

Chapter 10

3D Model Retrieval Using Medial Surfaces

**Kaleem Siddiqi, Juan Zhang, Diego Macrini, Sven Dickinson,
and Ali Shokoufandeh**

Abstract Graphs derived from medial representations have been used for 2D object matching and retrieval with considerable success (Pelillo et al., 1999; Siddiqi et al., 1999b; Sebastian et al., 2001). In this chapter we consider the use of graphs derived from medial surfaces for 3D object matching and retrieval. The medial representation allows for a qualitative abstraction based on a directed acyclic graph of components and also a degree of invariance to a variety of transformations including the articulation of parts. The formulation discussed in this chapter uses the geometric information associated with each node along with an eigenvalue labeling of the adjacency matrix of the subgraph rooted at that node. Comparative retrieval results are presented against the techniques of shape distributions (Osada et al., 2002) and harmonic spheres (Kazhdan et al., 2003b) on 425 models representing 19 object classes. These results demonstrate that medial surface based graph matching outperforms these techniques for objects with articulating parts.

K. Siddiqi

School of Computer Science & Centre for Intelligent Machines, McGill University, Canada,
e-mail: siddiqi@cim.mcgill.ca

J. Zhang

School of Computer Science & Centre for Intelligent Machines, McGill University, Canada,
e-mail: juan@cim.mcgill.ca

D. Macrini

Department of Computer Science, University of Toronto, Canada,
e-mail: dmac@cs.toronto.edu

S. Dickinson

Department of Computer Science, University of Toronto, Canada,
e-mail: sven@cs.toronto.edu

A. Shokoufandeh

Department of Computer Science, Drexel University, USA,
e-mail: ashokouf@cs.drexel.edu

10.1 Introduction

The problem of object recognition is one of significant interest to the computer vision community. It relates to the process of searching a database of models so as to efficiently retrieve instances that are similar to a particular exemplar. The general problem is difficult because objects can undergo significant deformation and articulation while retaining their identity. The challenge is to come up with representations that provide a degree of invariance under such transformations and which allow for matching algorithms to be applied. The medial models discussed in this book provide a particularly attractive choice because they allow for a reduction of the topology of an object to a graph indicating the relationship between its parts and sub-parts, where each node carries detailed geometric information. As a consequence, medial graphs have been used for 2D object matching and retrieval with demonstrated success in handling part articulation and deformation.

Broadly speaking, there are three classes of existing techniques: (1) graph edit distance based approaches (Sebastian et al., 2001, 2004), (2) sub-graph isomorphism approaches (Pelillo et al., 1999) and (3) graph-spectral approaches (Siddiqi et al., 1999b).¹ These three approaches share the common view that a representation of the 2D medial axis as a graph of components can handle differences in part structure as well as differences in part shapes. The part structure is reflected by the medial axis branching structure and hence the connectivity of the graph, while part shape is reflected in the geometric information associated with each branch. We begin with a brief overview of these classes of methods. Details of each method are presented in the associated references.

10.1.1 Graph Edit Distance Approaches

Graph edit distance approaches assume that similar objects have similar (but not necessarily identical) part structures and part shapes. The essential idea is to use a prescribed set of edit operations to transform one graph in to another. Each of these operations is assigned a cost, and the distance between two objects is determined by the lowest cost set of edit operations between their underlying graph representations. Whereas the notion of edit distance faces a serious issue of computational complexity for arbitrary graphs, polynomial time algorithms have been developed for the case of shock graphs (see Chapter 2) (Klein et al., 2001), which are attributed tree structures. An example of this approach is illustrated in Fig. 10.1, which is adapted from (Sebastian et al., 2004). In this particular example the edit operations take one graph to another by assigning costs to allowed medial axis transitions, as described in Chapter 2. For these costs to be useful in practice, they must take into account

¹ A fourth category for matching medial representations has been developed in detail in Chapter 9, but it assumes that candidate objects can be described by a fixed m-rep topology. Thus, this type of method is less applicable to the problem of 2D or 3D object retrieval, where some variation in part structure for objects within the same category is expected to occur.

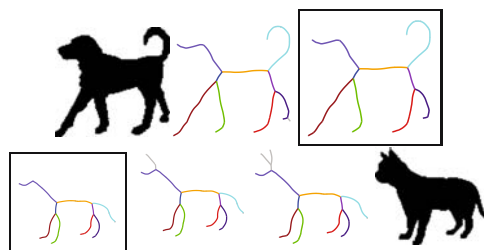


Fig. 10.1 (Adapted from (Sebastian et al., 2004).) Examples of the optimal deformation path between two shapes represented at the extremes of a sequence. The sequence shows operations (symmetry transforms) applied to the medial axis, and the resulting intermediate shock graphs. The boxed shock graphs, which have the same topology, are where the deformation of the two shapes meet in a common simpler shape

both medial axis branching structure (graph connectivity) and medial axis geometry (node attributes).

10.1.2 Subgraph Isomorphism Approaches

Subgraph isomorphism approaches seek to find maximal common subgraphs between two candidate graphs. In the context of matching 2D medial graphs, these approaches have been developed in (Pelillo et al., 1999), using the version of the shock graph developed in (Siddiqi et al., 1999b). The essential idea is to convert the maximal common sub-tree problem into a maximum clique problem on an association graph, and to solve the latter combinatorial problem by converting it to a related optimization problem. The optimization problem is solved by using discrete or continuous time replicator equations. These are differential equations developed in mathematical biology, which have the advantage that they are straightforward to simulate numerically. In these approaches part geometry can be considered in the form of attributes on nodes, leading eventually to a generalization of the maximum clique problem to a maximum weighted clique problem. However, the similarity between part structures, which is reflected in the connectivity of the association graph, is essentially separated from the similarity between part geometries, which is reflected in the weights on association graph nodes.

