ORIGINAL PAPER

Retrieving articulated 3-D models using medial surfaces

**Kaleem Siddiqi · Juan Zhang · Diego Macrini ·
Ali Shokoufandeh · Sylvain Bouix · Sven Dickinson**

Abstract We consider the use of medial surfaces to
represent symmetries of 3-D objects. This 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. We dem-
onstrate the use of this representation for 3-D object model
retrieval. Our formulation uses the geometric information
associated with each node along with an eigenvalue labeling
of the adjacency matrix of the subgraph rooted at that node.
We present comparative retrieval results against the tech-
niques of shape distributions (Osada et al.) and harmonic
spheres (Kazhdan et al.) on 425 models from the McGill
Shape Benchmark, representing 19 object classes. For objects
with articulating parts, the precision vs recall curves using
our method are consistently above and to the right of those
of the other two techniques, demonstrating superior retrieval
performance. For objects that are rigid, our method gives
results that compare favorably with these methods.

K. Siddiqi (✉) · J. Zhang
School of Computer Science and Centre for Intelligent Machines,
McGill University, Montreal, QC H3A 2A7, Canada
e-mail: siddiqi@cim.mcgill.ca

J. Zhang
e-mail: juan@cim.mcgill.ca

D. Macrini · S. Dickinson
Department of Computer Science, University of Toronto,
Toronto, ON M5S 3G4, Canada
e-mail: dmac@cs.toronto.edu

S. Dickinson
e-mail: sven@cs.toronto.edu

A. Shokoufandeh
Department of Computer Science, Drexel University,
Philadelphia, PA 19104, USA
e-mail: ashokouf@cs.drexel.edu

S. Bouix
Psychiatry Neuroimaging Laboratory,
Harvard University Medical School,
Boston, MA 02115, USA
e-mail: sylvain@bwh.harvard.edu

1 Introduction

3-D object recognition has long been a mainstay of the object
recognition community, with excellent surveys found in
[4,10,11]. With an explosive growth in the number of 3-D
object models stored in web repositories and other databases,
this problem has more recently attracted interest from the
graphics community. Recent advances include query-based
search engines which employ promising measures including
spherical harmonic descriptors [25] and shape distributions
[35]. Such systems can yield impressive results on databases
including hundreds of 3-D 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 are simple
enough so that efficient algorithmic implementations can be
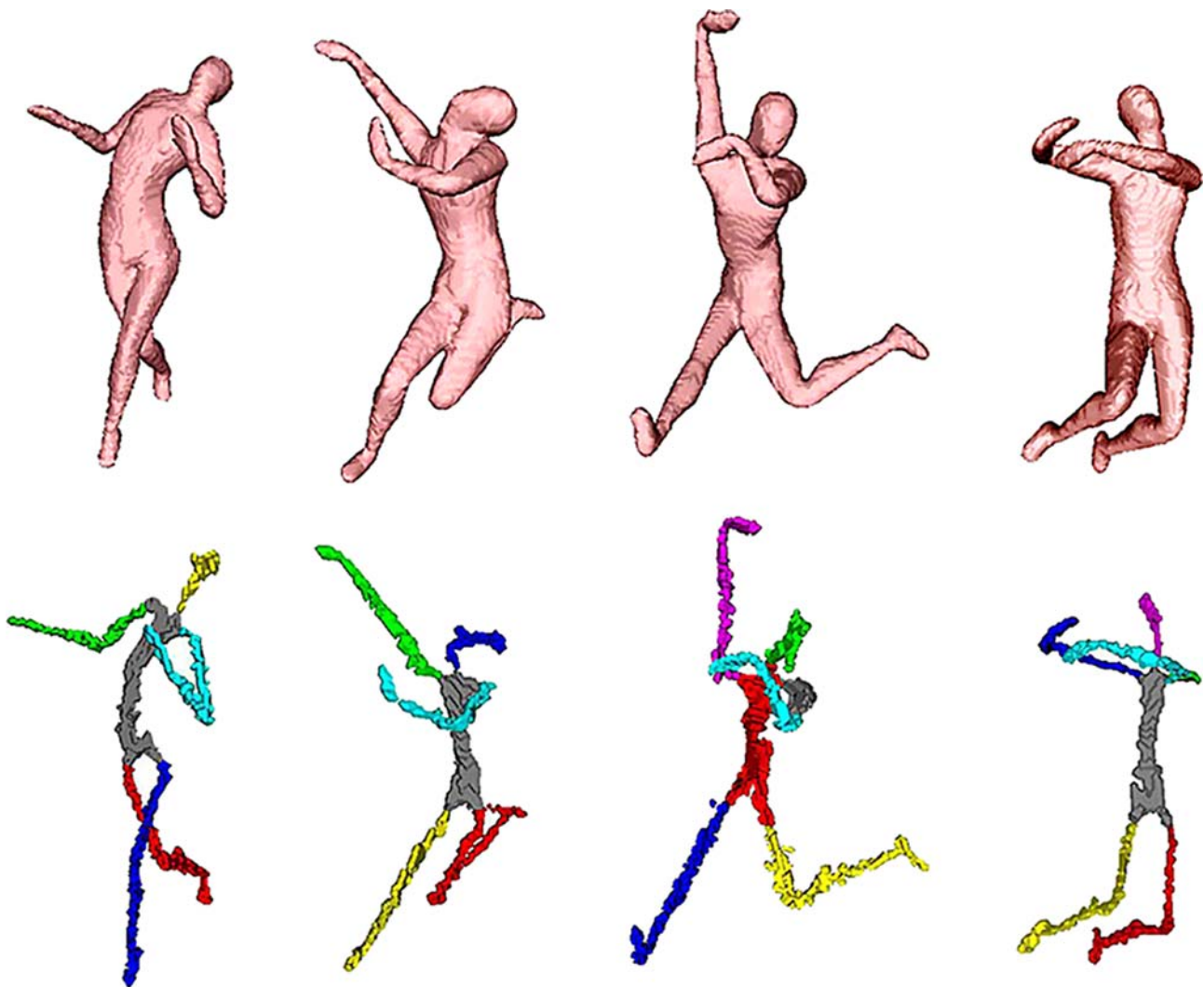
Fig. 1 Exemplars of the object class "human" created by changes in pose and articulations of parts (*top row*). The medial surface (or 3-D skeleton) of each is computed using the algorithm of [52] (*bottom row*).

The medial surface is automatically partitioned into distinct parts, each shown in a *different color*

sought. However, an inevitable cost is that such measures are inherently coarse, and can be sensitive to the deformation or articulation of object parts. As a motivating example, consider the 3-D models in Fig. 1. These four exemplars of an object class were created by combining part articulation with changes in pose. For such examples, the notion of a center of mass or an extrinsic reference point [1], which is required for the computation of descriptions such as shape histograms (sectors or shells) [3] or spherical extent functions [59], can be non-intuitive. 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. As well, measures based on reflective symmetries [24], and signatures based on 3-D moments [18] or chord histograms [35] are not invariant under such transformations.

The computer vision community has grappled with the problem of *generic* level object recognition by suggesting representations based on volumetric parts, including generalized cylinders, superquadrics and geons [5,6,31,37]. 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 3-D 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 [20] where information theoretic measures are used to compare probability distributions representing 3-D object surfaces. In the domain of graph theory there have also been attempts to address the problem of 3-D shape matching using

representations based on Reeb graphs [21,48]. These allow for topological properties to be captured, at least in a coarse sense.

