Canonical Skeletons for Shape Matching *

M. van Eede
University of Toronto

D. Macrini
University of Toronto

A. Telea
Technical University Eindhoven

C. Sminchisescu
University of Toronto

S. Dickinson
University of Toronto

Abstract

Skeletal representations of 2-D shape, including shock graphs, have become increasingly popular for shape matching and object recognition. However, it is well known that skeletal structure can be unstable under minor boundary deformation, part articulation, and minor shape deformation (due to, for example, small changes in viewpoint). As a result, two very similar shapes may yield two significantly different skeletal representations which, in turn, will induce a large matching distance. Such instability occurs both at external branches as well as internal branches of the skeleton. We present a framework for the structural simplification of a shape's skeleton which balances, in an optimization framework, the desire to reduce a skeleton's complexity by minimizing the number of branches, with the desire to maximize the skeleton's ability to accurately reconstruct the original shape. This optimization yields a canonical skeleton whose increased stability yields significantly improved recognition performance.

1. Introduction

The skeleton of a shape aims to capture its part structure. One of the first formal skeleton definitions is that of Blum [3], who defined the medial axis of a shape as the loci of centers of the maximal circles inscribed in the shape. Medial axes are created from a shape using the Medial Axis Transform (MAT), while a related skeleton definition uses the local maxima, also called creases or ridges, of a shape's distance transform (DT) [3]. The original shape can be reconstructed from the skeleton points $s \in S$ using the MAT's radius function, $R(s)$, or distance transform value $DT(s)$, which labels every skeleton point s with the radius of the maximal inscribed circle centered at s, i.e., the minimal distance from s to the boundary.

*The authors would like to thank Kaleem Siddiqi for his valuable feedback on an earlier draft of this paper.

Early algorithms used to compute the discrete, pixel-sampled MAT tended to create skeletons that were extremely sensitive to boundary noise, spatial sampling rate, and small perturbations of the shape boundary. Several researchers, including Blum himself, proposed solutions to robustly computing the MAT. The generalized MAT [3] considers only skeleton points s with radius $R(s) > r_0$ greater than some threshold r_0; however, this may result in a disconnected skeleton. Other approaches use branch pruning and multiscale representations [8, 10, 12, 11, 5, 20]. Branches created by spatially small boundary details can be pruned using a collapsed boundary length metric [12, 11, 5, 20]. However, this works only for the so-called external branches. A branch is external if it has exactly one terminal endpoint in the skeleton tree, and internal otherwise. Overall, most methods encounter problems in eliminating spurious internal branches while retaining important descriptive branches, as illustrated in Fig. 1(a,b), where random "bumps" and "notches" are added to a hand shape.

A second source of skeletal instability has been studied by August *et al.* [1], who have shown that shape boundary concavities produce so-called *ligature* branch segments whose points are related *only* to the concave boundary points; when a ligature segment spans the entire branch, it is called a ligature branch [3, 1]. Small positional changes of such concavities can cause significant ligature branch structural changes (see Fig. 1(c,d)), which ultimately give rise to significant differences in their corresponding shock graphs [16]. August *et al.* [2] show that the internal skeleton instabilities cannot be removed by boundary smoothing alone. August *et al.* [1] attempt to deal with this problem by eliminating those shock graph nodes that represent ligature branches, in order to produce more stable graphs. However, this requires the robust detection of concave corners, which is a challenging problem for discrete images.

Giblin and Kimia [7] have catalogued all the generic transitions of the medial axis and showed that the above two types of MAT instabilities are the only cases where small boundary changes produce large representation (skeleton)

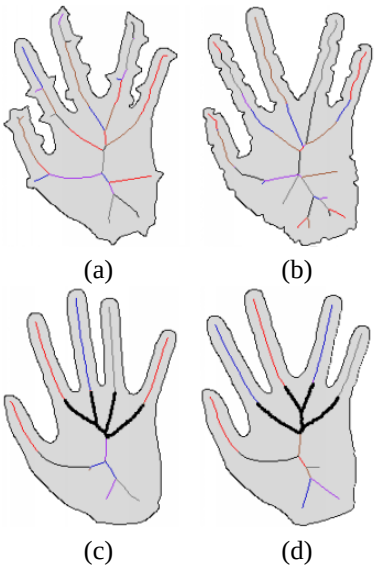


Figure 1. The Instability of a Shape's Skeleton. Considering only skeleton points with radius greater than some threshold r does not eliminate all spurious branches in the presence of bumps (a) or notches (b). Minor deformations in shape due to viewpoint change or articulation (c) and (d) may result in major changes in the topology of ligature segments (darkened) which, if spanning an entire branch, are called ligature branches. Changes in branch color reflect qualitative changes in the branch's radius function.