10.1.3 Graph Spectral Approaches

In graph spectral approaches the essential idea is to create a low-dimensional vector that reflects the topology of the graph. In the context of matching 2D medial directed acyclic graphs (DAGs), one such measure proposed in (Siddiqi et al., 1999b) is based on efficient techniques for computing the sum of the eigenvalues

of the adjacency matrix of the DAG. This approach allows geometric similarity between nodes, which may be interpreted as the similarity between part shapes, to be combined with a topological signature vector that captures an object's overall part structure. This combination is then used for both matching and indexing (Shokoufandeh et al., 1999, 2005).

The subject of this chapter is the application of medial graph matching to 3D object recognition. With regard to the choice of method, the sub-graph isomorphism approach carries the disadvantage that for graphs with a large number of nodes and possibly complicated topology, computational efficiency becomes an issue. The graph edit distance approach is an attractive one to pursue, given an appropriate measure of edit distance in 3D. To our knowledge creating such a measure, a challenging task, has not yet been reported. Therefore, in this chapter we focus on an extension of the third approach based on graph spectra.

10.2 3D Model Retrieval

With an explosive growth in the number of 3D object models stored in web repositories and other databases, the computer vision and computer graphics communities have begun to address the important and challenging problem of 3D object retrieval and matching. Although this problem traditionally falls in the domain of computer vision research, it is also of interest to those interested in applications in the areas of solid modeling and computer-aided design. Recent advances include query-based search engines (Funkhouser et al., 2003) which employ promising measures including spherical harmonic descriptors and shape distributions (Osada et al., 2002). Such systems can yield results on databases including hundreds of 3D models, in a matter of a few seconds.

Thus far the emphasis has broadly been on the use of qualitative measures of shape that are typically global. Such measures are robust in the sense that they can deal with noisy and imperfect models, and at the same time they are simple enough so that efficient algorithmic implementations can be sought. However, an inevitable cost is that such measures are inherently coarse and are sensitive to deformations of objects or their parts. As a motivating example, consider the 3D models in Fig. 10.2. These four exemplars of an object class were created by articulations of parts and changes of pose. For such examples, the very notion of a center of mass or a rigid reference point (Alt et al., 1994), which is crucial for the computation of descriptions such as shape histograms (sectors or shells) (Ankerst et al., 1999) or spherical extent functions (Vranic and Saupe, 2001), can be nonintuitive and arbitrary. In fact, the centroid of such models may actually lie in the background. To complicate matters, it is unclear how to obtain a global alignment of such models, and hence signatures based on a Euclidean distance transform (Borgefors, 1984; Funkhouser et al., 2003) have limited power in this setting. As well, measures based on reflective symmetries (Kazhdan et al., 2003a), and signatures based on 3D moments (Elad et al., 2001) or chord histograms (Osada et al., 2002) are not invariant under such transformations.

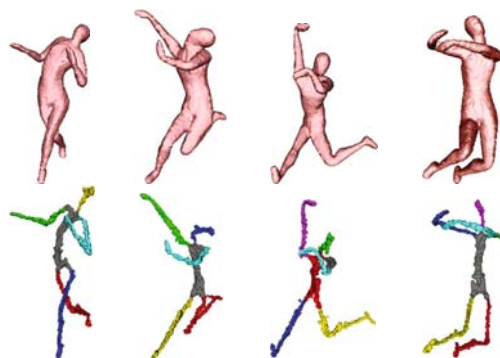


Fig. 10.2 Exemplars of the object class “human” created by changes in pose and articulations of parts (top row). The medial surface (or 3D skeleton) of each is computed using the algorithm of (Siddiqi et al., 2002) (bottom row). The medial surface is automatically partitioned into distinct parts, each shown in a different color

The computer vision community has grappled with the problem of *generic* or category-level object recognition by suggesting representations based on volumetric parts, including generalized cylinders, superquadrics and *geons* (Binford, 1971; Marr and Nishihara, 1978; Pentland, 1986; Biederman, 1987). Such approaches build a degree of robustness to deformations and movement of parts, but their representational power is limited by the vocabulary of geometric primitives that are selected. Motivated in part by such considerations, there have been attempts to encode 3D shape information using probabilistic descriptors. These allow intrinsic geometric information to be captured by low dimensional signatures. An elegant example of this is the geodesic shape distribution of (Hamza and Krim, 2003), where information theoretic measures are used to compare probability distributions representing 3D object surfaces. In the domain of graph theory there have also been attempts to address the problem of 3D shape matching using representations based on Reeb graphs (Shinagawa et al., 1991; Hilaga et al., 2001). These allow for topological properties to be captured, at least in a coarse sense.

An alternative approach is to use 3D medial loci. As pointed out by Blum, this offers the advantage that a graph of parts can be inferred from the underlying local mirror symmetries of the object (Blum, 1973). A formal abstraction of this type based on the generic singularities of the grassfire flow in 3D has already been discussed in Chapter 2 (see also Leymarie and Kimia, 2003). To further motivate this idea, consider once again the human forms of Fig. 10.2. A medial surface-based representation (bottom row) provides a natural decomposition, which is largely invariant to the articulation and bending of parts.

In this chapter, we build on the technique to compute medial surfaces covered in Chapter 4 (see also Siddiqi et al., 2002) by proposing an interpretation of its output as a directed acyclic graph (DAG) of parts. We then use refinements of algorithms based on graph spectra (Shokoufandeh et al., 2005) to tackle the problems of indexing and matching 3D object models. Graph matching algorithms have already shown

promise in the computer vision community for category-level view-based object indexing and matching using 2D skeletal graphs (Siddiqi et al., 1999b; Shokoufandeh et al., 1999; Pelillo et al., 1999; Sebastian et al., 2001). They have also been demonstrated in the context of matching 3D object models with tubular parts, using a centerline approximation of the 3D skeleton (Sundar et al., 2003). We demonstrate their significant potential for medial surface-based 3D object retrieval with experimental results on a database of 320 models representing 13 object classes, including exemplars of both rigid objects and ones with significant articulation of parts. Comparative results using the information retrieval notion of *precision versus recall* demonstrate that this method significantly outperforms the techniques of shape distributions (Osada et al., 2002) and harmonic spheres (Kazhdan et al., 2003b) for objects with articulating parts.