An alternative approach is to use 3-D medial loci (3-D skeletons), obtained by considering the locus of centers of maximal inscribed spheres along with their radii [7]. As suggested by Blum, this offers the advantage that a graph of parts can be inferred from the medial manifolds, which capture local symmetries of the object. To motivate this idea, consider once again the human forms of Fig. 1. A medial surface-based representation (bottom row) provides a natural decomposition, which is largely invariant to the articulation and bending of parts.

Whatever the representation, an efficient indexing mechanism is required to select a few promising candidates for a more costly verification. If the models in the database are organized judiciously, an exhaustive search can be avoided. A decision tree [22,32] is a mechanism for hierarchically partitioning a database. A query shape is matched to the root, and depending on the results of the match, the process is applied recursively to one of its children. At each step, the space of possible models is reduced. Within this framework, a spectral graph decomposition was reported by Sengupta and Boyer for the partitioning of a database of 3-D models, where nodes in a graph represent 3-D surface patches [45].

A closely related approach to the partition (decision tree) scheme described above is to organize the database into a set of prototypes (clustering). In this case, the database is organized by grouping similar objects and choosing a representative (prototype) for each group. This idea can be recursively applied, forming a hierarchical representation of the database. Shapiro and Haralick [46] used a simple relational distance metric, followed by either clustering by similar values of the metric or by constructing a binary decision tree, to organize a large database of relational models. Sengupta and Boyer [44] presented one of the earliest frameworks for object recognition through a hierarchically structured database of parametric structural description graphs. The hierarchical structure was constructed through clustering and computing representative members of each cluster.

The decision tree and hierarchical organization approaches described above attempt to reduce the complexity of the query (indexing) process by reducing the number of database comparisons that need to be made. Further indexing efficiency can be gained through a reduction in the dimensionality of the index. If the query takes the form of a vector, then classical nearest-neighbor search can be used to find similar candidates in sublinear time; examples include Turk and Pentland [58], Murase and Nayar [33], Sclaroff and Pentland [42], and Lowe [29]. If the query takes the form of a structured (e.g., graph) representation, it must be abstracted to a vector. Sossa and Horaud [55] and Shokoufandeh et al. [51]

have proposed powerful spectral abstractions of graph structure that support efficient indexing.

In this article, we build on a recent technique to compute medial surfaces [52] 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 [51] to tackle the problems of matching and indexing 3-D object models. These and related algorithms have already shown promise in the computer vision community for generic level view-based object indexing and matching using 2-D skeletal graphs [36,43,50,54]. They have also been demonstrated in the context of matching 3-D object models with tubular parts, using a centerline approximation of the 3-D medial surface into a *curve* skeleton. In [57], skeletal fragments were mapped to nodes in an undirected graph, and a precursor to the matching algorithm described in this paper was used to compare query and model graphs. Later, in [12], an improved curve skeleton yielded graphs which were mapped many-to-many, allowing the restriction of one-to-one node correspondence to be overcome.

We demonstrate the significant potential for medial surface-based 3-D object retrieval with experimental results on 425 models representing 19 object classes from the McGill Shape Benchmark [60], 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 for objects with articulated parts this method outperforms the techniques of shape distributions [35] and harmonic spheres [25]. To our knowledge these are the first comprehensive empirical results on the use of medial surfaces and their graph spectra in the context of 3-D object model matching and indexing.

2 Medial surfaces and DAGs

We now review the salient aspects of medial surfaces and the average outward flux-based algorithms for computing them that are the basis for the 3D object retrieval experiments reported in this article. We point the interested reader to the book [53], which provides a comprehensive coverage of the mathematics, algorithms and applications of medial representations.

Recent approaches for computing 3-D skeletons include those which use Voronoi techniques [2,13,28,34] and those based on distance functions [9,39,52]. Methods in the first class have the advantage that they can be employed on input data in the form of points sampled from (or meshes describing) an object's surface. Unfortunately, automatic segmentation of the resulting skeletons into their constituent components remains an unsolved problem. Methods in the second class are typically based on digital distance functions [8] and hence assume that the object models have

first been voxelized. Whereas this adds a computational burden these methods provide the advantage that the digital classification of [30] allows for the taxonomy of generic 3-D skeletal points [19] to be interpreted on a rectangular lattice, leading to a graph of parts. In the current article we choose to employ the average outward flux-based method of [52] which uses limiting properties of the gradient of the Euclidean distance function. The mathematical and computational properties of this approach, in comparison against Voronoi methods and those based on height ridges, are further developed in [38]. Whereas the implementation we use in the current article requires a voxelized model as input, an extension is now available which can handle 3D objects with surfaces described by polyhedral meshes [56].

2.1 Average outward flux

Let D be defined as the Euclidean distance function of each point within a 3-D object to the closest point on the object's boundary. It is well known that the singularities of D coincide with the skeleton. Let $S = \partial R$ be the bounding surface of a region R within the object, with surface area element dS. The outward flux of the vector field ∇D through S is defined as the surface integral

$$OF = \int \int_S \nabla D \cdot \mathbf{N} \, dS,$$

where $\mathbf{N}$ is the unit outward normal at each point of S. The divergence theorem relates the outward flux to the divergence of ∇D

$$\int \int \int \text{div}(\nabla D) \, dV \equiv \int \int_S \nabla D \cdot \mathbf{N} \, ds, \qquad (1)$$

where dV is a volume element. The average outward flux is defined by normalizing the outward flux by the surface area of S.

$$AOF = \frac{\int \int_S \nabla D \cdot \mathbf{N} \, dS}{\text{area}(S)}. \qquad (2)$$

It is a standard fact that the outward flux measures the degree to which the flow generated by ∇D is volume preserving in 3-D, for the region over which it is computed. To elaborate, the outward flux (and hence also the average outward flux) is negative if the volume enclosed by the region ∂R is shrinking under the action of the flow, positive if it is growing and zero otherwise. This quantity is clearly strongly dependent on the shape of the region R. When considering a region R that contains a medial point, unfortunately the standard form of the divergence theorem does not apply since the vector field ∇D is multi-valued. Instead, the limiting behavior of the average outward flux as the region R shrinks to

a point can be considered. If R is a convex region shrunk by a constant factor in every direction, [14] has shown that for non-medial points, this limit is zero and that for medial points there is a constant $c_R > 0$, depending on the shape of the region R, such that the average outward flux approaches a strictly negative number bounded above by $c_R < \nabla D \cdot \mathbf{N}' >$, where $\mathbf{N}'$ is now a one-sided normal to the medial set. The outward flux on the other hand tends to zero at every point, medial or not. The proof of these results relies on an alternate form of the divergence theorem that can be applied in regions intersecting the medial set,[1] and which is developed in some detail in [14–16].

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) 3-D model. The average outward flux of ∇D is computed through unit spheres centered at each rectangular lattice point, using Algorithm 1. The choice of an isotropic region R (a sphere) is appropriate in the absence of any prior information, and has the additional feature that the limiting value of the average outward flux then has the property that it reveals the object angle, as explained in [15,16]. 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. Note that whereas Algorithm 1 uses a coarse sampling of the sphere, based a digital neighborhood, this could be refined for more precise calculations. In particular, the extension proposed in [56] uses computational geometry techniques to sample points densely on a shrinking sphere and for these sampled points, close to exact distance D to the boundary mesh is used.

Algorithm 1: Average Outward Flux.

