarity

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. Löffler [49] applies view based similarity to retrieve 3D models using a 2D query interface. In the preprocessing phase, for each 3D model a descriptor is obtained consisting of a number of binary images. In the query phase, a sketch or a 2D image is used as a query to retrieve a number of 3D models, whose images match the query. Also, Funkhouser et al. [30] apply view based similarity to implement a 2D sketch query interface. In the preprocessing phase a descriptor of each 3D model is obtained by 13 thumbnail images of boundary contours of the 3D object as seen from 13 orthographic view directions. Then in the query phase the user defines a 3D shape query by drawing one or more sketches. 3D shape models are retrieved by comparing these sketches with the descriptors from the shapes in the database using image matching.

In the approach described by Cyr and Kimia [22], a query is specified by a view of a 3D object. A descriptor of a 3D object consists of a number of views of the 3D object. The number of views of each object is kept small by clustering views, and by representing each cluster with one view, which is represented by a shock graph. They recognize a 3D shape by comparing a view of the shape with all views of 3D objects using shock graph matching. However, they do not address the shock graph indexing problem, resorting to a linear search of all views in the database in order to retrieve an object.

Using shock matching, Macrine et al. [52] apply indexing using topological signature vectors to implement view based similarity matching more efficiently. Also, recently, view based similarity has been applied to retrieve 3D objects by Chen et al. [17]. They consider two models to be similar, if they look similar from all viewing angles. Therefore, a lightfield descriptor is introduced, which 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. Now, 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. The running time of the retrieval process is reduced by a clever multi-step approach supporting early rejection of non-relevant models. The experimental results show that their approach obtains better retrieval results then the 3D harmonics approach proposed by Funkhouser et al. [30] using a test database containing 1,833 models, at the cost of much more processing time.

3.3.2. Volumetric error based similarity

Novotni and Klein [60] describe a geometry similarity approach to 3D shape matching based on calculating a volumetric error between one object and a sequence of offset hulls of the other object. A drawback of their method is that their dissimilarity measure is not symmetric and does not obey the triangle inequality. Sánchez-Cruz and Bribiesca present a method [69] relating the volumetric error between two voxelized shapes to a transportation distance measuring how many voxels have to move and how far to change one shape into another. Since in general these voxelized shapes will have many voxels, the computation of this transportation distance will be high.

3.3.3. Weighted point set based similarity

Another approach is based on shape descriptors consisting of weighted 3D points. Dey et al. [24] present a method to obtain a descriptor of a shape, given by a point sample, by first decomposing the shape into its components. They obtain as shape descriptor a weighted point set by representing each component by a weighted point, where the weight of the point denotes the volume of the component. They match weighted point sets by a measure which does not obey the triangle inequality. Tangelder and Veltkamp [78] use as shape descriptor a weighted point sets consisting of points with a high curvature value. A measure for the curvature is used as a weight. They compare weighted point sets using a variant of the Earth Mover's distance, the proportional transportation distance, which obeys the triangle inequality [33]. Shamir et al. [72] propose a shape descriptor consisting of a hierarchy of weighted point sets, representing spherical shape approximations. They utilize this multiresolution approximation to implement an algorithm to simultaneously align and compare two shapes.

3.3.4. Deformation based similarity

A number of methods [20, 8] compare a pair of 2D shapes by measuring the amount of deformation required to register the shapes exactly. These methods depend on the natural arc length parameterization of their contours, which is not straightforwardly generalized to 3D. As a result, methods that apply deformation for shape recovery [79] or shape evolution [23] are very difficult to apply for 3D shape matching.

4. Overview and conclusions

In this section we summarize our discussion on shape matching methods from the previous section and indicate directions for further research.