10.3 Medial Surfaces and DAGs

A number of algorithms for computing 3D medial loci and related representations have been covered in this book. These include the average outward flux-based and object angle extension skeletons of Chapter 4 and the related algorithms based on continuous properties of the Euclidean distance function; the methods based on digital distance transforms of Chapter 5; methods based on Voronoi diagrams (Chapters 6 and 7); methods for computing m-reps (Chapter 8) and methods which combine constructs from computational geometry with wavefront propagation such as the shock-scaffold technique (Leymarie and Kimia, 2003). See also the applications discussed in Chapter 11. For several of these algorithms the segmentation of the 3D skeleton into its constituent medial manifolds remains a challenge. In this chapter we choose to employ the method of Chapter 4 since it has the advantage that the digital classification of Malandain et al. (1993) allows for the taxonomy of generic 3D skeletal points (Giblin and Kimia, 2004) to be interpreted on a rectangular lattice, leading to a graph of parts.

Under the assumption that the initial model is given in triangulated form, we begin by scaling all the vertices so that they fall within a rectangular lattice of fixed dimension and resolution. We then sub-divide each triangle to generate a dense intersection with this lattice, resulting in a binary (voxelized) 3D model. The average outward flux of the Euclidean distance function's gradient vector field is computed through unit spheres centered at each rectangular lattice point, using the algorithm of Chapter 4 (Section 2.3). As explained in that chapter, this quantity has the property that it approaches a negative number at skeletal points and goes to zero elsewhere (Siddiqi et al., 2002), and thus it can be used to drive a digital thinning process. Furthermore, the limiting average outward flux values for the case of shrinking discs reveals the object angle, and thus this quantity may be viewed as a type of *flux invariant* for both obtaining the medial locus and for determining the geometry of the bounding surface implied by it (Dimitrov et al., 2003; Dimitrov, 2003). The thinning process has to be implemented with some care, as described in Chapter 4, so

that the topology of the object is not changed. As mentioned above, this process uses the digital classification of points due to Malandain et al. (1993) to label points on the digital medial locus according to the A_k^m classification given in Chapters 1 and 2, i.e., as surface points, rim points, junction points or curve points. See also Table 4.3 of Chapter 4. We refer to this as a *medial surface* representation. This suggests the following 3-step approach for segmenting the (voxelized) medial surface into a set of connected parts:

1. Identify all manifolds comprised of 26-connected surface points and border points.
2. Use junction points to separate these manifolds, but allow junction points to belong to all manifolds that they connect.
3. Form connected components with the remaining curve points, and consider these as parts as well.

This process of automatic skeletonization and segmentation is illustrated for two object classes, a chair and a human form, in Fig. 10.3.

We now propose an interpretation of the segmented medial surface as a directed acyclic graph (DAG). We begin by introducing a notion of *saliency* which captures the relative importance of each component. Consider that the envelope of maximal inscribed spheres of appropriate radii placed at all skeletal points reconstructs the original object's volume (Blum, 1973). The contribution of each component to the overall volume can thus be used as a measure of its significance. Since the

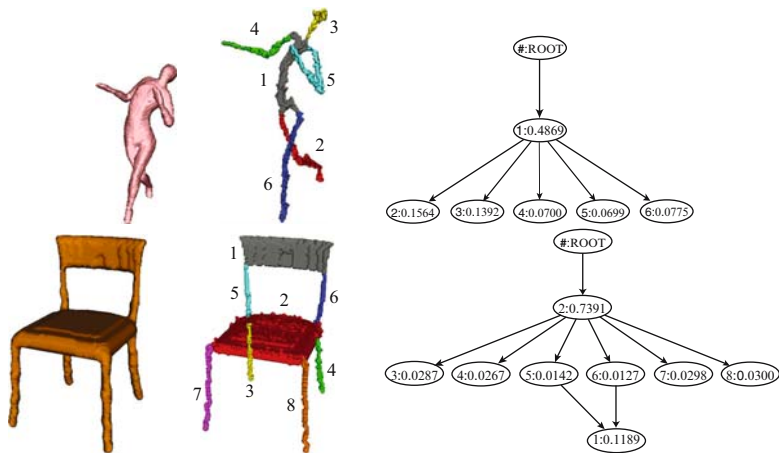


Fig. 10.3 A voxelized human form and chair (left) and their segmented medial surfaces (middle). A hierarchical interpretation of the medial surface, using a notion of part saliency, leads to a directed acyclic graph DAG (right). The nodes in the DAGs have labels corresponding to those on the medial surface, and the saliency of each node is also shown

spheres associated with adjacent components can overlap, an objective measure of component j 's saliency is given by

$$Saliency_j = \frac{Voxels_j}{\sum_{i=1}^N Voxels_i},$$

where N is the number of components and $Voxels_i$ is the number of voxels *uniquely* reconstructed by component i .

The above notion is a reasonable choice for a saliency measure in the context of 3D model retrieval, but is certainly not the only one. In fact, more principled saliency measures could be developed by using the metric measure introduced by Damon in Chapter 3, or by computing appropriate boundary and regional integrals via their analogous medial integral versions, as discussed in that chapter. At present the development of such saliency measures, as well as computational approaches to approximate them, remains the subject of future work.

We now propose the following construction of a DAG, using each component's saliency. Consider the most salient component as the root node (level 0), and place components to which it is connected as nodes at level 1. Components to which these nodes are connected are placed at level 2, and this process is repeated in a recursive fashion until all nodes are accounted for. The graph is completed by drawing edges between all pairs of connected nodes, in the direction of increasing levels, hence avoiding the occurrence of any cycles. However, to allow for 3D models comprised of disconnected parts we introduce a single dummy node as the parent of all DAGs for a 3D model.