Data : Voxelized 3-D Object Model.
Result : Average Outward Flux Map.
Compute the Euclidean distance transform D of the model ;
Compute the gradient vector field ∇D;
Compute the average outward flux of ∇D:

For (each point $\mathbf{x}$) $\text{AOF}(\mathbf{x}) = \frac{1}{26} \sum_{i=1}^{26} < \mathbf{N_i}, \nabla D(\mathbf{x}_i) >$;

(where $\mathbf{x}_i$ is a 26-neighbor of $\mathbf{x}$ in 3-D and $\mathbf{N_i}$ is the outward normal at $\mathbf{x}_i$ of the unit sphere centered at $\mathbf{x}$)

2.2 Topology preserving thinning

The average outward flux measure is used to drive a thinning process (Algorithm 2) that does not alter the digital topology

---

[1] Such a form is necessary because the divergence of ∇D is not defined at medial points, where ∇D is multi-valued.

of the object. This is done by identifying each *simple* or removable point **x**, for which a characterization based on the 26-neighborhood of each lattice point **x** is provided in [30]. With O being the set of points in the interior of the voxelized object and N_{26}^* being the 26-neighborhood of **x**, not including **x** itself, this characterization is based on two numbers:

1. C^*: the number of 26-connected components 26-adjacent to **x** in $O \cap N_{26}^*$, and
2. $\bar{C}$: the number of 6-connected components 6-adjacent to **x** in $\bar{O} \cap N_{18}$.

It can be shown that a digital point **x** is *simple* if $C^*(\mathbf{x}) = 1$ and $\bar{C}(\mathbf{x}) = 1$.

Algorithm 2: Topology Preserving Thinning.

Data : 3-D Object Model, Average Outward Flux Map.
Result : 3-D Skeleton (Medial Surface).
for *(each point **x** on the boundary of the object)* **do**
 if *(**x** is simple)* **then**
  insert(**x**, maxHeap) with AOF(**x**) as the sorting key for insertion;

while *(maxHeap.size > 0)* **do**
 x = HeapExtractMax(maxHeap);
 if *(**x** is simple)* **then**
  if *(**x** is an end point)* and *(AOF(**x**) < Thresh)* **then**
   mark **x** as a medial surface (end) point;
  else
   Remove **x**;
   for *(all neighbors **y** of **x**)* **do**
    if *(**y** is simple)* **then**
     insert(**y**, maxHeap) with AOF(**y**) as the sorting key for insertion;

The taxonomy of generic 3-D skeletal points in the continuum, i.e., those which are stable under small perturbations of the object, is provided in [19]. Using the notation A_n^k, where n denotes the number of points of contact of the maximal inscribed sphere with the surface and k the order of these contacts, the taxonomy includes: (1) A_1^2 points which form a smooth medial manifold, (2) A_3 points which correspond to the rim of a medial manifold, (3) A_1^3 points which represent the intersection curve of three medial manifolds, (4) an A_1^4 point at the intersection of four A_1^3 curves, and (5) an $A_1 A_3$ point at the intersection between an A_3 curve and an A_1^3 curve.

It is clear from this classification that 3-D skeletons are essentially comprised of medial manifolds, their rims and intersection curves, and this is why we refer to this as a *medial surface* representation. As shown in [30], the numbers C^* and

Table 1 The topological classification of Malandain et al.

| $\bar{C}$	C^*	Type
0	Any	Interior point
Any	0	Isolated point
1	1	Border (simple) point
1	2	Curve point
1	>2	Curves junction
2	1	Surface point
2	>2	Surface-curve(s) junction
>2	1	Surfaces junction
>2	≥2	Surfaces-curves junction

$\bar{C}$ can also be used to classify surface points, rim points, junction points and curve points on a rectangular lattice. These results are summarized in Table 1. 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 three models in Fig. 2. Qualitatively the segmented medial surfaces provide intuitive part decompositions. However, these decompositions are not perfect because the topological labeling uses only local (digital) connectivity, but no geometric information. As one example, the head and neck region of the deer model are not separated from the two ears because on a discrete lattice the voxels in the connecting regions correspond to the same manifold. We shall discuss this issue further in Sect. 6. However, we emphasize that our matching and indexing techniques [51,54], which use an eigenvalue labeling to characterize the topological structure of parts, can tolerate a degree of variation in part decomposition.

2.3 From medial surfaces to DAGs

We now propose an interpretation of the segmented medial surface as a directed acyclic graph (DAG), where we shall treat each component as a node. This will in turn allow the subsequent matcher and indexer to cope with both changes in part structure, as reflected by connectivity in the graph, and changes in part shape, as reflected by the geometric information associated with each node. We begin by introducing
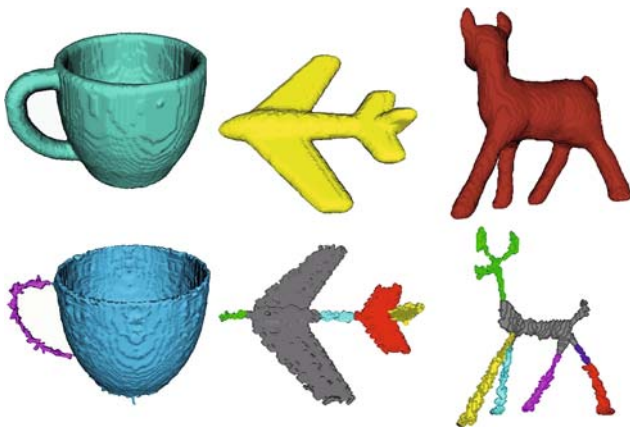
Fig. 2 Voxelized models of a cup, an airplane and a deer (*top row*). Their medial surfaces are computed using the algorithm of [52] and are automatically partitioned into distinct parts, each shown in a different color (*bottom row*)

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 [7]. The contribution of each component to the overall volume can thus be used as a measure of its significance. Since the spheres associated with adjacent components can overlap, an objective measure of component j's saliency is given by

$$\text{Saliency}_j = \frac{\text{Voxels}_j}{\sum_{i=1}^{N_c} \text{Voxels}_i},$$

where N_c is the number of components and $\text{V}oxels_i$ is the number of voxels *uniquely* reconstructed by component i.
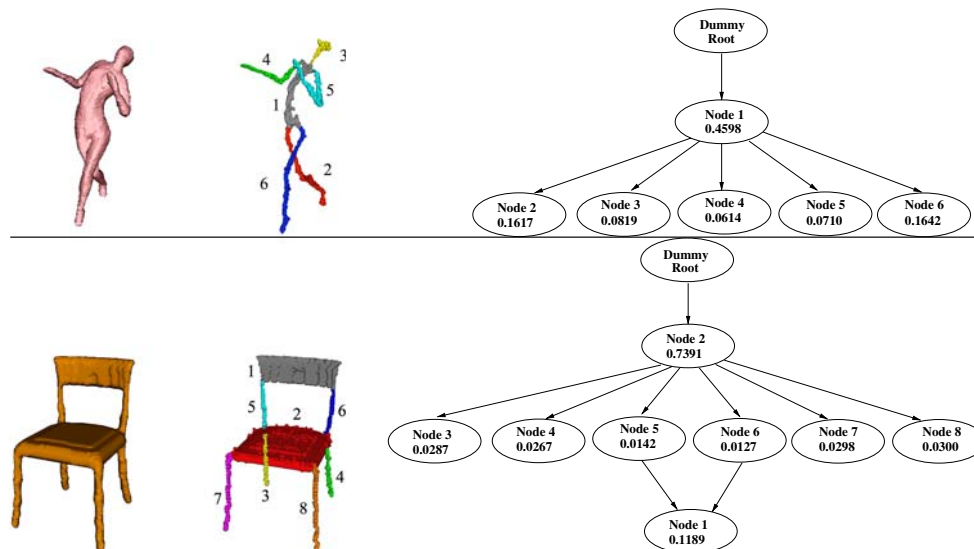
We propose the following construction of a DAG, using each component's saliency. Consider the most salient component as the root node (level 0), and place components to which it is connected as nodes at level 1. Components to which these nodes are connected are placed at level 2, and this process is 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 3-D models comprised of disconnected parts we introduce a single dummy node as the parent of all DAGs for a 3-D model.