Feature based methods, categorized into (1) global features, (2) global feature distributions, (3) spatial maps and (4) local features, characterize shapes by their feature values. The shape matching methods from the first three categories represent the feature values by a vector in a high d-dimensional vector space. Since the feature values are typically computed by sampling 3D shapes, no restrictions on the kind of shape model are imposed and in general the descriptor computation is fast. Because a feature vector is a point in a fixed d-dimensional space, two models can be compared fast by computing their distance in this space. Also, indexing is straightforward and retrieval can be implemented efficiently by nearest neighbour search. In general these methods are robust, because they are based on sampling. For most features, normalization is required e.g. using the PCA, or rotation invariant shape descriptors should be obtained (e.g. using Kazhdan's method [43]). The discriminative abilities of Osada's method [66] have been improved by further refinements of distribution methods as well as by several methods based on spatial maps [43, 61, 84]. If details of shapes are not taken into account, these methods distinguish shapes very well. Details may be taken into account using higher order moments, but this has not been verified by experiments. A drawback of these methods is that partial matching is not supported, be-

	References	Shape model	Triangle inequality	Efficiency	Discriminative power	Partial matching	Robustness	Normalization required
Global feature	[21, 28, 42] [67, 88, 89]	all models	yes	fast	low	no	high	no
Global feature distribution	[37, 38, 63] [64, 65, 66]	all models	yes	fast	medium	no	high	only [64]
Spatial map	[5, 30, 43] [46, 47, 61] [84, 85, 86]	all models	yes	fast	high [43, 61, 85]	no	high	not for [30, 43, 61]
Local feature	[18, 41, 45] [74, 87]	mesh	unknown	medium	medium	yes	medium	no
Model graph	[19, 26, 27, 29] [53, 54, 55, 91]	solid except [91]	not applicable in practice	medium	medium	yes	medium	no
Skeleton	[39, 51, 76]	volume	not applicable in practice	medium	medium	yes	medium	no
Reeb graph	[12, 13, 15, 34]	volume	not applicable in practice	medium	medium	yes	medium	no
View	[17, 22, 30] [49, 52]	mesh	[17]	medium	high [17]	no	high	no
Volumetric error	[60, 69]	volume	[69]	fast [60]	medium	no	high	yes
Weighted point set	[24, 72, 78]	mesh	[78]	medium	medium	no	medium	yes
Deformation	[23, 79]	mesh	unknown	slow	medium	no	high	no

Table 1. Comparison of shape matching methods. References indicate which papers provide the indicated property. If no reference is indicated the property is valid in general.

cause they do not encode the relation between the features and parts of an object. Further, these methods provide no feedback to the user about why shapes match. Compared to the methods from the first three categories, local feature methods, which compute feature value vectors for a number of surface points, matching is less efficient, efficient indexing is not straightforward, and the obtained dissimilarity measure is not obey the triangle inequality. But the locality of the method provides opportunities for partial matching. Therefore, more research on these methods is worthwhile.

Model graphs are extracted from solid model representations [35] used by most CAD systems. Therefore, the model graph based approaches are only applicable to solid models, while skeletal graph and Reeb graph approaches are applicable to volume models including models of natural shapes like humans and animals represented as volumes. The only exception is the model graph based approach by Zuckerberger et al. [91], which is also applicable to natural shapes.

For graph based descriptors the complexity of the exact computation of a metric obeying the triangle inequality prohibits practical application. Hence, the efficient implementation of approximate matching methods is a current research issue. Pure graph based methods have a limited discriminating power, because only topology is taken into account. To improve discriminative abilities most authors apply graph based matching in combination with other methods. For instance, Sundar et al. [76] match two skeletal graphs based upon the topology and radial distance about the edges of the skeletal graphs. For graph based methods, minor changes in topology may result in significant differences in similarity. Hence, these methods are less robust than feature based methods. Advantages of graphs based methods are that no pose normalization is required, and that a graph based structure is suited to implement partial matching.