This process is illustrated in Fig. 10.3 (right column) for the human and chair models, with the saliency values shown within the nodes. Note how this representation captures the intuitive sense that the human is a torso with attached limbs and a head, a chair is a seat with attached legs and a back, etc. This DAG representation of the medial surface is quite different than the graph structure that follows from a direct use of the taxonomy of 3D skeletal points in the continuum presented in Chapter 2 (Giblin and Kimia, 2004). Our motivation is to be able to exploit the hierarchical structure-indexing and structure-matching algorithms reported in Siddiqi et al. (1999b); Shokoufandeh et al. (2005). However, this conversion can also lead to some limitations; we shall return to a discussion of these at the end of this chapter.

10.4 Indexing

A linear search of the 3D model database, i.e., comparing the query 3D object model to each 3D model and selecting the closest one, is inefficient for large databases. An indexing mechanism is therefore essential to select a small set of candidate models to which the matching procedure is applied. When working with hierarchical structures in the form of DAGs, indexing is a challenging task that can be formulated as the fast selection of a small set of candidate model graphs that share a subgraph with the query. But how do we test a given candidate without resorting to subgraph

isomorphism and its intractability? The problem is further compounded by the fact that due to perturbation and noise, no significant isomorphisms may exist between the query and the (correct) model. Yet, at some level of abstraction, the two structures (or two of their substructures) may be quite similar. Thus, our indexing problem can be reformulated as finding model (sub)graphs whose structure is *similar* to the query (sub)graph.

Choosing the appropriate level of abstraction with which to characterize a DAG is a challenging problem. We seek a description that, on the one hand, provides the low dimensionality essential for efficient indexing, while on the other hand, is rich enough to prune the database down to a tractable number of candidates. In recent work (Shokoufandeh et al., 2005) we draw on the eigenspace of a graph to characterize the topology of a DAG with a low-dimensional vector that will facilitate an efficient nearest-neighbor search in a database.

The eigenvalues of a graph’s adjacency matrix encode important structural properties of the graph, characterizing the degree distribution of its nodes. Moreover, we have shown that the magnitudes of the eigenvalues are stable with respect to minor perturbations of graph structure due to, for example, noise, segmentation error, or minor within-class structural variation (Shokoufandeh et al., 2005).

We can now proceed to define an index based on the eigenvalues. One simple structural abstraction would be a vector of the sorted magnitudes of the eigenvalues of a DAG’s adjacency matrix.² However, for large DAGs, the dimensionality of the index would be prohibitively large (for efficient nearest-neighbor search), and the descriptor would be global, prohibiting effective indexing of query graphs with added or missing parts. This problem can be addressed by exploiting eigenvalue sums rather than the eigenvalues themselves, and by computing both global and local structural abstractions (Siddiqi et al., 1999b). Let V be the root of a DAG whose maximum branching factor is Δ , as shown in Fig. 10.4. Consider the sub-

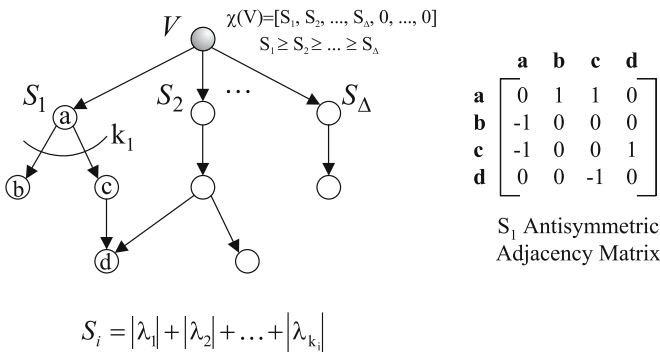


Fig. 10.4 Forming a Low-Dimensional Vector Description of Graph Structure. At node a , we compute the sum of the magnitudes of the k_1 largest eigenvalues of the adjacency sub-matrix defined by the subgraph rooted at a . The sorted sums S_i become the components of $\chi(V)$, the *topological signature vector* (or TSV) assigned to V

² Since the eigenvalues of an antisymmetric matrix are complex we utilize the magnitude of an eigenvalue.

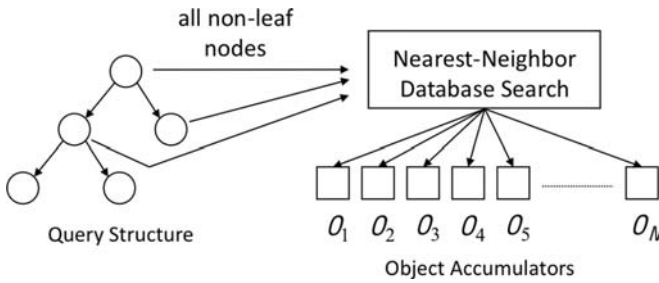


Fig. 10.5 Indexing Mechanism. Each non-trivial node (whose TSV encodes a topological abstraction of the subgraph rooted at the node) votes for models sharing a structurally similar subgraph. Each object accumulator O_i is a bin that stores the number of votes received for object model i . Object models receiving strong support are candidates for a more comprehensive matching process. Adapted from (Shokoufandeh et al., 2005)

graph rooted at node a , the first child of V , and let the out-degree of a be k_1 . We compute the sum S_1 of the magnitudes of the k_1 largest eigenvalues of the adjacency sub-matrix defined by the subgraph rooted at node a , with the process repeated for the remaining children of V . The sorted S_i 's become the components of a Δ -dimensional vector $\chi(V)$, called a *topological signature vector* (TSV), assigned to V . If the number of S_i 's is less than Δ , the vector is padded with zeroes. We can recursively repeat this procedure, assigning a vector to each nonterminal node in the DAG, computed over the subgraph rooted at that node.