This process is illustrated in Fig. 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. Our DAG representation of the medial surface is quite different than the graph structure that follows from a direct use of the taxonomy of 3-D skeletal points in the continuum [19]. Our motivation is to be able to exploit the hierarchical structure indexing and matching algorithms reported in [51, 54]. However, this conversion carries with it some limitations and we shall return to discuss these in Sect. 6.

3 Indexing

A linear search of the 3-D model database, i.e., comparing the query 3-D object model to each 3-D model and selecting the closest one, is inefficient for large databases. An indexing

Fig. 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

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, and 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 isomorphism 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 [51], we draw on the eigen space 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 [51].

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 extraneous 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 [54]. Let V be the root of a DAG whose maximum branching factor is Δ, as shown in Fig. 4. Consider the subgraph 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

$$S_i = |\lambda_1| + |\lambda_2| + \ldots + |\lambda_{k_i}|$$
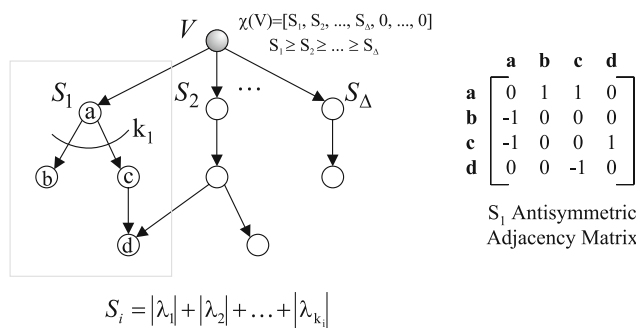
Fig. 4 Forming a low-dimensional vector description of graph structure. At node **a**, we compute the sum of the magnitudes of the k_1 largest eigenvalues of the adjacency sub-matrix defined by the subgraph rooted at **a**. The sorted sums S_i become the components of $\chi(V)$, the *topological signature vector* (or TSV) assigned to V

recursively repeat this procedure, assigning a vector to each nonterminal node in the DAG, computed over the subgraph rooted at that node.

In summing the magnitudes of the eigenvalues, some uniqueness has been lost in an effort to reduce dimensionality. The k_i largest eigenvalues are chosen for two reasons: (1) the largest eigenvalues are more informative of subgraph structure, and (2) by summing k_i elements, the sums are effectively normalized according to the local complexity of the subgraph root, thereby distinguishing subgraphs that have richer part structure at coarser levels. The dimensionality of the TSV, χ, is bounded by the maximum branching factor in the graph, which is typically small, and not by the size of the graph, which can be large for complex 3-D models. The TSV therefore represents a low-dimensional abstraction of the degree distribution (shape) of a DAG. Moreover, the TSV is robust to minor topological deformations of the DAG due to within-class variation, or due to imperfections in the skeletonization and segmentation process.

Indexing now amounts to a nearest-neighbor search in a model database, as shown in Fig. 5. In our experiments, we have used the SR-tree technique proposed by Katayama and Satoh [23]. The TSV of each non-leaf node (the root of
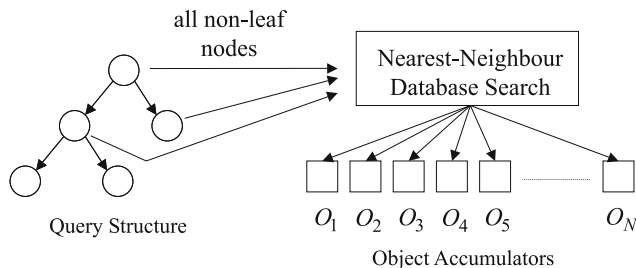
Fig. 5 Indexing mechanism. Each non-trivial (non-leaf) node (whose TSV encodes a topological abstraction of the subgraph rooted at the node) votes for models sharing a structurally similar subgraph. Models receiving strong support are candidates for a more comprehensive matching process

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 TSV "votes" for nearby "model" TSVs, thereby accumulating evidence for models that share the substructure defined by the query TSV. 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. Both the TSV construction and indexing algorithm are described in detail in [51].

4 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 [54].

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. At first glance, such a formulation might seem like a bad idea (by throwing away important graph structure) until one recalls that the graph structure is effectively encoded in the node's TSV. Is it then possible to reformulate a noisy, largest isomorphic subgraph problem as a simple bipartite matching problem?

Unfortunately, in discarding all the graph structure, the underlying hierarchical structure has also been discarded.

There is nothing in the bipartite graph matching formulation that ensures that hierarchical constraints among corresponding nodes are obeyed, i.e., that parent/child nodes in one graph do not match child/parent nodes in the other. This reformulation, although softening the overly strict constraints imposed by the largest isomorphic subgraph formulation, is perhaps too weak. Since no polynomial-time solution is known to exist for enforcing the hierarchical constraints in the bipartite matching formulation, an approximate solution to finding corresponding nodes between two DAGs, subject to hierarchical constraints and accommodating perturbations, is sought [49, 54].

The key idea is to use a modification of Reyner's algorithm [40], that combines the above bipartite matching formulation with a greedy, best-first search in a recursive procedure to compute the corresponding nodes in two rooted DAGs, as shown in Fig. 6. As in the above bipartite matching formulation, the maximum cardinality, minimum weight matching in the bipartite graph spanning the two sets of nodes from the query and model graphs is computed, as shown in Fig. 6a. Edge weight encodes a function of both topological similarity as well as domain-dependent node similarity, described in the following subsection. The result will be a selection of edges yielding a mapping between query and model nodes. As mentioned above, the computed mapping may not obey hierarchical constraints. We therefore greedily choose only the best edge (the two most similar nodes in the two graphs, representing in some sense the two most similar subgraphs), as shown in Fig. 6b, add it to the solution set, and recursively apply the procedure to the subgraphs defined by these two nodes, as shown in Fig. 6c. Unlike a traditional depth-first search, which backtracks to the next statically determined branch, this algorithm effectively recomputes the branches at each node, always choosing the next branch to descend in a best-first manner. In this way, the search for corresponding nodes is focused in corresponding subgraphs (rooted DAGs) in a top-down manner, thereby ensuring that hierarchical constraints are obeyed. The structural abstraction offered by the TSV effectively unifies the indexing and matching procedures, providing an efficient model retrieval mechanism. Details can be found in [49, 51].

4.1 Node similarity

The above matching algorithm requires a node similarity function that compares the shapes of the 3-D parts associated with two nodes. A variety of the measures used in the literature as signatures for indexing entire 3-D models could be used to compute similarities between medial surface-based parts (nodes) [3, 18, 24, 35, 59], because their shapes are likely to be relatively simple; significant protrusions or elongations will lead to distinct parts being formed. Some care has to be
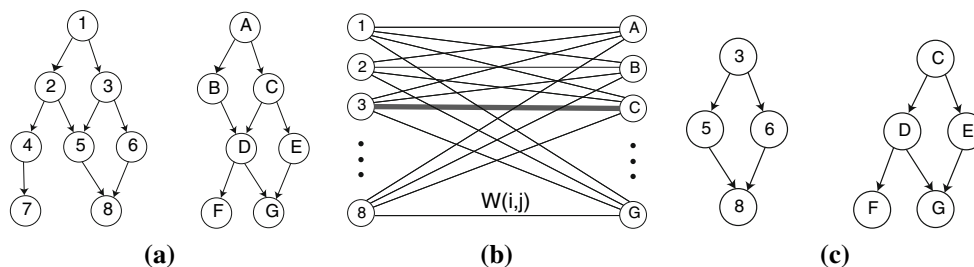
Fig. 6 Matching Algorithm. Given two graphs to be matched (**a**), form a bipartite graph (**b**) spanning their nodes but excluding their edges. The edge weights **W(i,j)** not only encode node content similarity (see Sect. 4), but the structural similarity of their underlying subgraphs, as encoded by the difference in their respective TSV's. The best matching pair is identified, the two nodes are removed from their respective graphs and added to the solution set of correspondences, and the process applied recursively to their subgraphs (**c**)

taken in the implementation of methods which require a form of global alignment. In the experiments carried out in this article we have opted for a 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. Such a signature can distinguish between components that have very different average mean curvatures, such as elongated, blob-like or flat parts. As well, it is somewhat robust to moderate amounts of part bending or twisting. The mean curvature histogram 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 detectable mean curvatures are in the range $[-1, 1]$. 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_b. A mean curvature histogram vector $\hat{M}_i$ 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:

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

By construction, this similarity function is in the interval $[0, 1]$. We have found the L_2 metric to give good empirical results, but other metrics could also be used. Furthermore, this measure could be modified to take into account overall part sizes. In our experiments we choose not to do this since our object models have undergone a global size normalization.

5 Experimental results
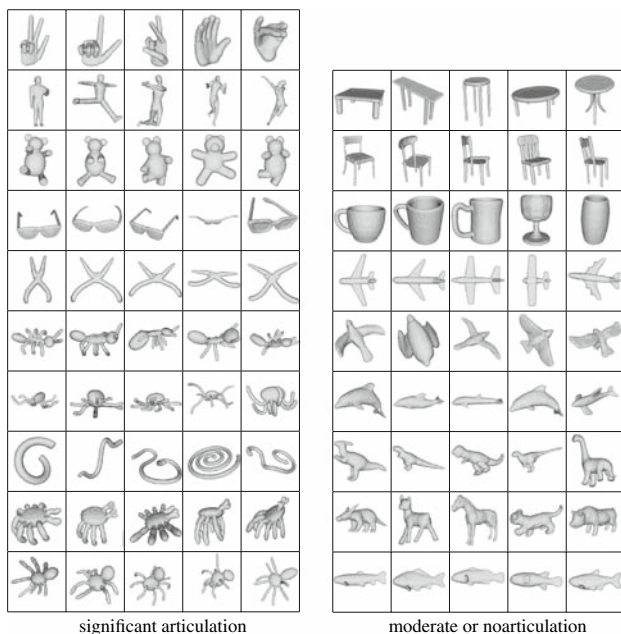
5.1 The McGill shape benchmark

In order to test our 3-D object retrieval algorithms we have considered using the Princeton Shape Benchmark [47]. This standardized database, which contains 1,814 3-D object models organized by class, is an effective one for comparing the performance of a variety of methods including those in [3,18,24,35,59]. However, a majority of the models in the database correspond to rigid, man-made objects. The natural objects include a variety of animals, trees, plants, humans and body parts. However, only a limited number of these have articulated parts. When such models are present, the precise nature of part articulation typically defines a unique base level category. For example, *animal-biped-human* contains human models which are upright, *animal-biped-human-arms-out* contains similar models with outstretched hands and *animal-biped-human-walking* contains those in a walking pose. Results reported in [47] indicate that a number of global shape descriptors perform suitably at such base levels of classification, but degrade rapidly at coarser levels, e.g., the classification *human*. In the context of *generic* 3-D model retrieval, such coarser levels in fact correspond to the notion of a *basic level* or *entry level* categorization [5,41], whose exemplars might reflect a variety of complex poses and articulations, such as those seen in Fig. 1. Our matching and indexing algorithms have the potential to accommodate part articulation and deformation, because the medial surface DAG is a part-based representation.

To demonstrate this, we have constructed the McGill Shape Benchmark [60], adopting several of the models in the Princeton repository, but also adding a substantial number of our own. The database includes 455 models that are available both in voxelized and mesh form. 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, and fish. These classes are divided into two categories,

those with significant part articulation, and those with moderate or no part articulation. Figure 7 depicts 5 representative exemplars from each of the object classes. The full database can be downloaded from http://www.cim.mcgill.ca/~shape.
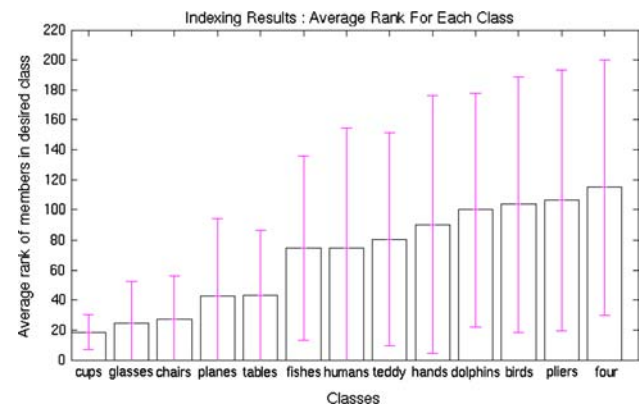
5.2 Indexing results

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. 8. 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. 9. 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.

Fig. 7 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
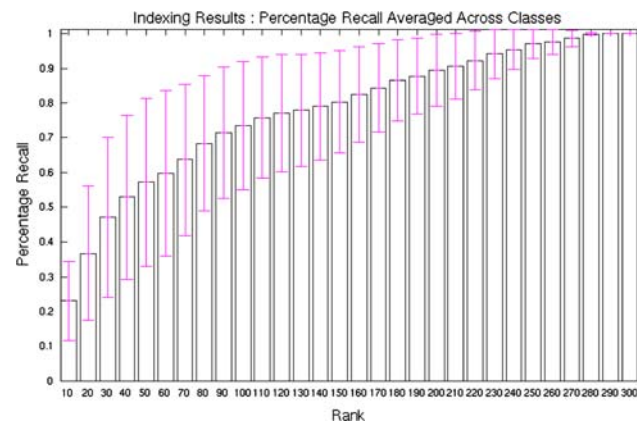
Fig. 8 Indexing results: percentage recall. For several rank thresholds, $N = 10, 20, \ldots$, we plot the percentage of models in the database in the same category as the query (not including the query itself) with indexing rank $\leq N$. The results averaged across all classes are shown along with error bars depicting ± 1 standard deviation

Fig. 9 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

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 will evaluate the matcher quantitatively in Sect. 5.4 run a linear search on the entire database for each query. This approach may not scale well, since the indexing problem is essentially ignored.

5.3 Matching results

For the same set of 320 models used to evaluate the indexer in the previous section, we show the average similarity scores obtained using our medial surface-based matcher (MS) in Table 2, organized by object class. Red and blue boxes are drawn, respectively, around the two highest similarity scores.