The view based similarity approach recently implemented by Chen et al. [17] provides good retrieval results at the cost of processing time. It does not require pose normalization, because its similarity comparison is rotation independent. Since their experimental results, obtained with their own database, show better discriminative abilities than other methods, it would be interesting to compare these methods using other databases. The volumetric error approach implemented by Novotni et al. [60] allows fast matching, but uses a dissimilarity measure that does not obey the triangle inequality. In contrast, Sánchez-Cruz and Bribiesca [69] use a pseudo-metric dissimilarity measure which is expensive to compute. Among the weighted point set approaches only Tangelder et al. use a pseudo-metric to compare weighted point sets. Finally, deformation based methods which have been applied to 2D shapes, are too slow for 3D shape matching. Table 1 summarizes the above discussion with respect to several requirements on content based 3D shape retrieval.

We identify the follow research issues:

- **Comparison using benchmarks**: Comparison of different shape matching methods using publicly available benchmark databases, containing models classified in different categories containing similar shapes. Due to the lack of publicly available benchmarks in the past, it was not possible to compare the shape matching results obtained by different researchers. Very recently, the first benchmarks for mesh models are made available at web pages from Princeton university [2, 73], the university of Konstanz [3], and Utrecht University [4, 78]. The Princeton Shape Benchmark database contains 1,834 3D models downloaded from the web, subdivided into a training set and a test set, containing 907 models each, classified into 90 and 92 classes respectively. The database at the university of Konstanz provides a test set containing 473 models classified into 55 classes, and 1366 unclassified models. The database at the university of Utrecht consists of 512 models classified into six categories. To the best of our knowledge no such publicly available benchmark for volume models is available yet.

- **Obtaining rotation invariance**: To obtain rotation invariant dissimilarity measures, Vranić et al. [85] advocate the use of the continuous PCA method for pose normalization, while Kazhdan et al. [43], and Novotni and Klein [61], favour using rotation invariant shape descriptors that need no pose normalization. Further research is needed to compare both approaches using the same benchmarks and the best PCA method.
- **Efficient indexing**: The vantage method [83] can be applied to compute an efficient index structure for pseudo-metrics that require much computing time. Also, clustering of d-dimensional feature vectors [9] can be applied for efficient indexing. For graph descriptors the development of an efficient indexing method is a major research topic. For instance, Sebastian et al. [71] describe a promising approach to indexing shock graphs.
- **Partial matching**: Local feature based methods and graph methods seem to be applicable to partial matching, but no practical results are known at present.
- **Combining shape matching methods**: Since the capabilities of feature based methods (fast computation, pseudo-metric, discriminative abilities, robustness) are orthogonal to the capabilities of graph based methods (partial matching, no normalization required), combining different approaches may produce more powerful shape matching methods. Also combining geometry and topology based approaches may produce better shape matching methods.

Acknowledgements

The figures 2, 4, and 5 are extracted from [30], [43], and [76], respectively. We thank Patrick Min for his valuable comments on a draft of this paper and the reviewers for their useful suggestions.

References

[1] 3D model retrieval system, http://3d.csie.ntu.edu.tw/~dynamic/.

[2] 3D model search engine, http://shape.cs.princeton.edu.

[3] 3D model similarity search engine, http://merkur01.inf.uni-konstanz.de/CCCC/.

[4] 3D shape retrieval engine, http://www.cs.uu.nl/centers/ give/imaging/3Drecog/3Dmatching.html.

[5] M. Ankerst, G. Kastenmuller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *Symposium on Large Spatial Databases*, pages 207–226, 1999.

[6] I. Atmosukarto and P. Naval. A survey of 3D model retrieval systems, www.comp.nus.edu/~indri/research.htm, 2003.

[7] J. Barros, J. French, W. Martin, P. Kelly, and M. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *Proc. of SPIE*, volume 267, pages 392–403, 1996.

[8] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.

[9] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[10] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourrne. The Protein Data Bank. *Nucleic Acids Res.*, 28:235–242, 2000.

[11] P. J. Besl and R. C. Jain. Three-dimensional object recognition. *Computing Surveys*, 17(1):75–145, 1985.