Indexing now amounts to a nearest-neighbor search in a model database, as shown in Fig. 10.5. The TSV of each non-leaf node (the root of a graph “part”) in each model DAG defines a vector location in a low-dimensional Euclidean space (the model database) at which a pointer to the model containing the subgraph rooted at the node is stored. At indexing time, a TSV is computed for each non-leaf node, and a nearest-neighbor search is performed using each “query” TSV. Each query TSV “votes” for nearby “model” TSVs and these votes are stored in object accumulator bins, with a distinct bin for each object. In this fashion evidence for models that share the substructure defined by the query TSV is accumulated. Indexing could, in fact, be accomplished by indexing solely with the root of the entire query graph. However, in an effort to accommodate large-scale perturbation (which corrupts all ancestor TSVs of a perturbed subgraph), indexing is performed locally (using all non-trivial subgraphs, or “parts”) and evidence combined. The result is a small set of ranked model candidates which are verified more extensively using the matching procedure described next.

10.5 Matching

Each of the top-ranking candidates emerging from the indexing process must be verified to determine which is most similar to the query. If there were no noise, our problem could be formulated as a graph isomorphism problem for vertex-labeled

graphs. With limited noise, we would search for the largest isomorphic subgraph between query and model. Unfortunately, with the presence of significant noise, in the form of the addition and/or deletion of graph structure, large isomorphic subgraphs may simply not exist. This problem can be overcome by using the same eigen-characterization of graph structure we use as the basis of our indexing mechanism (Siddiqi et al., 1999b).

As we know, each node in a graph (query or model) is assigned a TSV, which reflects the underlying structure in the subgraph rooted at that node. If we simply discarded all the edges in our two graphs, we would be faced with the problem of finding the best correspondence between the nodes in the query and the nodes in the model; two nodes could be said to be in close correspondence if the distance between their TSVs (and the distance between their domain-dependent node labels) was small. In fact, such a formulation amounts to finding the maximum cardinality, minimum weight matching in a bipartite graph spanning the two sets of nodes. In a modification of Reyner's algorithm (Reyner, 1977), we combine the above bipartite matching formulation with a greedy, best-first search in a recursive procedure to compute the corresponding nodes in two rooted DAGs which, in turn, yields an overall similarity measure that can be used to rank the candidate. Details of the algorithm can be found in Siddiqi et al. (1999b); Macrini (2003).

10.5.1 Node Similarity

The above matching algorithm requires a node similarity function that compares the shapes of the 3D parts associated with two nodes. A variety of the measures used in the literature as signatures for indexing entire 3D models could be used to compute similarities between two parts (nodes) (Osada et al., 2002; Ankerst et al., 1999; Vranic and Saupé, 2001; Elad et al., 2001; Kazhdan et al., 2003a). Some care would of course have to be taken in the implementation of methods which require a form of global alignment. In the experiments carried out in this chapter we have opted for a much simpler 1D signature vector, which is based on the use of a mean curvature histogram. The essential idea is to compute a distribution of mean curvature values over all the level sets of the Euclidean distance function within the interior of a part. This is implemented as follows.

First, consider the volumetric part that a node i represents, along with its Euclidean distance function D . At any point within this volume, the mean curvature of the iso-distance level set is given by $\text{div}(\frac{\nabla D}{\|\nabla D\|})$. On a voxel grid with unit spacing the observable mean curvatures are in the range $[-1, 1]$ because the smallest principal curvature that can be measured corresponds to a sphere having radius 1. We compute a histogram of the mean curvature over all voxels in the volumetric part, over this range, using a fixed number of bins N . A mean curvature histogram vector \mathbf{M}_i is then constructed with entries representing the fraction of total voxels in each bin. The similarity between two nodes i and j is then based on an L_2 distance between their mean curvature histogram vectors:

$$\text{Similarity}(i, j) = [1 - \underbrace{\sqrt{\sum_{k=1}^N [\mathbf{M}_i(k) - \mathbf{M}_j(k)]^2}}_{\text{Distance}(i, j)}].$$

By construction, this similarity function is in the interval $[0, 1]$. This measure could be further modified to take into account overall part sizes. In the experiments described in the following section we choose not to do this since our object models have undergone a global size normalization.

10.6 Experimental Results

In order to test our 3D object retrieval algorithms we have used selected models from the Princeton Shape Benchmark (Shilane et al., 2004). This standardized database, which contains 1,814 3D object models organized by class, is an effective one for comparing the performance of a variety of methods including those in (Kazhdan et al., 2003a; Osada et al., 2002; Ankerst et al., 1999; Vranic and Sauppe, 2001; Elad et al., 2001). However, this database contains only a limited number of models with articulating parts and hence we have supplemented it with a set of articulated models that we have created. The resulting database, which we call the *McGill Shape Benchmark*, includes 455 exemplars of which we have used 425 in our experiments. The full database can be viewed under <http://www.cim.mcgill.ca/~shape>. The exemplars span 19 *basic level* object classes (hands, humans, teddy bears, spectacles, ants, octopuses, snakes, crabs, spiders, tables, chairs, cups, airplanes, birds, dolphins, dinosaurs, four-legged animals, fish). These classes are divided into two categories, those with significant part articulation, and those with moderate or no part articulation. In our experiments we merge the categories “four-legged” and “dinosaurs”, treating them as a single category “four-limbs” Fig. 10.6 depicts 5 exemplars from each of the object classes.

To obtain a fully satisfactory set of exemplars, one would have to sample from a large population of models to be recognized, both with and without articulating parts. We have attempted heuristically to accomplish this in a small way, but carefully achieving that goal is currently beyond the scope of our experimental work. The results which follow must be interpreted with this caveat.

