3D Object Retrieval using Many-to-many Matching of Curve Skeletons

Nicu D. Cornea[1] M. Fatih Demirci[2] Deborah Silver[1] Ali Shokoufandeh[2] Sven J. Dickinson[3] Paul B. Kantor[1]
[1]Rutgers University
[2]Drexel University
[3]University of Toronto

Abstract

We present a 3D matching framework based on a many-to-many matching algorithm that works with skeletal representations of 3D volumetric objects. We demonstrate the performance of this approach on a large database of 3D objects containing more than 1000 exemplars. The method is especially suited to matching objects with distinct part structure and is invariant to part articulation. Skeletal matching has an intuitive quality that helps in defining the search and visualizing the results. In particular, the matching algorithm produces a direct correspondence between two skeletons and their parts, which can be used for registration and juxtaposition.

1. Introduction

3D object models are now widespread and are used in many diverse applications, such as computer graphics, scientific visualization, CAD, computer vision, medical imaging, etc. Large databases of 3D models are now publicly available, such as the Princeton Shape Benchmark Database [19] or the 3D Cafe repository [1], with datasets contributed by the CAD community, computer graphic artists or scientc visualization community. The problem of searching for a specific shape in a large database of 3D models is an important area of research. Text descriptors associated with the 3D shapes can be used to drive the search process, as is done for 2D images [9, 16]. However, text descriptions may not be available and furthermore, could not apply for part-matching or *similarity*-based matching.

Matching 3D objects is a difficult problem, with a complex relation to the 2D shape-matching problem. While the 3D nature of the representation helps remove some of the viewpoint, lighting, and occlusion problems in computer vision, other issues arise. Of course, the added dimension and the inherent increase in data size make the matching process more computationally expensive. Furthermore, many of the models are degenerate, containing holes, intersecting polygons, overly thin regions, etc. And there are many different types of matching that may be desirable. Given a query object, one may want to search an entire database for a matching exemplar, if one exists. On the other hand, if the database contains categorical models, one may want to find the category to which the query exemplar belongs.

In this paper, we use the *curve-skeleton* of a 3D shape to drive the matching process. The skeleton is a useful shape abstraction that captures the essential topology of an object in both two and three dimensions. It provides the following characteristics, not present in global shape descriptors:
Part/Component Matching: curve-skeletons incorporate the notion of parts or components and so they can accommodate part matching, where the object to be matched is part of a larger object, or vice versa. This feature can give the users more control over the matching algorithm, allowing them to specify what part of the object they would like to match or whether the matching algorithm should weight one part of the object more than another.
Registration and visualization: The skeleton can be used to register the two matched objects and visualize the result in a common space. This is very important in scientific applications where one is interested in both finding a similar object and understanding the extent of the similarity [21].
Intuitiveness: The skeleton is an intuitive representation of shape and can be easily understood by the user, providing more control in the matching process.
Articulated transformation invariance: The method presented here can be used for articulated object matching, because the skeleton topology does not change as a result of articulated motion. An example was shown in [21].

This work enhances the framework presented in [21] by using a new skeletonization algorithm and an extension of the many-to-many matching algorithm introduced in [12]. In [21], the idea of using skeletons for matching was described; however, only a small database was used for experimentation, and in general, the methodology was not scalable. In this paper, the entire methodology has been re-

Proceedings of the International Conference on Shape Modeling and Applications (SMI'05)
0-7695-2379-X/05 $20.00 © 2005 IEEE

vised by using a more robust skeletonization algorithm and a more robust matching algorithm. We demonstrate the efficacy of this new matching framework on the objects from the Princeton Shape Database [19]. Our matching algorithm is based on establishing correspondences among two skeletal representations via distribution-based matching in metric spaces. While the performance of our algorithm is comparable to that of other existing 3D matching methods (eg. [14, 19]), the locality of our skeletal representation and matching algorithm has some other benefits, such as allowing part matching, articulated matching, etc.

2. Previous Work

Work related to this paper include 3D matching, skeleton detection, graph matching and point-to-point matching.