changes.[1] Such instability poses a major obstacle to effective object recognition, in general, and generic object recognition, in particular, where representational invariance to part articulation, minor shape deformation, and minor changes in viewpoint is essential. If such changes in an object's shape induce major changes in its underlying skeletal (or shock) graph structure, the distance between two graphs (as computed by a skeletal graph matcher, such as [17, 13]) will not reflect the distance between the two shapes. Our challenge is to eliminate these two types of instability by structurally simplifying a shape's skeleton, so that non-salient branches, both internal and external, are removed, leaving a canonical skeleton that captures only the salient part structure of the shape.

In this paper, we introduce an optimization framework for structural simplification that balances, on one hand, our desire to abstract or simplify a shape's skeletal representation (for fewer nodes in its underlying graph will lead to less complex graph matching) with, on the other hand, our desire to yield a representation that is true to the original shape, i.e., a skeleton whose reconstruction error is minimized. This trade-off between abstraction (complexity) and faithfulness (reconstruction error) is task dependent, and different recognition domains may weight these two goals differently. Still, they provide a pair of opposing forces which will help us converge on a canonical skeleton for shape matching. We describe our framework and demonstrate its efficacy on the domain of shock graph-based object recognition, showing how our structural simplification applied to both database and query shapes can yield a significant improvement in recognition performance.

[1] In fact, these two forms of instability are characterized by transitions of the symmetry set [7]. Another form of instability addresses the movement of a branch point as a function of boundary deformation (see, for example, Bouix *et al.* [4]). We do not address this form of instability since its affect on skeletal structure is minimal.

2. Structural Simplification

Our structural simplification procedure is divided into two stages, both of which balance reconstruction error with branch complexity. The first stage removes unstable external branches, while the second stage removes unstable internal branches. Removing external branches first is motivated by the fact that an external branch may separate two internal branches that will merge naturally after removing the external branch, thereby simplifying internal structure. Removing internal branches involves first identifying the ligature branches, which represent candidates for removal. Removing a candidate ligature branch requires the modification of neighboring branches subject to all of them obeying the properties of a MAT as well as minimizing reconstruction error. This latter problem was considered by Telea *et al.* [19], who proposed a principled framework that generates a simplified, abstracted skeleton hierarchy by analyzing the quasi-stable points of a Bayesian-inspired energy function. However, their method does not generate a true axis of symmetry, in the sense of the MAT definition, and thus cannot be directly used with existing skeleton-based shape matching techniques, such as [17, 13].

2.1. Candidate Branches for Pruning

Before applying our two-stage optimization procedure for branch pruning, we must identify candidate external and internal branches for pruning, and rank-order them by increasing saliency. The saliency of a branch, both internal and external, is related to the reconstruction error (detailed later in this section) induced by the skeleton minus the branch; branches that contribute less to the shape, and whose removal yields small reconstruction error, will be ranked before branches that contribute more to the shape, and whose removal yields large reconstruction error. In the

case of external branches, *all* external branches are considered as candidates for pruning. However, in the case of internal branches, only ligature branches are candidates for pruning and must therefore first be identified.

To identify the ligature branches, we will analyze the radius function of each internal branch. Specifically, let a skeleton S be a discrete, connected set of points in $\mathbb{N}^2$, and let the local neighborhood of a discrete point be its 8-neighborhood. We will approximate the radii of a branch's skeleton points as a function of the cumulative piecewise linear distance, d_i, along the branch $\{s_i\}$ with endpoints s_0 and s_n, where $s_i = [x_i, y_i]$, for $0 \leq i \leq n$. This distance is given by $d_i = \sum_{k=0}^{i-1} ||s_{k+1} - s_k||_2$, and the radius $\hat{R}(d_i)$ of the skeleton point s_i at distance d_i from s_0 is equal to $R(s_i)$.