[12] D. Bespalov, A. Shokoufandeh, and W. C. Regli. Reeb graph based shape retrieval for CAD. In *DETC'03*, 2003.

[13] D. Bespalov, A. Shokoufandeh, W. C. Regli, and W. Sun. Scale-space representation of 3D models and topological matching. In *Solid Modeling '03*, pages 208–215, 2003.

[14] S. Biasotti, S. Marini, M. Mortara, and G. Patané. An overview of properties and efficacy of topological skeletons in shape modelling. In *SMI 2003*, pages 245–254, 2003.

[15] S. Biasotti, S. Marini, M. Mortara, G. Patané, M. Spagnuolo, and B. Falcidieno. 3D shape matching through topological structures. In *DGCI 2003*, pages 194–203, 2003.

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

[17] D.-Y. Chen, 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.

[18] S. J. Chua and R. Jarvis. Point signatures: a new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–5, 1997.

[19] V. Cicirello and W. C. Regli. Machining feature-based comparisons of mechanical parts. In *SMI 2001*, pages 176–185, 2001.

[20] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In *ECCV '92*, Lecture Notes in Computer Science 588. Springer, 1992.

[21] J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, and R. Macleod. Coarse filters for shape matching. *IEEE Computer Graphics and Applications*, 22(3):65–74, 2002.

[22] C. M. Cyr and B. Kimia. 3D object recognition using shape similiarity-based aspect graph. In *ICCV01*, pages I:254–261, 2001.

[23] D. DeCarlo and D. Metaxas. Shape evolution with structural and topological changes using blending. *PAMI*, 20(11):1185–1205, 1998.

[24] T. K. Dey, J. Giesen, and S. Goswami. Shape segmentation and matching with flow discretization. In *Workshop on Algorithms and Data Structures*, Lecture Notes in Computer Science 2748, pages 25–36. Springer, 2003.

[25] Digital Michelangelo project, http://graphics.stanford.edu/data/mich/.

[26] M. El-Mehalawi and R. A. Miller. A database system of mechanical components based on geometric and topological similarity. part I: representation. *Journal of Computer-Aided Design*, 35(1):83–94, January 2003.

[27] M. El-Mehalawi and R. A. Miller. A database system of mechanical components based on geometric and topological similarity. part II: indexing, retrieval, matching and similarity assessment. *Journal of Computer-Aided Design*, 35(1):95–105, January 2003.

[28] M. Elad, A. Tal, and S. Ar. Content based retrieval of VRML objects - an iterative and interactive approach. In *Eurographics Multimedia Workshop*, pages 97–108, 2001.

[29] A. Elinson, D. S. Nau, and W. C. Regli. Feature-based similarity assessment of solid models. In *Solid Modeling '97*, pages 297–310, 1997.

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

[31] N. Gagvani and D. Silver. Parameter controlled volume thinning. *Graphical Models and Image Processing*, 61(3):149–164, 1999.

[32] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[33] P. Giannopoulos and R. C. Veltkamp. A pseudo-metric for weighted point sets. In *ECCV 2002*, pages 715–730, 2002.

[34] M. Hilaga, Y. Shinagawa, and T. Kohmura. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH 2001*, pages 203–212, 2001.

[35] C. M. Hoffmann. *Geometric & Solid Modeling, An Introduction*. Morgan Kaufman, 1989.

[36] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *30th Symposium on Theory of Computing*, pages 604–613, 1998.

[37] C. Y. Ip, D. Lapadat, L. Sieger, and W. C. Regli. Using shape distributions to compare solid models. In *Solid Modeling '02*, pages 273–280, 2002.

[38] C. Y. Ip, L. Sieger, W. C. Regli, and A. Shokoufandeh. Automated learning of model classifications. In *Solid Modeling '03*, pages 322–327, 2003.

[39] N. Iyer, Y. Kalyanaraman, K. Lou, S. Janyanti, and K. Ramani. A reconfigurable 3D engineering shape search system part I: shape representation. In *DETC'03*, 2003.