A number of different approaches have been proposed for the matching problem. Using a simplified description of a 3D model, usually in lower dimensions (also known as a shape signature), reduces the 3D matching problem to comparing these different signatures. The dimensional reduction and the simple nature of these shape descriptors make them ideal for applications involving searching in large databases of 3D models. Osada et al. in [14] propose the use of a distribution, sampled from one of many shape functions, as the shape signature. Among the shape functions, the distance between two random points on the surface proved to be the most effective at retrieving similar shapes. In [19], a shape descriptor based on 2D views (images rendered from uniformly sampled positions on the viewing sphere), called the *Light Field Descriptor*, performed better than descriptors that use the 3D properties of the object. In [11], Kazhdan et al. propose a shape description based on a spherical harmonic representation.

Related to the work described here is the 1D skeleton determination (also known as the curve-skeleton). This is related to the medial surface. A curve-skeleton is used both for its intuitiveness and for its simplicity even though it is not unique. There are many skeleton extraction techniques: thinning-based methods attempt to iteratively peel layers off a voxelized representation of the object while attempting to maintain the topology [7]; geometric methods use Voronoi diagrams to determine a medial axis [3]; distance field methods use a distance function to classify and retain centered voxels [22]; while a continuous quench function computed from the object, identifies the extremes as medial axis points [4]. For a full description, please see [6].

Most of the previous work on point and skeleton matching has focused on solving one-to-one correspondence problems. Kim and Kak [13] used a combination of discrete relaxation and bipartite matching in model-based 3-D object recognition in computer vision. Pellilo et al. [15] devised a quadratic programming framework for the matching of 2D skeletons using a maximal clique formulation. Siddiqi et al. combined a bipartite matching framework with a spectral decomposition of graph structure to match shock graphs [20].

The problem of many-to-many matching has also been studied, most often in the context of edit-distance (see, e.g. [18]). In such a setting, one seeks a minimal set of relabelings, additions, deletions, merges, and splits of nodes and edges that transform one graph into another. In [12], we introduced a many-to-many matching algorithm and studied its utility for 2D skeletal representations. We used a specific measure, the Earth Mover's Distance, to compute distances between sets of weighted vectors. The matching algorithm in this paper is an extension of the one presented in [12].

3. Approach

Before matching, the curve-skeleton of each object must be computed (Section 3.1). Next, we match the exemplar skeleton against all other skeletons in the database (Section 3.2). Finally, we will rank the results and visualize the best match.

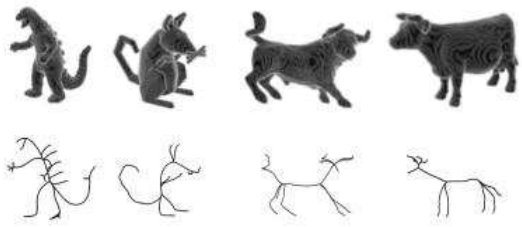

Figure 1. Some examples of 3D shapes and their computed skeletons.

3.1. The Curve-Skeleton

In this work, we utilize a curve skeleton for the matching. The curve skeleton is a concise intuitive representation of the object, used in many CAD and computer graphics modeling programs and consists of a set of connected 1D curves (1 voxel thick). Although it is not unique in 3D, it has a number of properties that are advantageous to matching as enumerated in the introduction: intuitiveness, part/component matching, registration and articulated transformation invariance.

Our curve-skeleton extraction algorithm works on a volumetric representation of the 3D object. It is based on the

method presented by Chuang et. al. [4] which uses a generalized potential field generated by charges placed on the surface of the object. The generalized potential at a point due to a nearby point charge is defined as a repulsive force, pushing the point away from the charge with a strength that is inversely proportional to some power of the distance between the point and the charge. This step produces a vector field.