We consider a least-squares fitting error for each line segment. Since we do not expect outliers, an unweighted least-squares method provides a good approximation. To compute the $n+1$ indices of endpoints for n line segments that minimize the fitting error, we define the following function:

$$\hat{E}(n,i,k) = \begin{cases} \text{LSF}(i,k) & \text{if } n = 1, \\ \min\limits_{i<j<k} \left\{ \hat{E}(\lfloor n/2 \rfloor, i, j) + \hat{E}(\lceil n/2 \rceil, j, k) \right\} & \text{otherwise;} \end{cases} \quad (1)$$

where $\text{LSF}(i,k)$ is the line and its associated error that best fits, in the least-squares sense, the data between endpoints indexed by i and k, i.e., $\text{LSF}(i,k) = e(m_{ik}, b_{ik})$, for $e(m,b) = \sum_{j=i}^{k} (R(d_j) - (md_j + b))^2$ and $(m_{ik}, b_{ik}) = \underset{m,b \in \mathbb{R}}{\text{argmin}} \{e(m,b)\}$. In turn, $\hat{E}(n,i,k)$ is the minimum error that can be achieved when fitting points i to k with n segments. Note that the segments are constrained to be continuous on s but not on $R(s)$.

We implement the function $\hat{E}(n,i,k)$ using dynamic programming and use it to find the smallest value of n whose minimum error is smaller than half the number of skeleton points in a branch. This piecewise linear representation of the radius function of a skeleton branch allows us to identify the ligature segments within a branch. Since ligature segments are associated with concave boundary corners, we know that they must start at a branch junction point, have decreasing radii, and end at the first abrupt change in the slope of $R(s)$ (see Fig. 2).

The identification of ligature segments will allow us to reconnect a skeleton when removing internal branches without significantly affecting the original shape boundary. We locate the endpoints of a ligature segment within a branch by detecting significant "accelerations" in the branch's radius function, i.e., differences between the slopes of two adjacent line segments that exceed a threshold. Let m_0 and m_1 be the slopes of adjacent line segments with equal sign. We

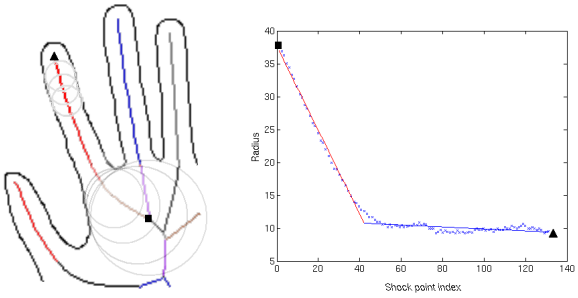


Figure 2. Approximating a Branch's Radius Function for Ligature Segment Identification. (a) The radii of maximally inscribed circles rapidly decrease as we move toward the concave corner between the fingers. (b) We compute a piecewise linear approximation to the radius function.

group together the points associated with these segments if

$$\frac{|m_0 - m_1|}{\max(|m_0|, |m_1|)} \leq \tau_l,$$

where τ_l is the ligature segment threshold. Note that $\max(|m_0|, |m_1|) > 0$ because in this step we have only decreasing branches. Hence, our ligature detection does not depend on the precise detection of local concave boundary corners, but rather on a more robust, global measure of relative slope change. We do not simply remove ligature branches, as in [1], but rather mark them as potential removals during our optimization procedure that balances reconstruction error with branch complexity.

When removing a ligature branch, we re-attach the skeleton branches that were connected to it in order to preserve skeleton connectedness, as illustrated in Fig. 3. Consider the removal of the small ligature branch in Fig. 3(left) below the junction of the index and middle fingers. If we were to deepen the concavity between the index and middle finger, as shown in Fig. 3(middle), the small target ligature branch would effectively disappear. Our strategy, therefore, will be to approximate this deepening of the concavity by modifying the branches adjacent to the ligature branch. To do this, we will alter only those branches attached to the smaller (in terms of radius) end of the ligature branch; in Fig. 3, this corresponds to the endpoint that leads to the ring finger. The target ligature branch will be removed, and the adjoining branches will be modified to connect to the larger end of the removed ligature branch.