[40] N. Iyer, K. Lou, S. Janyanti, Y. Kalyanaraman, and K. Ramani. Three dimensional shape searching: State-of-the-art review and future trends. 2004. widget.ecn.purdue.edu/˜iyern/publications/Iyer_CAD_review.pdf, submitted for publication.

[41] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *PAMI*, 21(5):635–651, May 1999.

[42] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz. A reflective symmetry descriptor for 3D models. *Algorithmica*, 38(1), 2004.

[43] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, 2003.

[44] J. Koendering. *Solid shape*. The MIT Press, 1990.

[45] M. Körtgen, G.-J. Park, M. Novotni, and R. Klein. 3D shape matching with 3D shape contexts. In *The 7th Central European Seminar on Computer Graphics*, April 2003.

[46] H.-P. Kriegel, S. Brecheisen, P. Kröger, M. Pfeile, and M. Schubert. Using sets of feature vectors for similarity search on voxelized CAD objects. In *SIGMOD 2003*, 2003.

[47] H.-P. Kriegel, P. Kröger, Z. Mashael, M. Pfeile, M. Pötke, and T. Seidl. Effective similarity search on voxelized CAD objects. In *DASFAA'03*, 2003.

[48] M. Levoy, 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 *Siggraph 2000*, pages 131–144, 2000.

[49] J. Löffler. Content-based retrieval of 3D models in distributed web databases by visual shape information. In *IV2000*, 2000.

[50] S. Loncaric. A survey of shape analyis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[51] K. Lou, S. Janyanti, N. Iyer, Y. Kalyanaraman, S. Prabhakar, and K. Ramani. A reconfigurable 3D engineering shape search system part II: database indexing, retrieval and clusturing. In *DETC'03*, 2003.

[52] D. Macrini, A. Shokoufandeh, S. Dickenson, K. Siddiqi, and S. Zucker. View-based 3-D object recognition using shock graphs. In *ICPR 2002*, 2002.

[53] D. McWherter, M. Peabody, W. C. Regli, and A. Shokoufandeh. An approach to indexing databases of graphs. Technical report, Drexel University, Sep 2001.

[54] D. McWherter, M. Peabody, W. C. Regli, and A. Shokoufandeh. Transformation invariant shape similarity comparison of solid models. In *DETC'01*, 2001.

[55] D. McWherter, M. Peabody, A. Shokoufandeh, and W. C. Regli. Database techniques for archival of solid models. In *Solid Modeling '01*, pages 78–87, 2001.

[56] B. T. Messmer and H. Bunke. A decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition*, 32:1979–1998, 1999.

[57] P. Min, J. A. Halderman, M. Kazhdan, and T. A. Funkhouser. Early experiences with a 3D model search engine. In *Web3D Symposium*, pages 7–18, March 2003.

[58] P. Min, M. Kazhdan, and T. Funkhouser. A comparison of text and shape matching for retrieval of online 3D models. 2004. *submitted for publication*.

[59] National design repository,
http://edge.mcs.drexel.edu/repository/frameset.html.

[60] M. Novotni and R. Klein. A geometric approach to 3D object comparison. In *SMI 2001*, pages 154–166, 2001.

[61] M. Novotni and R. Klein. 3D Zernike descriptors for content based shape retrieval. In *Solid Modeling 2003*, 2003.

[62] Ogden IV system, http://www.nime.ac.jp/˜motofumi/Ogden/.

[63] R. Ohbuchi, T. Minamitani, and T. Takei. Shape-similarity search of 3D models by using enhanced shape functions. In *Theory and Practice of Computer Graphics 2003, p. 97-104*, 2003.

[64] R. Ohbuchi, T. Otagiri, M. Ibato, and T. Takei. Shape-similarity search of three-dimensional models using parameterized statistics. In *Pacific Graphics 2002*, pages 265–274, 2002.