10.6.1 Matching Results

On a large database we envision running the indexing strategy first to obtain a smaller subset of candidate 3D models and to match the query only against these. However, given the moderate size of our database we were able to generate the $425 \times 425 = 180,625$ pairs of matches in a matter of 25–30 minutes

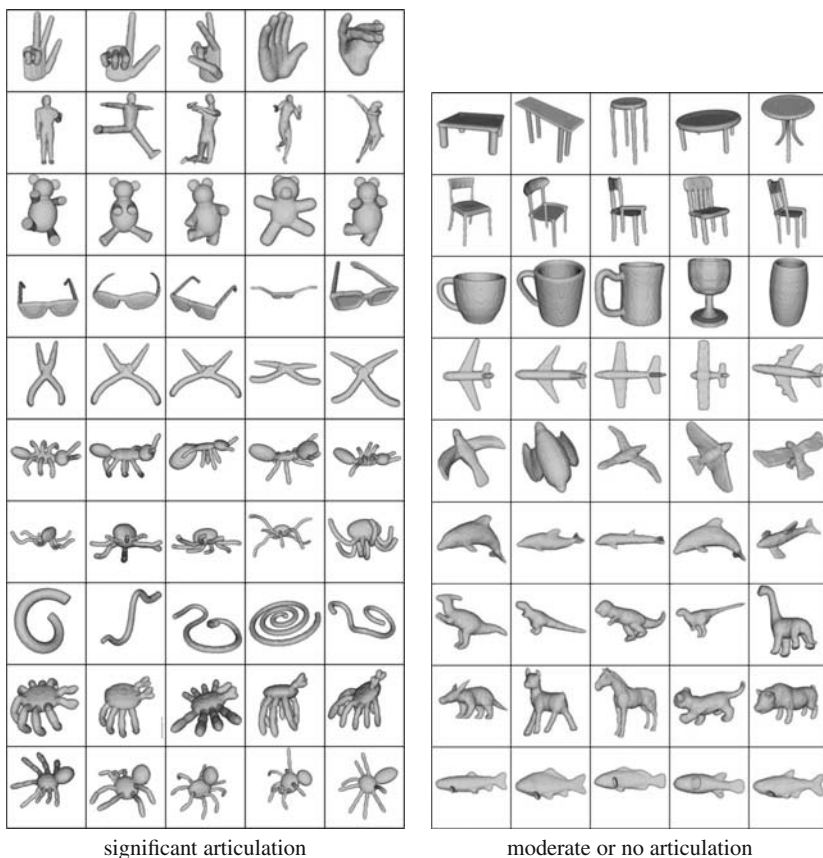


Fig. 10.6 The McGill Shape Benchmark: 5 exemplars are shown from each of the 19 object classes. Exemplars from classes on the left have significant part articulation, whereas those on the right have moderate to no part articulation. The full database of 455 models can be viewed at <http://www.cim.mcgill.ca/~shape/benchMark/>

on a 3.0 GHz desktop PC. We compared the results using medial surfaces (MS) with those obtained using harmonic spheres (HS) (Kazhdan et al., 2003b) and shape distributions (SD) (Osada et al., 2002). For both HS and SD we used as input a mesh representation of the bounding voxels of the voxelized model used for MS. The pair-wise distances between models using harmonic spheres were obtained using Michael Kazhdan's executable code (<http://www.cs.jhu.edu/~misha>) and those using shape distributions were based on our own implementation of the algorithm described in Osada et al. (2002). For this latter implementation we took care to sample points uniformly and randomly on each outward face of each boundary voxel so that the signature curves were faithful. In particular, we were able to reproduce several of the D_2 shape distributions in Fig. 3 of Osada et al. (2002). The comparisons between the three techniques were performed using the standard