The branches to be modified may consist of both non-ligature and ligature segments, as shown in Fig. 3(left), where the index finger consists of a non-ligature segment (red) at its extremity and a ligature segment (brown) attached to the ligature branch to be removed. The first step in

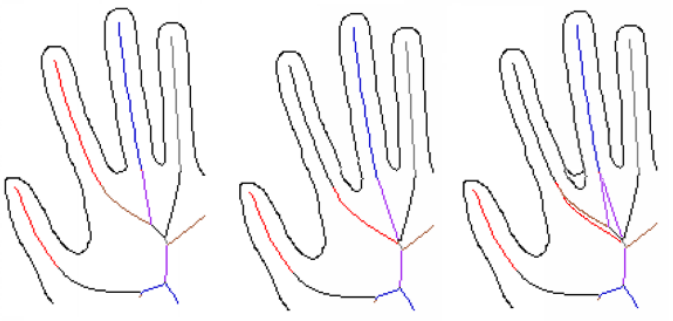


Figure 3. Removing ligature branches: (left) original skeleton in preparation for removal of target ligature branch below index and middle fingers; (middle) if we deepen the concavity between the two fingers slightly, we effectively end up with a skeleton without the target ligature branch; (right) overlay of the left and middle figures, motivating our approximation method that replaces the adjoining ligature segments and the target ligature branch with straight-line approximations.

the adjoining branch modification is to replace the adjoining ligature segments with linear approximations from their smaller endpoints to the larger endpoint of the removed ligature branch.[2] This effectively bridges the gap left by the removed ligature branch. However, we must still assign correct radius values to our skeleton approximation.

Our new approximations to the two adjoining ligature segments of our target ligature branch may effectively deepen the concavity between the two fingers, and as a result, part of the ligature approximation may become non-ligature. For this portion of the ligature approximation, the radius values will be assigned based on a linear extrapolation of the adjoining non-ligature segment's radius function. For the portion that remains a ligature segment, a linear interpolation between the two endpoints is used to assign the remaining radius values.

2.2. Pruning as Optimization

As mentioned earlier, our simplified skeleton balances reconstruction error with shape (branch) complexity, and in our branch selection process, both candidate internal and external branches for pruning are rank-ordered by increasing reconstruction error. The reconstruction error is area-based, and therefore reflects the area difference between the reconstructed shape from the skeleton with the branch and the reconstructed shape minus the branch. However, a simple area difference will fail to capture salient shape differences due, for example, to the removal of a long, thin part (e.g., the leg of a giraffe) whose area relative to the entire object is small, but whose contribution to salient part structure is large.

To account for part structure, similar to Styner *et al.* [18], we weight each pixel's contribution to the area by its normalized distance transform, in which the skeleton receives value 1 and the boundary receives value 0. In this manner, the skeleton of a long, thin part is weighted the same as that of a long, thick part, as are their respective boundaries. However, the larger area of the thick part will result in a larger integration of normalized distance transform values, and hence a larger reconstruction error. In this way, we can balance salient part structure with part mass, yielding an effective reconstruction error.

Specifically, for each shape point p, we associate the closest skeleton point $s^p \in S$:

$$s^p = \min_{s \in S} ||s - p|| \tag{2}$$

The reconstruction error of a point p, $E(p)$, is now given by:

$$E(p) = 1 - \frac{||s^p - p|| - R(s^p)}{R(s^p)} \tag{3}$$

where $R(s^p)$ is the radius of s^p. The reconstruction error $R(S)$ for a shape S with respect to the original shape S_O is:

$$R(S) = \frac{\sum_{p \in S_O - S} E(p)}{\sum_{p \in S_O}} \tag{4}$$

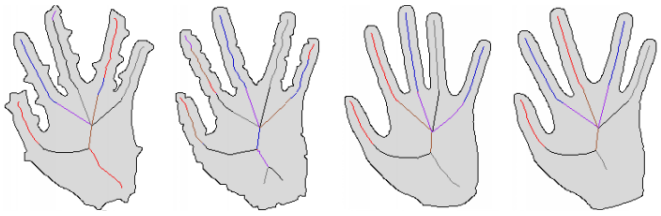
The cost function $C(S)$ for a skeleton S with branch complexity $B(S)$ and reconstruction error $R(S)$ has the form:

$$C(S) = B(S) + \omega R(S), \tag{5}$$

where $B(S)$ is simply the number of branches (nodes) of skeleton (graph) S. The ω we used ranges from 10,000 to 350. A high ω strongly penalizes the reconstruction error which, in practice, yields no simplification at all. Decreasing ω puts less emphasis on exact reconstruction and favors skeletons with lower branch complexity, obtained by removing less salient external branches and internal ligature branches.