Given a 3D vector field, we use concepts from vector field visualization to identify two types of *seed points* that we will used to construct a curve-skeleton: critical points and high divergence points. At critical points, the magnitude of the vector vanishes, which is why they are also called *zeros* of the vector field. A full discussion of the visualization of vector-field topology and the different types of critical points can be found in [8] and [10]. In addition to critical points, we also use the divergence of the vector field. The user specifies a percentage of the highest divergence value which will be used as a threshold to select new seed points. By varying this parameter, one can generate an entire hierarchy of skeletons of various complexities and select the best one for a given application. In the experiments presented in Section 4, we used a 40% divergence threshold for all our skeletons.

Skeleton segments are discovered using a *force-following* algorithm on the underlying vector field, starting at each of the identified seed points. The force following process evaluates the vector (force) value at the current point and moves in the direction of the vector with a small pre-defined step. At critical points, where the force vanishes, the initial directions are determined by evaluating the eigenvalues and eigenvectors of the Jacobian at the critical point. For more details on computing the curve-skeleton see [6]. Figure 1 shows a few examples of 3D objects and their respective skeletons.

The skeleton obtained using the above algorithm consists of a set of points sampled by the force following algorithm. Each skeleton point is then equipped with a distance-transform value [7], a real number specifying the distance to the closest point on the surface of the object. This additional information is used by the many-to-many matching process.

3.2. Many-to-many matching using EMD

To match two 3D skeletons, we use a distribution-based similarity measure, known as the Earth Mover's Distance (EMD) under transformation [5, 12]. The Earth Mover's Distance is designed to evaluate dissimilarity between two multi-dimensional distributions. The EMD approach assumes that a distance measure between single features, called the ground distance, is given. The EMD then "lifts" this distance from individual features to full distributions. The main advantage of using EMD lies in the fact that it subsumes many histogram distances and permits partial matches in a natural way. This important property allows the similarity measure to deal with uneven clusters and noisy datasets [12].

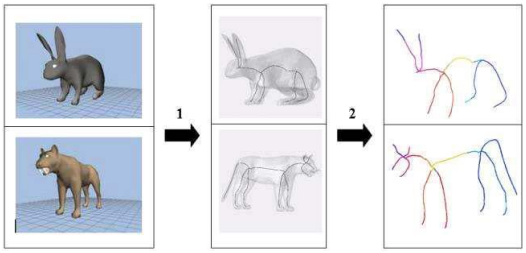

Figure 2. Computing similarity between two given objects. Step 1: compute skeletons. In Step 2 the correspondence is shown with color coded regions.

Computing the EMD is based on a solution to the well-known *transportation problem* [2], whose optimal value determines the minimum amount of "work" required to transform one distribution into the other. More formally, let $P = \{(p_1, w_{p_1}), \ldots, (p_m, w_{p_m})\}$ be the first distribution with m points, and let $Q = \{(q_1, w_{q_1}), \ldots, (q_n, w_{q_n})\}$ be the second distribution with n points. Let $D = [d_{ij}]$ be the ground distance matrix, where d_{ij} is the ground distance between points p_i and q_j. Our objective is to find a flow matrix $F = [f_{ij}]$, with f_{ij} being the flow between points p_i and q_j, that minimizes the overall cost $\text{Work}(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}$ subject to the following list of constraints:

$$
\begin{aligned}
& f_{ij} \geq 0,\ 1 \leq i \leq m,\ 1 \leq j \leq n \\
& \textstyle\sum_{j=1}^{n} f_{ij} \leq w_{p_i},\ 1 \leq i \leq m \\
& \textstyle\sum_{i=1}^{m} f_{ij} \leq w_{q_j},\ 1 \leq j \leq n \\
& \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}\right)
\end{aligned}
$$

The optimal value of the objective function $\text{Work}(P, Q, F)$ defines the Earth Mover's Distance between the two distributions.

An extension of the original EMD approach [17] allows us to match point sets that are "non-rigidly" embedded into the Euclidean space by allowing sets to undergo transformations. Assuming that a transformation is applied to the second distribution, distances d_{ij}^T are defined as $d_{ij}^T = d(p_i, T(q_j))$ and the objective function becomes $\text{Work}(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}^T$. The minimal

value of the objective function Work(P, Q, F, T) defines the Earth Mover's Distance between the two distributions that are allowed to undergo a transformation.