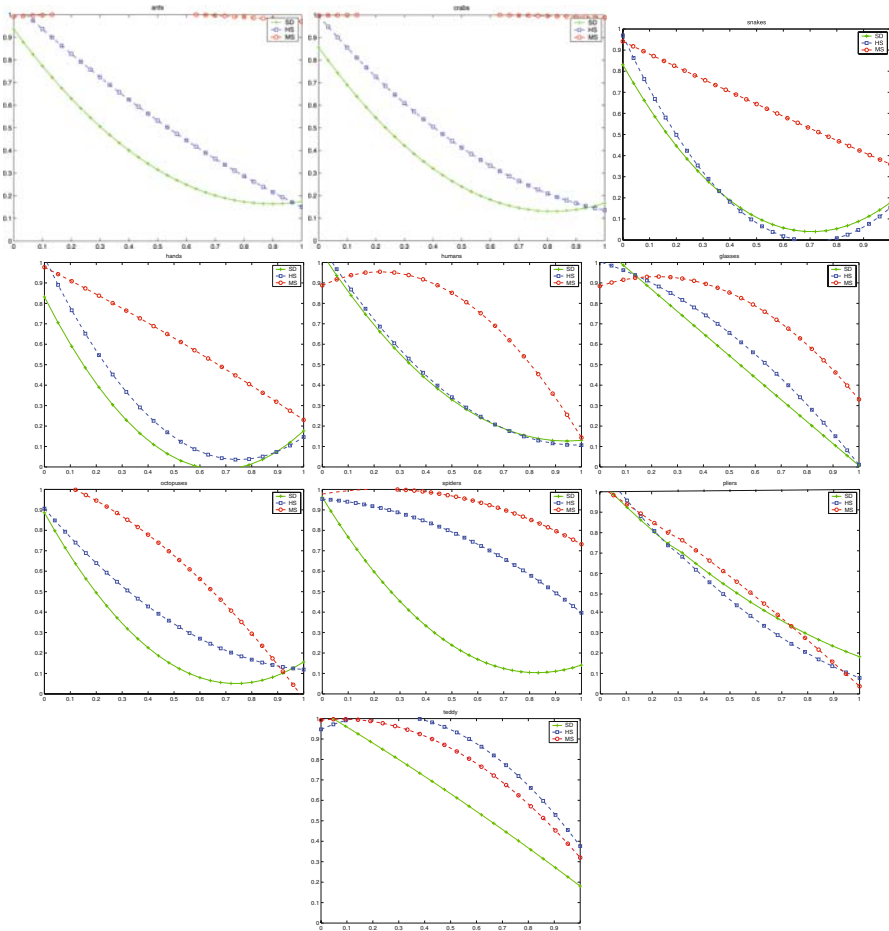


Fig. 10.7 Precision (y axis) versus Recall (x axis): Objects with articulating parts. The results using medial surfaces (MS) are shown with red circles, those using harmonic spheres (HS) with blue squares and those using shape distributions (SD) with green crosses. *Top row:* Ants, crabs, snakes. MS gives superior results. *Second row:* Hands, humans, spectacles. MS gives superior results. *Third row:* Octopuses, spiders, pliers. MS gives superior results. *Fourth row:* Teddy bears. HS gives slightly better results than MS

information retrieval notion of *precision versus recall*, where curves shifted upwards and to the right indicate superior performance.

The results for objects with articulating parts are presented in Fig. 10.7. For the category teddy bears both MS and HS give excellent results. However, for all other categories MS outperforms the other two techniques. For most of these models part structure is largely preserved, but parts articulate and deform. A particularly interesting case is the category snakes, whose exemplars consist of a single tube like

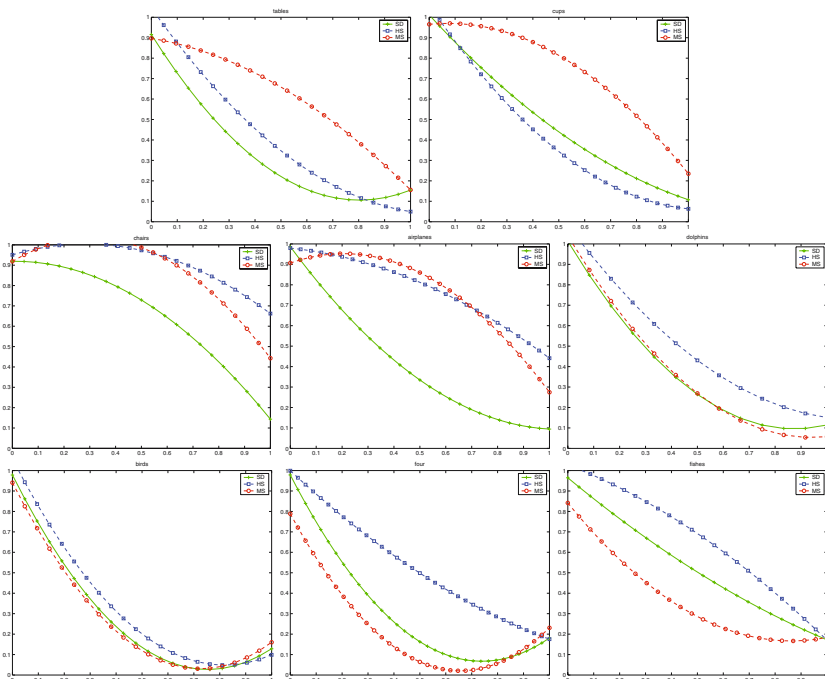


Fig. 10.8 Precision (y axis) versus Recall (x axis): Objects with moderate or no articulation. The results using medial surfaces (MS) are shown with red circles, those using harmonic spheres (HS) with blue squares and those using shape distributions (SD) with green crosses. *Top row:* Tables, cups. MS gives superior results. *Second row:* Chairs, airplanes, dolphins. MS and HS give comparable results for chairs and airplanes. For dolphins HS gives superior results. *Third row:* Birds, four-limbed, fishes. The results are comparable for birds. For four-limbed and fishes MS and SD give superior results

structure that is deformed in a variety of ways, causing significant difficulty for both HS and SD.

Figure 10.8 shows the results for objects with moderate or no part articulation. For categories in the top row MS gives superior results. For categories in the middle row HS and MS give comparable results, with the exception of dolphins for which HS gives superior results. For categories in the third row the results are comparable for birds, but for four-limbs and fish, both HS and SD outperform MS. The HS technique does particularly well on these categories, taking advantage of the pose alignment of the four-limbed models, and the “flat” mass distribution of the fish models. The MS technique would requires a degree of regularization to handle categories with changing part structure; we shall discuss this limitation further in Section 10.7.

10.6.2 Indexing Results

In order to test our indexing algorithm, which utilizes only the topological structure of medial surface-based DAGs, we carried out two types of experiments, using 320 models from the McGill Shape Benchmark (we excluded the categories ants, octopuses, snakes, crabs and spiders). In the first we evaluated percentage recall. For a number of rank thresholds the percentage of models in the database in the same category as a query (not including the query itself) with higher indexing rank, are shown in Fig. 10.9. The results indicate that on average 70% of the desired models are in the top 80 (25% of 320) ranks. In the second experiment we examine the average ranks according to object classes. For all queries in a class the rank of all other objects in that class is computed. The ranks averaged across that class are shown in Fig. 10.10. The results indicate that for 9 of the 13 object classes the average rank is in the top 80 (25% of 320). The higher average ranks for the remaining classes are due to the fact that certain categories have similar part decompositions. In such cases topological structure on its own is not discriminating enough, and part shapes also have to be taken into account.