[65] R. Ohbuchi and T. Takei. Shape-similarity comparison of 3D models using alpha shapes. In *Pacific Graphics 2003*, 2003.

[66] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002.

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

[68] W. C. Regli and V. A. Cicirello. Managing digital libraries for computer-aided design. *Journal of Computer-Aided Design*, 32(2):119–132, Feb 2000.

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

[70] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing shock graphs. In *ICCV*, pages 755–762, 2001.

[71] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Shock-based indexing into large shape databases. In *ECCV (3)*, pages 731–746, 2002.

[72] A. Shamir, A. Sotzio, and D. Cohen-Or. Enhanced hierarchical shape matching for shape transformation. *International Journal for Shape Modeling IJSM*, 9(2), 2004.

[73] P. Shilane, M. Kazhdan, P. Min, and T. Funkhouser. The Princeton Shape Benchmark. In *SMI 2004*, 2004.

[74] H.-Y. Shum, M. Hebert, and K. Ikeuchi. On 3D shape similarity. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 526–531, 1996.

[75] Stanford digital Formae Urbis Romae project,
http://formaurbis.stanford.edu/index.html.

[76] H. Sundar, D. Silver, N. Gagvani, and S. Dickenson. Skeleton based shape matching and retrieval. In *SMI 2003*, pages 130–139, 2003.

[77] M. T. Suziki. A search engine for polygonal models to support development of 3D e-learning applications. In *The Tenth International World Wide Web Conference Poster Proceedings*, pages 182–183, 2001.

[78] J. W. H. Tangelder and R. C. Veltkamp. Polyhedral model retrieval using weighted point sets. *International Journal of Image and Graphics*, 3(1):209–229, 2003.

[79] D. Terzopoulus and D. Metaxas. Dynamic models with local and global deformations: Deformable superquadrics. *PAMI*, 13(7):703–714, 1991.

[80] The protein data bank, http://www.rcsb.org/pdb/.

[81] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[82] R. C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. In M. Lew, editor, *Principles of Visual Information Retrieval*, pages 87–119. Springer Verlag, 2001.

[83] J. Vleugels and R. C. Veltkamp. Efficient image retrieval through vantage objects. *Pattern Recognition*, 35(1):69–80, 2002.

[84] D. V. Vranić. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *ICIP 2003*, 2003.

[85] D. V. Vranić, D. Saupe, and J. Richter. Tools for 3D-object retrieval: Karhunen-Loeve-transform and spherical harmonics. In *Proc. IEEE 2001 Workshop on Multimedia Signal Processing*, 2001.

[86] M. Yu, I. Atmosukarto, W. K. Leow, Z. Huang, and R. Xu. 3D model retrieval with morphing-based geometric and topological feature maps. In *CVPR03*, pages II: 656–661, 2003.

[87] T. Zaharia and F. Prêteux. 3D shape-based retrieval within the MPEG-7 framework. In *Proceedings SPIE Conference 4304*, pages 133–145, January 2001.

[88] C. Zhang and T. Chen. Efficient feature extraction for 2D/3D objects in mesh representation. In *CIP 2001*, 2001.

[89] C. Zhang and T. Chen. Indexing and retrieval of 3D models aided by active learning. In *ACM Multimedia*, pages 615–616, 2001.

[90] K. Zhang, J. T. L. Whang, , and D. Shasha. On the editing distance between undericted acyclic graphs. *Int'l J. Foundations of Computer Science*, 7(1):43–57, 1996.

[91] E. Zuckerberger, A. Tal, and S. Shlafman. Polyhedral surface decomposition wiht applications. *Computers and Graphics*, 26(5):733–743, 2002.