Table 2 Average matching results using MS

| Instance	✈	🐦	🪑	☕	🐬	🐎	✋	🧍	🪑	🧸	🐟	👓	✕
✈	.61	.37	.00	.45	.23	.20	.02	.02	.10	.00	.09	.16	.26
🐦	.37	.38	.00	.21	.25	.18	.12	.10	.18	.02	.18	.23	.25
🪑	.00	.00	.51	.29	.17	.15	.07	.03	.00	.15	.02	.00	.07
☕	.45	.21	.29	.64	.34	.23	.04	.04	.00	.01	.05	.04	.28
🐬	.23	.25	.17	.34	.43	.24	.16	.15	.12	.04	.22	.06	.19
🐎	.20	.18	.15	.23	.24	.28	.20	.22	.14	.07	.26	.05	.14
✋	.02	.12	.07	.04	.16	.20	.51	.46	.37	.08	.45	.00	.09
🧍	.02	.10	.03	.04	.15	.22	.46	.53	.29	.03	.47	.02	.06
🪑	.10	.18	.00	.00	.12	.14	.37	.29	.58	.04	.31	.02	.23
🧸	.00	.02	.15	.01	.04	.07	.08	.03	.04	.48	.08	.15	.07
🐟	.09	.18	.02	.05	.22	.26	.45	.47	.31	.08	.56	.02	.12
👓	.16	.23	.00	.04	.06	.05	.00	.02	.02	.15	.02	.71	.21
✕	.26	.25	.07	.28	.19	.14	.09	.06	.23	.07	.12	.21	.40

Each object in the database is matched against all the other objects in the database. Each cell shows the average similarity between objects selected from two fixed object classes. In each row red and blue boxes are drawn, respectively, around the two highest average similarity scores. In all cases the highest score coincides with the correct object class. In most cases there is also a very significant difference between the top two average similarity scores

In all cases the highest score coincides with the correct object class. In most cases there is also a significant difference between the top two average similarity scores. In Fig. 10 we provide examples of the correspondences that the matcher finds between two models. In the first case both models are taken from the class humans and the matcher finds intuitive part correspondences despite significant part articulation. In the second, one model is a table and the other is a chair. The part correspondences found are nonetheless intuitive with the surface of the table matching the seat of the chair, and the legs matching legs. These results illustrate the significant potential of medial surface based representations and their graph spectra for *generic* level 3-D model retrieval.

5.4 Comparative matching results: precision versus recall curves

We now carry a more thorough evaluation of the matcher by providing comparative matching results using a more
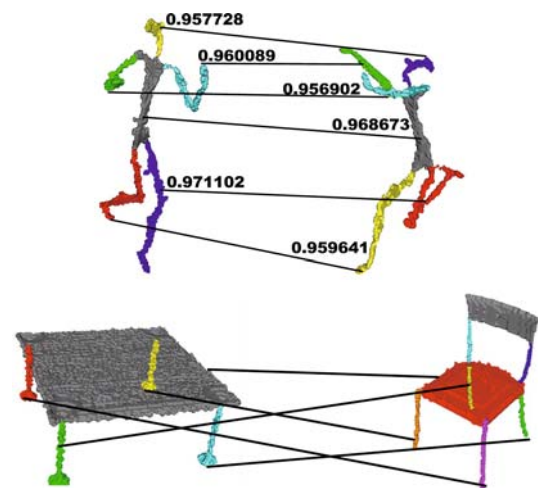

Fig. 10 An illustration of the part correspondences found by the matcher for two cases: a within class match and a between class match. In both examples the nodes of the medial surface-based DAGs are shown with different colors and lines are drawn between corresponding components

complete set of 425 of the 455 models in the McGill Shape Benchmark. This selection includes exemplars from each of the 19 object classes in Fig. 7, but we merge the categories "four-legged" and "dinosaurs", treating them as a single category "four-limbs". The results obtained using medial surfaces (MS) were compared against those obtained using harmonic spheres (HS) [25] and shape distributions (SD) [35]. Since these competing approaches lack an indexing mechanism, and therefore match the (extracted features of the) query to (the extracted features of) *each* model in the database, we turn off our indexing mechanism and perform a similar linear search. On a large database, we envision running the indexing strategy first to obtain a smaller subset of candidate 3-D models and to match the query only against these. However, given the moderate size of our database we were able to generate the $425 \times 425 = 180,625$ matches using medial surfaces (MS) in a matter of 25–30 min on a 3.0 GHz desktop PC.

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 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 [35]. 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 $D2$ shape distributions in Fig. 3 of [35]. The comparisons between the three techniques were performed using the standard information retrieval notions of *precision* and *recall*. Given a query object model, a relevant database object model is considered to be one that belongs to the same category, while an irrelevant database model is one from a different category. Given a number of queries, recall refers to the ratio of the number of relevant models retrieved to the total number of relevant models in the database. Precision refers to the ratio of the number of relevant models retrieved to the total number of (relevant and irrelevant) models retrieved. In a precision versus recall plot it is customary to plot the precision values for each level of recall. Curves shifted upwards and to the right indicate superior retrieval performance. In the results that follow, the precision versus recall plots are grouped according to the category of the query object models.

The results for objects with articulating parts are presented in Fig. 11. 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 structure that is
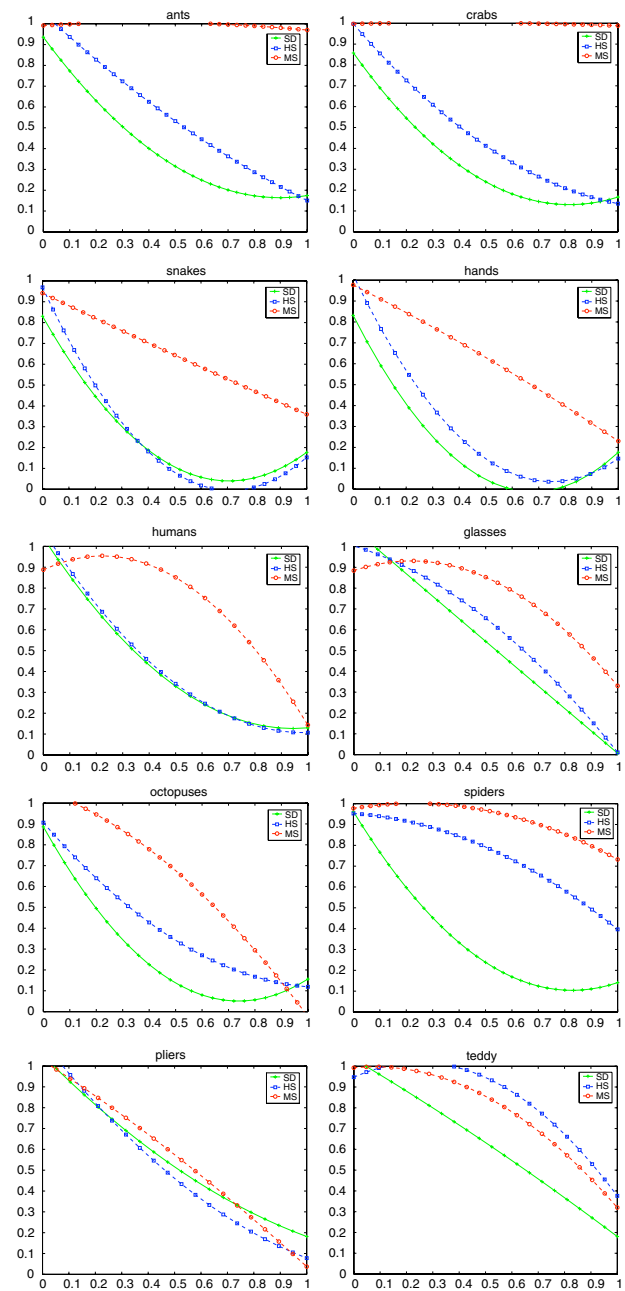
Fig. 11 Precision (*y* axis) versus recall (*x* axis): Objects with articulating parts. The results using medial surfaces (MS) are shown with red circles, those using harmonic spheres (HS) with blue squares and those using shape distributions (SD) with green crosses. Top row Ants and crabs. MS gives superior results. Second Row Snakes and hands. MS gives superior results. Third Row Humans and glasses. MS gives superior results. Fourth Row Octopuses and spiders. MS gives superior results. Fifth Row Pliers and teddy bears. The three methods give comparable results for pliers. HS gives slightly better results than MS for teddy bears