It should be emphasized that the indexer is a fast screener which can quickly prune the database down to a much smaller set of candidates to which the matcher can be applied. Furthermore, the eigen-characterization used to compute the index is also used at matching time, so the same eigen-structure calculation is exploited for both steps. The systems against which we have evaluated the matcher in the previous section (SD and HS) run a linear search on the entire database for each

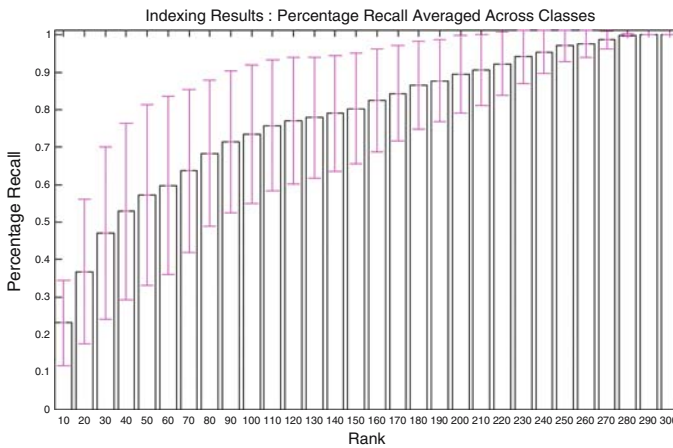


Fig. 10.9 Indexing results: percentage recall. For several rank thresholds, $N = 10, 20, \dots$, we plot the percentage of models in the database in the same category as the query (not including the query itself) with indexing rank $\leq N$. The results averaged across all classes are shown along with error bars depicting ± 1 standard deviation

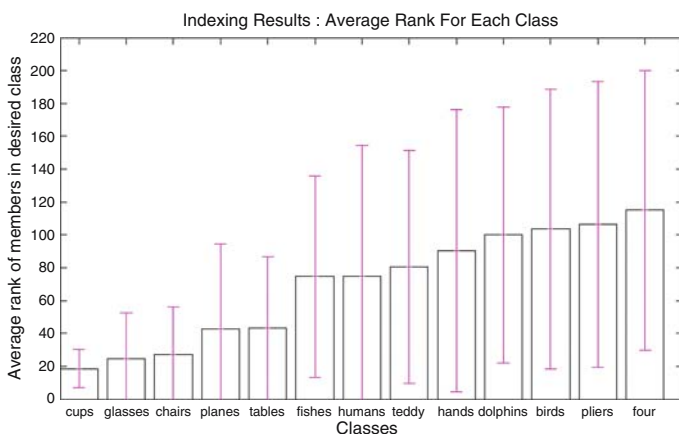


Fig. 10.10 Indexing results: average ranks. For all queries in a class the rank of all other objects in that class are computed. The ranks averaged across that class are shown, along with error bars depicting ± 1 standard deviation

query. That approach may not scale well, since the indexing problem is essentially ignored.

10.7 Discussion and Conclusion

Medial representations have the potential to advance the state-of-the-art in 3D object model retrieval, particularly for databases in which exemplars within the same object class undergo significant part articulation (assuming moderate changes to the part structure). In this chapter, using the method for computing and segmenting medial surfaces of Chapter 4, we have proposed a DAG representation that captures a notion of part saliency. We have then built on algorithms in the computer vision literature to address the problem of 3D model indexing and matching in a uniform framework and have presented retrieval results on a database including models with articulating parts.

The major current limitations of this approach include (1) the assumption that the original object models can be voxelized, (2) the coarse nature of the part similarity measure based on mean curvature histograms, (3) the assumption that objects with complex part topologies can yield stable graph structures using medial surface decompositions on a digital lattice. We discuss each of these weaknesses in turn.

First, it is feasible to make voxelization possible by “patching” models with a few missing triangles. However, for models with incomplete surfaces and large holes, and hence no well defined notion of an interior and an exterior, medial surface-based DAGs would not be appropriate. In current research we are incorporating

computational geometry techniques for computing the Euclidean distance function directly from a mesh, which provides the additional advantage that the points on the shrinking sphere used to measure the average outward flux can be sampled very densely using a coarse-to-fine algorithm (Stolpner and Siddiqi, 2006). It might also be fruitful to explore Voronoi methods, discussed in Chapters 6 and 7, for computing medial surface-based DAGs that could in principle be applied directly to point clouds, provided that the sampling density is high enough (Amenta et al., 2001b) or to use the shock scaffold technique (Leymarie and Kimia, 2003).

With regard to the limitations of the part similarity measure, we expect that the performance of graph theoretic algorithms for comparing medial surface based representations will improve with more discriminating measures, and any one of a number suggested in the literature can be investigated.

The instability for complex objects of the graph structures we compute, as exemplified by the poorer results on the four-limbed animals and the fish, has a variety of aspects. One aspect has to do with the assumption that we have made in converting a medial surface to a DAG, that an object has a well-defined part hierarchy. Such an assumption can fail for objects which have several main parts of comparable sizes (e.g., a caterpillar). Since this property would in turn be reflected in component parts with approximately the same node saliency, such models could at least be flagged. A second aspect has to do with instabilities in the branching topology of a medial surface based DAG, e.g., the precise manner in which the limbs attach to the torso can change with part deformation and movement. This latter aspect can be dealt with, at least in part, by exploring coarser representations based on the medial surface, e.g., by using Damon's metric measure along with the medial integrals developed in Chapter 4 to develop and incorporate a notion of ligature (Blum, 1973) in 3D. A third aspect has to do with the sensitivity of segmentation techniques that use only digital labelings on a rectangular lattice. These can suffer from discretization artifacts. As mentioned above, we are currently carrying out research in the direction of using computational geometry approaches to apply the average outward flux implementation in 3D directly to a mesh, as well as to refine the sampling (Stolpner and Siddiqi, 2006). Preliminary evidence suggests that the medial surfaces so obtained are more precise and that they may allow for estimates of the differential geometry, specifically the expectation that at medial surface junctions there is a discontinuity in the tangent plane, to be used to improve the segmentation process.

Acknowledgements We are grateful to Sylvain Bouix and Ran Chen for help with the numerical simulations and the preparation of the 3D models. This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada, the Canadian Foundation for Innovation, FQRNT Québec, CITO, IRIS, and PREA.