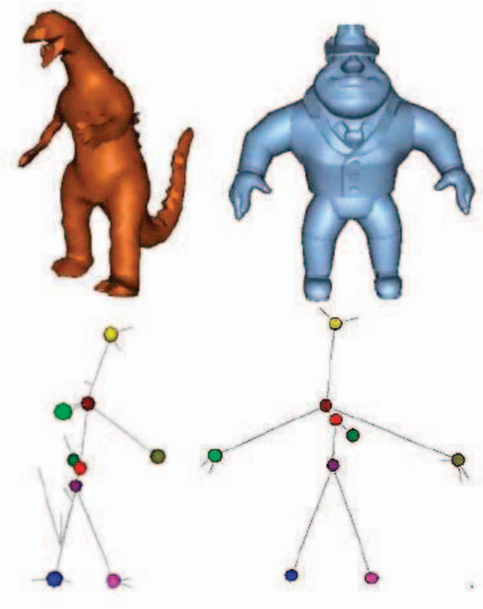

Figure 5. Skeletal graph matching with colors showing the node-to-node correspondence based upon the topology and radial distance about the edge [76].

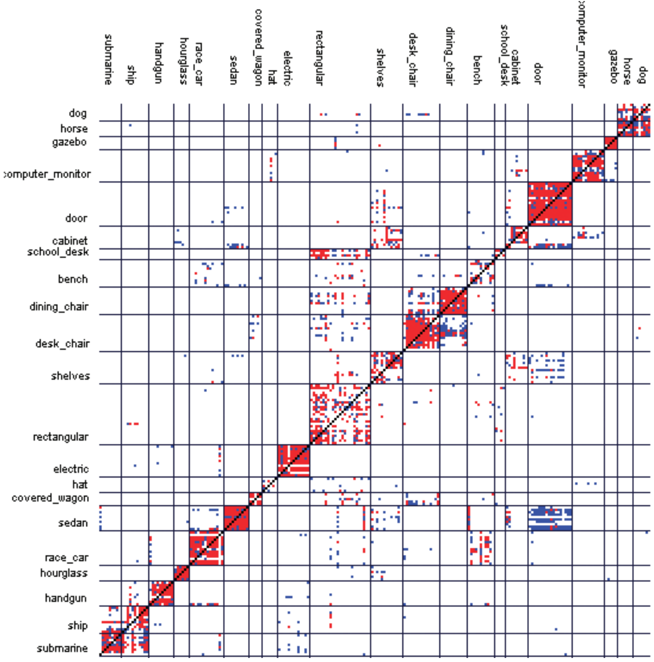

Figure 1. Tier image visualizing nearest neighbor (black), first tier (red), and second tier (blue) computed by matching every model (rows) with every other model (columns) in the base classification of the test set using the LFD algorithm – separating lines and labels indicate classes. Note that the full image is 907x907 "pixels," and only a small portion is shown.

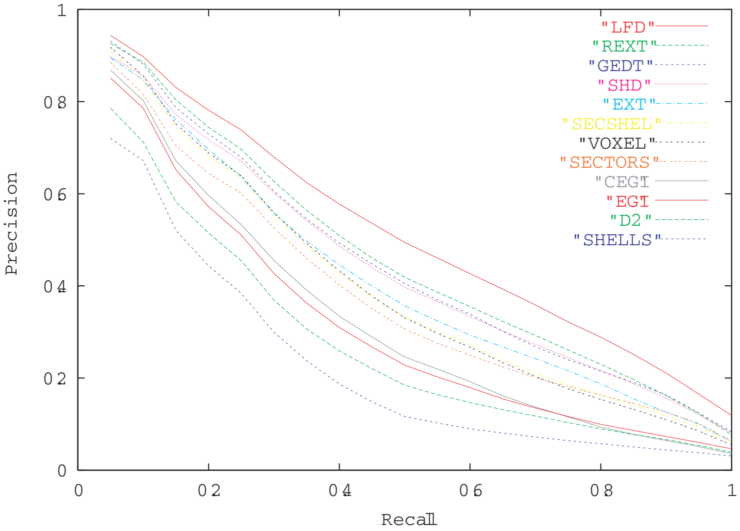

Figure 2. Precision-recall curves computed for 12 shape descriptors for tests with the PSB base classification.