An iterative process (which called **FT**, short for "an optimal **F**low and an optimal **T**ransformation"), that achieves a local minimum of the objective function was suggested in [17]. Starting with an initial transformation $T^{(0)} \in \mathcal{T}$ from a given $T^{(k)} \in \mathcal{T}$, they compute the optimal flow $F = F^{(k)}$ that minimizes the objective function Work$(P, T^{(k)}(Q), F)$, and from a given optimal flow $F^{(k)}$ they compute an optimal transformation $T = T^{(k+1)} \in \mathcal{T}$ that minimizes the objective function Work$(P, T(Q), F^{(k)})$. The iterative process stops when the improvement in the objective function value falls below a threshold. The resulting optimal pair (F, T) depends on the initial transformation $T^{(0)}$.

We will use the Earth Mover's Distance under transformation between two skeletons as a measure of their similarity: a small value of this distance indicating high similarity between the two skeletons. Figure 6 shows the distances between the query and the top matched objects.

Figure 2 shows an example of matching between two objects: in step 1, the curve-skeleton for each object is computed while in step 2, the many-to-many matching establishes the distance and the correspondence between the two skeletal representations. The skeleton regions that were matched to each other are shown in the same color in Figure 2.

4. Experiments

To evaluate the utility of our skeletal representation and many-to-many matching algorithm, we performed 2 sets of experiments: 3D base classification and part matching.

We first tested our proposed approach to retrieving similar objects on a subset of 1081 objects from the Princeton Shape Benchmark Database [16], grouped into 99 non-empty classes from both the test and training classifications [19] (the 3D base classification task). In our experiments, we first created 3D skeletons for each object. We used a divergence threshold of 40% for all our skeletons. We then computed the distance from each object to the remaining database entries using our many-to-many matching algorithm. If the conceptual classes correspond to bodies which vary only in scale, or by articulated transformation, our algorithm should return an object that belongs to the same class as the query. We will classify this as a "correct matching". Based on the overall matching statistics, we observe that in 71.1% of the experiments, the overall *best match* selected by our algorithm belonged to the same class as the query (also known as the nearest neighbor criterion [19]). In 74.3% of the experiments, the best match belonged to the same parent class as that of query.

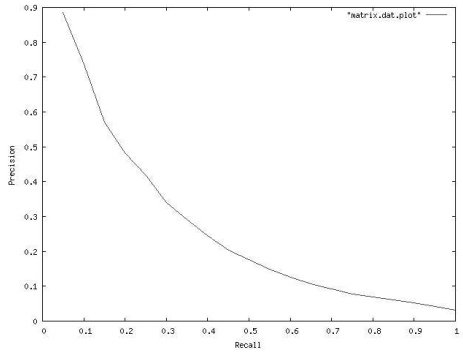

Figure 3. Precision/Recall for many-to-many matching algorithm in object retrieval experiment.

In a second experiment, we asked how many of the models in the query's class appear within the top $T-1$ matches, where T is the size of the query's class (*first tier* [19]). This number was 17.2%. Repeating the same experiment, but considering the top $2 \times T - 1$ matches (*second tier* [19]) covers 22.7% of the members of the class.

Comparing these results with those reported by Shilane et. al [19] in Table 4 of their work, it should be noted that our method outperforms all methods on the nearest neighbor criterion, but does not do as well on the first and second tier criteria. The sharp drop in the precision-recall curve in Figure 3 illustrates the loss of precision for the first and second tier criteria. The precision-recall plot shows the relation between recall (the ratio of models from the class of the query returned within the top N matches) and the precision (the ratio of the top N matches that belong to the query class) [19]. Figure 3 shows the precision-recall plot averaged over all models and looking at the first 20 best matches only.