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

Figure 12 shows the results for objects with moderate or no part articulation. For categories in the top row MS gives
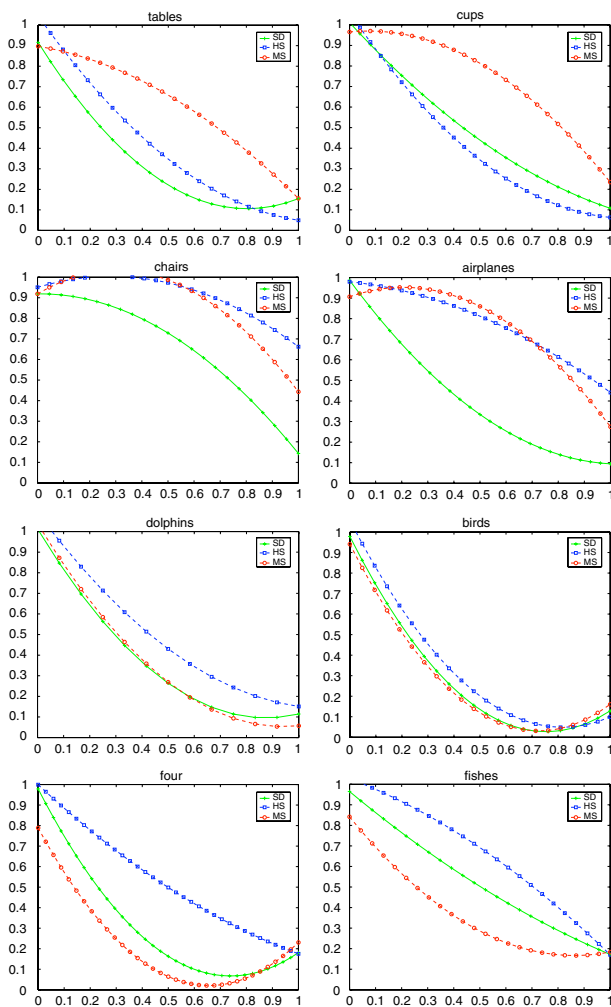
Fig. 12 Precision (*y* axis) versus Recall (*x* axis): Objects with moderate or no articulation. The results using medial surfaces (MS) are shown with red circles, those using harmonic spheres (HS) with blue squares and those using shape distributions (SD) with green crosses. Top Row Tables and cups. MS gives superior results. Second Row Chairs and airplanes. MS and HS give comparable results. Third Row Dolphins and birds. For dolphins HS gives superior results. The three methods give comparable results for birds. Fourth Row Four-limbed and fishes. MS and SD give superior results

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 fishes, 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, and we shall discuss this limitation further in Sect. 6.

6 Discussion and conclusion

Medial representations have the potential to advance the state-of-the-art in 3-D object model retrieval, particularly when exemplars within the same object class undergo significant part articulation (assuming moderate changes to the part structure). In this article, using average outward flux based methods for computing and segmenting medial surfaces, we have proposed a DAG representation which captures a notion of part saliency. We have then built on algorithms in the computer vision literature to address the problem of 3-D model matching and indexing in a unified framework and have presented retrieval results on the McGill Shape Benchmark.

The major current limitations of our 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, and (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 "patch" models with a few missing triangles, so that voxelization becomes possible. 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. We have recently developed an extension of the average outward flux-based skeletonization method that can be applied directly to a mesh [56]. It might also be fruitful to explore Voronoi methods for computing medial surface-based DAGs that could in principle be applied directly to point clouds, provided that the sampling density is high enough [2] or to use the shock scaffold technique [28].

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

The third concern, as exemplified by the poorer results on the four-limbed animals and the fishes, points to some limitations of the current representation and its computation. One aspect has to 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 incorporating Blum's notion of ligature [7], which has helped to regularize 2D shock graphs. It might be fruitful to explore the use of a coarser DAG where only salient regions of the medial surface were retained [17]. A third aspect has to do with the sensitivity of segmentation techniques that use only digital labeling on a rectangular lattice. These can suffer from discretization artifacts. The extended average outward flux-based method developed in [56] has the advantage that the points on the shrinking sphere used to measure the average outward flux can be sampled densely and furthermore the rectangular lattice at which these spheres are centered can be refined in a coarse-to-fine manner. As a consequence, the medial surfaces so obtained are more precise. Preliminary evidence suggests that this improved method may allow for estimates of the differential geometry to be used to aid in the segmentation process, since at medial surface junctions there is an expected discontinuity in the tangent plane.

Beyond addressing the above limitations, in future work we would like to explore the construction of a coarse-to-fine medial representation that represents the shape of an object at different levels of abstraction [31]. Such models facilitate efficient coarse-to-find matching strategies and offer a powerful mechanism for further hierarchical database organization. In fact, such models can be automatically learned from examples, e.g., [26,27].

References

1. Alt, H., Aichholzer, O., Rote, G.: Matching Shapes With a Reference Point. In: Proceedings of the Tenth Annual Symposiusm On Computational Geometry, pp. 85–92 (1994)

2. Amenta, N., Choi, S., Kolluri, R.: The power crust, unions of balls, and the medial axis transform. Comput. Geometry. Theory Appl. **19**(2), 127–153 (2001)

3. Ankerst, M., Kastenmüller, G., Kriegel, H., Seidl, T.: 3D Shape Histograms for Similarity Search and Classification in Spatial Databases. In: Advances in Spatial Databases, 6th International Symposium, vol. 18, pp. 700–711 (1999)

4. Besl, P., Jain, R.: Three-dimensional object recognition. ACM Comput. Surv. **17**(1), 75–145 (1985)

5. Biederman, I.: Recognition–By–Components: a theory of human image understanding. Psychol. Rev. **94**(2), 115–147 (1987)

6. Binford, T.O.: Visual Perception by Computer. In: IEEE Conference on Systems and Control (1971)

7. Blum, H.: Biological Shape and Visual Science. J. Theor. Biol. **38**, 205–287 (1973)

8. Borgefors, G.: Distance Transformations in Arbitrary Dimensions. Comput. Vis. Graph. Image Process. **27**, 321–345 (1984)

9. Borgefors, G., Nyström, I., Sanniti di Baja, G.: Computing Skeletons in Three Dimensions. Pattern Recogn. **32**(7), 1225–1236 (1999)

10. Campbell, R.J., Flynn, P.J.: A Survey of Free-Form Object Representation and Recognition Techniques. Comput. Vis. Image Understanding **81**, 166–210 (2001)

11. Chin, R., Dyer, C.: Model-Based Recognition in Robot Vision. ACM Comput. Surv. **18**(1), 67–108 (1986)

12. Cornea, N., Demirci, M., Silver, D., Shokoufandeh, A., Dickinson, S., Kantor, P.: 3D Object Retrieval using Many-to-many Matching of Curve Skeletons. In: Proceedings, IEEE International Conference on Shape Modeling and Applications, pp 368–373. Boston (2005)

13. Culver, T., Keyser, J., Manocha, D.: Exact Computation of the Medial Axis of a Polyhedron. Comput. Aided Geometric Des. **21**(1), 65–98 (2004)

14. Damon, J.N.: Geometry and medial structure. In: Siddiqi, K., Pizer, S.M. (eds.) Medial Representations: Mathematics, Algorithms and Applications. Springer, Heidelberg (2007) (to appear)

15. Dimitrov, P.: Flux Invariants For Shape. M.Sc. thesis, School of Computer Science, McGill University (2003)

16. Dimitrov, P., Damon, J.N., Siddiqi, K.: Flux Invariants for Shape. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Madison (2003)