In Figure 4, we have presented the matching results for a small subset of objects. The first column of each row shows the query object; the remaining elements of each row represent the top 10 closest objects of the database determined by our matching algorithm. These are all instances where the closest object is an object from a similar class. In some cases, while the algorithm has identified an object with similar structure as best match, it was still penalized for selecting an object from an incorrect category. The query object (race car) in row one and its best matched object are an example of such a case. They can be attributed to the particular hierarchy of categories used by the Princeton Shape Benchmark Database [16]. When similarity of shape is desired, a method which relies on shape would help retrieve objects not normally associated with the exemplar and not typically categorized with it.

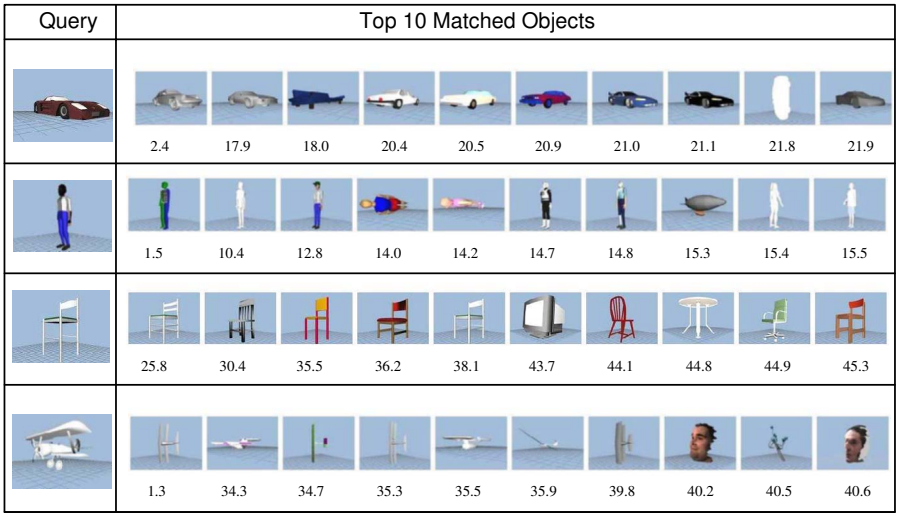

Figure 4. Models are sorted by the similarity to the query object.

4.1. Visualizing the Similarity

An important aspect of the matching is the computation of correspondence between the matched objects. Our many-to-many matching algorithm provides a direct correspondence between the skeleton points of the two matched objects. This allows one to register the query object to the database objects and aid in visualization and a better understanding of the match. The correspondences between the matched objects are illustrated with color coded regions in Figures 2 and 5. Global shape descriptors perform poorly at this task because global information cannot preserve local correspondences.

4.2. Part Matching

Matching of a part within a complex whole is useful for CAD-type databases and also for recognition in laser-scanned images, which tend to cluster objects together. It is presumably also central to medical applications in which a particular configuration is to be found somewhere in a larger object. Specifically, given a part of an object as a query, one attempts to locate objects containing similar subparts. Here, the difficulty lies in the fact that none of the database objects contains an exact copy of the query.

In our next experiment, we used a query part (a torso) and matched it against several *simple* and *composite* objects in the database, some containing the query part. The composite objects were obtained by a union operation applied to two simple objects – the kind of composition one would expect to encounter in laser-scanned scenes. Figure 5 shows the query object (and the corresponding skeleton) in (a) and some of the objects that was matched against in (b). In (c) we show the computed correspondence between the query skeleton and the skeletons of the objects in (b). Points that match the query skeleton are shown in red.

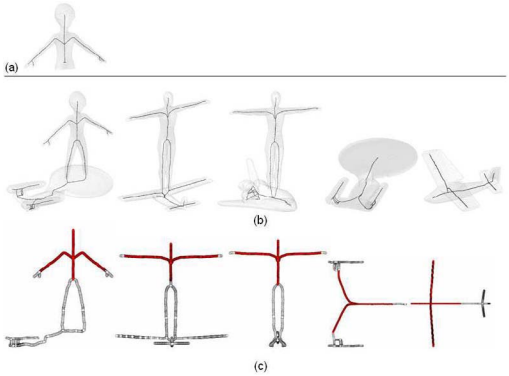

Figure 5. Part Matching Example: the query object in (a) is matched against each of the objects in (b). The correspondences between their skeletons are shown in red in (c).

5. Discussion

Compared to our previous skeletonization method described in [21], the potential field-based skeletonization algorithm is more robust to reasonable amounts of noise on the boundary of the object (see [6]). However, the curve-skeleton may not be the best abstraction for all types of objects. For example thin disk-like objects or a mushroom shape are not well represented by a curve-skeleton. The

EMD-based matching compensates for a certain amount of differences in skeletonization by looking at loose relationships among the skeleton points.

6. Conclusions and Future Work

In this paper, we have presented a matching framework that is an extension to our many-to-many matching algorithm in [12]. In addition to new skeletonization and matching approaches, we have demonstrated the performance of the method on a database of over 1000 objects, with retrieval results comparable to the global shape descriptor methods presented in [19].

The skeleton-based approach has a number of advantages over the global shape descriptor methods. It is an intuitive representation of 3D objects that can be easily used to understand the similarities present in the matched objects. Since our many-to-many matching algorithm provides a direct correspondence between skeleton points in two matched objects, one can use this correspondence for registration and juxtaposition. The skeleton captures both global and local properties of the shape, so it can be used for many different matching tasks. This is demonstrated in part matching, where only a portion of the skeleton is matched. Part matching is useful in CAD environments or segmentation of laser-scanned scenes.

As future work, we plan on improving the skeletonization code further so that it can correctly handle objects with thin regions. Our part matching examples have shown many-to-many matching can be used to locate a part in a database of composite objects. The inverse problem is also of interest, where given a composite object, one would like to identify its component parts among the objects of a database.

7. Acknowledgements

This work is supported in part by ITR NSF [EIA-0205178].

References

[1] 3D Cafe. http://www.3dcafe.com/asp/freestuff.asp.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*, pages 4–7. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[3] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *6th ACM Symposium on Solid Modeling*, pages 249–260, 2001.

[4] J. Chuang, C. Tsai, and M. K. Skeletonization of three-dimensional object using generalized potential field. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1241–1251, 2000.

[5] S. D. Cohen and L. J. Guibas. The earth mover's distance under transformation sets. In *Proceedings, 7th International Conference on Computer Vision*, pages 1076–1083, Kerkyra, Greece, 1999.

[6] N. Cornea, D. Silver, X. Yuan, and R. Balasubramanian. Computing hierarchical curve-skeletons of 3d objects. *CAIP Technical Report CAIP-TR277 (http://www.caip.rutgers.edu/∼cornea/Skeletonization)*, Nov 2004 (Submitted for publication).

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

[8] A. Globus, Levit, and T. Lasinski. Tool for visualizing the topology of three-dimensional vector fields. In *IEEE Visualization*, pages 33–40, 1991.

[9] Google Image Search. http://www.google.com.

[10] J. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.

[11] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Symposium on Geometry Processing*, pages 167–175, 2003.

[12] Y. Keselman, A. Shokoufandeh, M. Demirci, and S. Dickinson. Many-to-many graph matching via metric embedding. In *CVPR*, pages I–850– I–857 vol.1, June 2003.

[13] W. Kim and A. C. Kak. 3D object recognition using bipartite matching embedded in discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):224–251, 1991.

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

[15] M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, November 1999.

[16] Princeton Shape Retrieval and Analysis, 3D Model Search. http://shape.cs.princeton.edu/search.html.

[17] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *IJCV*, 40(2):99–121, 2000.

[18] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing shock graphs. In *IEEE International Conference on Computer Vision*, pages 755–762, 2001.

[19] P. Shilane, M. K. P. Min, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling International*, Genoa, Italy, June 2004.

[20] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30:1–24, 1999.

[21] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Shape Modelling and Applications Conference, SMI 2003*, Seoul, Korea, May 2003.

[22] Y. Zhou, A. Kaufman, and A. Toga. Three-dimensional skeleton and centerline generation based on an approximate minimum distance field. *The Visual Computer*, 14:303–314, 1998.