17. van Eede, M., Macrini, D., Telea, A., Sminchisescu, C., Dickinson, S.: Canonical Skeletons for Shape Matching. In: Proceedings, International Conference on Pattern Recognition, Hong Kong (2006)

18. Elad, M., Tal, A., Ar, S.: Content Based Retrieval of VRML Objects- An Iterative and Interactive Approach. In: 6th Europgraphics Workshop on Multimedia, pp. 107–118. Manchester (2001)

19. Giblin, P.J., Kimia, B.B.: A formal Classification of 3D Medial Axis Points and Their Local Geometry. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 238–251 (2004)

20. Hamza, A.B., Krim, H.: Geodesic Object Representation and Recognition. In: Proceedings of DGCI, vol. LNCS 2886, pp. 378–387 (2003)

21. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. In: Proceedings of ACM SIGGRAPH, pp. 203–212 (2001)

22. Ikeuchi, K., Kanade, T.: Automatic Generation of Object Recognition Programs. Proc. IEEE **76**, 1016–1035 (1988)

23. Katayama, N., Satoh, S.: The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries. In: ACM SIGMOD International Conference on Management of Data, pp. 369–380 (1997)

24. Kazhdan, M., Chazelle, B., Dobkin, D., Funkhouser, T., Rusinkiewicz, S.: A Reflective Symmetry Descriptor for 3D Models. Algorithmica **38**(1), 201–225 (2003)

25. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In: Symposium on Geometry Processing (2003)

26. Keselman, Y., Dickinson, S.J.: Generic Model Abstraction from Examples. IEEE Trans. Pattern Anal. Machine Intell. **27**(7), 1141–1156 (2005)

27. Levinshtein, A., Sminchisescu, C., Dickinson, S.J.: Learning Hierarchical Shape Models from Examples. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds) EMMCVPR, Lecture Notes in Computer Science, vol. 3757, pp. 251–267. Springer, Heidelberg (2005)

28. Leymarie, F.F., Kimia, B.B.: Computation of the Shock Scaffold for Unorganized Point Clouds in 3D. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 821–827. Madison (2003)

29. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)

30. Malandain, G., Bertrand, G., Ayache, N.: Topological Segmentation of Discrete Surfaces. Int. J. Comput. Vis. **10**(2), 183–197 (1993)

31. Marr, D., Nishihara, K.H.: Representation and Recognition of the Spatial Organization of Three Dimensional Shapes. Proc. R. Soc. Lond. B **200**, 269–294 (1978)

32. Messmer, B., Bunke, H.: Subgraph Isomorphism in Polynomial Time. Tech. Rep. 95-003, IAM (1995). citeseer.nj.nec.com/messmer95subgraph.html

33. Murase, H., Nayar, S.: Visual Learning and Recognition of 3D Objects from Appearance. Int. J. Comput. Vis. **14**, 5–24 (1995)

34. Näf, M., Kübler, O., Kikinis, R., Shenton, M.E., Székely, G.: Characterization and Recognition of 3D Organ Shape in Medical Image Analysis Using Skeletonization. In: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (1996)

35. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape Distributions. ACM Trans. Graphics **21**(4), 807–832 (2002)

36. Pellilo, M., Siddiqi, K., Zucker, S.W.: Matching Hierarchical Structures Using Association Graphs. IEEE Trans. Pattern Anal. Mach. Intell. **21**(11), 1105–1120 (1999)

37. Pentland, A.: Perceptual Organization and the Representation of Natural Form. Artif. Intell. **28**, 293–331 (1986)

38. Pizer, S.M., Siddiqi, K., Székeley, G., Damon, J.N., Zucker, S.W.: Multiscale Medial Axes and Their Properties. Int. J. Comput. Vis. **55**(3), (2003)

39. Pudney, C.: Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images. Comput. Vis. Image Understanding **72**(3), 404–413 (1998)

40. Reyner, S.W.: An Analysis of a Good Algorithm for the Subtree Problem. SIAM J. Comput. **6**, 730–732 (1977)

41. Rosch, E.: Principles of Categorization. In: Cognition and Categorization. L. Erlbaum Associates (1978)

42. Sclaroff, S., Pentland, A.: Modal Matching for Correspondence and Recognition. IEEE Trans. Pattern Anal. Mach. Intell. **17**(6), 545–561 (1995)

43. Sebastian, T., Klein, P., Kimia, B.: Recognition of Shapes by Editing their Shock Graphs. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 551–571 (2004)

44. Sengupta, K., Boyer, K.: Organizing Large Structural Modelbases. IEEE Trans. Pattern Anal. Mach. Intell. **17**(4), 321–332 (1995)

45. Sengupta, K., Boyer, K.: Modelbase Partitioning Using Property Matrix Spectra. Comput. Vis. Image Understanding **70**(2), 177–196 (1998)

46. Shapiro, L.G., Haralick, R.M.: Organization of Relational Models for Scene Analysis. IEEE Trans. Pattern Anal. Mach. Intell. **4**(6), 595–602 (1982)

47. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton Shape Benchmark. In: Shape Modeling International. Genova, Italy (2004)

48. Shinagawa, Y., Kunii, T.L., Kergosien, Y.L.: Surface Coding Based on Morse Theory. IEEE Trans. Comput. Graphics Appl. **11**(5), 66–78 (1991)

49. Shokoufandeh, A., Dickinson, S.J., Jonsson, C., Bretzner, L., Lindeberg, T.: The Representation and Matching of Qualitative Shape at Multiple Scales. In: Proceedings, ECCV, pp. 759–775, Copenhagen (2002)

50. Shokoufandeh, A., Dickinson, S.J., Siddiqi, K., Zucker, S.W.: Indexing Using a Spectral Encoding of Topological Structure. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 491–497. Fort Collins, (1999)

51. Shokoufandeh, A., Macrini, D., Dickinson, S., Siddiqi, K., Zucker, S.: Indexing Hierarchical Structures Using Graph Spectra. IEEE Trans. Pattern Anal. Mach. Intell. **27**(7), 1125–1140 (2005)

52. Siddiqi, K., Bouix, S., Tannenbaum, A., Zucker, S.W.: Hamilton-Jacobi Skeletons. Int. J. Comput. Vis. **48**(3), 215–231 (2002)

53. Siddiqi, K., Pizer, S.M.: Medial Representations: Mathematics, Algorithms and Applications. Springer, Heidelberg (2007) (to appear)

54. Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., Zucker, S.W.: Shock Graphs and Shape Matching. Int. J. Comput. Vis. **35**(1), 13–32 (1999)

55. Sossa, H., Horaud, R.: Model Indexing: The Graph-Hashing Approach. In: Proceedings, IEEE CVPR, pp. 811–814 (1992)

56. Stolpner, S., Siddiqi, K.: Revealing Significant Medial Structure in Polyhedral Meshes. In: Third International Symposium on 3D Data Processing, Visualization and Transmission. Chapel Hill (2006)

57. Sundar, H., Silver, D., Gagvani, N., Dickinson, S.: Skeleton Based Shape Matching and Retrieval. In: International Conference On Shape Modeling International and Applications, pp. 130–142. Seoul (2003)

58. Turk, M., Pentland, A.: Eigenfaces for Recognition. J. Cogn. Neurosci. **3**(1), 71–86 (1991)

59. Vranic, D., Saupe, D.: 3D Model Retrieval With Spherical Harmonics and Moments. In: Proceedings of the DAGM, pp. 392–397 (2001)

60. Zhang, J., Kaplow, R., Chen, R., Siddiqi, K.: The McGill Shape Benchmark (2005). www.cim.mcgill.ca/~shape/benchMark/