Our optimization is two-pass. First, we optimize our cost function by considering only candidate external branch removal. Next, we fix the remaining external branches, and optimize in a second pass by considering only candidate internal branches. Fig. 4 shows the results of simplifying the four skeletons shown in Fig. 1, with $\omega = 600$. Note that the

[2] Ligature segments correspond to maximal circles that share two (in the case of full ligature) boundary concavity points) or one (in the case of semi-ligature) boundary point [1]. These constraints lead to ligature segments with low curvature, facilitating our straight-line approximation. This approximation can be improved by considering the constraints on the gradient of radius values of skeleton points, as defined by Damon [6].



(a) (b) (c) (d)

Figure 4. Structural simplification applied to the objects in Figure 1. Whereas articulation and noise led to four different skeletal topologies in Figure 1, their structural simplifications are almost identical.

four divergent skeletons in Fig. 1 have converged toward a canonical skeleton structure that is invariant to noise and articulation.

3. Experiments

We evaluate our framework for the tasks of object recognition and pose estimation. Since we are interested in obtaining stable skeletons for improved object recognition, we will adopt the shock graph as an abstracted skeletal representation, allowing us to utilize a powerful shock graph matcher [9]. A shock graph [16] is based on a coloring of the skeleton points (shocks) according to their radius function. The taxonomy consists of four distinct types: the radius function along the medial axis varies monotonically at a 1, achieves a strict local minimum at a 2, is constant at a 3 and achieves a strict local maximum at a 4. Once a skeleton's points are labeled according to this taxonomy, a partitioning of the points is performed. Each resulting segment becomes a node in the shape's shock graph, with directed edges joining adjacent skeleton segments and directed from larger to smaller (in terms of radii) nodes. Details and examples can be found in [15].

Our view-based 3-D object database consists of 120 views (8 objects at 15 views each). A recognition trial consists of removing a view from the database (without replacement), indexing into the database to retrieve a small subset of candidates (using the indexing framework described in [14]), matching the query to each candidate to yield a distance, rank-ordering the candidates by increasing distance (decreasing similarity), and choosing the closest candidate. If the parent object of the closest candidate is the same as the parent of the query, recognition is successful. If recognition is successful *and* one of the neighboring views of the query (on the viewsphere of the object) has high rank (in the rank-ordered list of candidates), then pose estimation is also successful. The small changes in viewpoint between samples on the viewsphere can introduce significant changes in shock graph structure. Hence, we expect our structural simplification to reduce the structural changes between neighboring views, leading to improved recognition and pose estimation performance. We expect similar improvement for the case of articulated objects, but leave such experiments as future work.

The results for both object recognition and pose estimation are shown in Figure 5. Each figure shows recognition performance (% trials correct) as a function of the weighting parameter ω, which varies from 10,000 (no smoothing) to 350 (maximum smoothing). The optimum recognition performance was achieved with $\omega = 600$. As can be seen from the two plots, structural simplification results in a 4% improvement in recognition performance and a 12% improvement in pose estimation performance. Note that overly large values of ω (very mild smoothing) can lead to structural inconsistencies across the views of an object, resulting in a dip in recognition performance.

A more dramatic improvement is found with perturbed (noisy) queries, in which between 3 and 40 small circular or triangular bumps and/or notches are randomly added to the boundary of the query shape, resulting in the introduction of structural instabilities. Figure 6 illustrates both recognition and pose estimation results with (blue) and without (red) structural simplification; in each case, we have chosen $\omega = 600$, the optimum value for unperturbed queries. The performance varies as a function of the nature of the perturbation, i.e., whether the perturbation contains notches, bumps, or both, as well as the magnitude of the perturbation given by the radius of the bump or notch (which varies from 3 to 6 pixels). As can be seen from the two plots, structural simplification results in up to 16% improvement for recognition and up to 20% improvement for pose estimation. These results clearly indicate the ability of our framework to simplify a noisy, unstable skeleton representation to yield an invariant, canonical skeletal representation.

4. Conclusions

Skeletal descriptions of a shape offer a powerful shape representation for object recognition, yet their structural instability has long been an obstacle to their widespread use. Our structural simplification framework isolates this instability at both external and internal branches, and removes non-salient branches. The removal of internal branches requires a proper smoothing of neighboring branches so that the resulting skeleton is a MAT and reconstruction error is minimized. The pruning of branches is formulated as an optimization process which balances branch complexity with reconstruction error. Results on a shock graph recognition experiment indicate a significant improvement in recognition and pose estimation performance when both query and

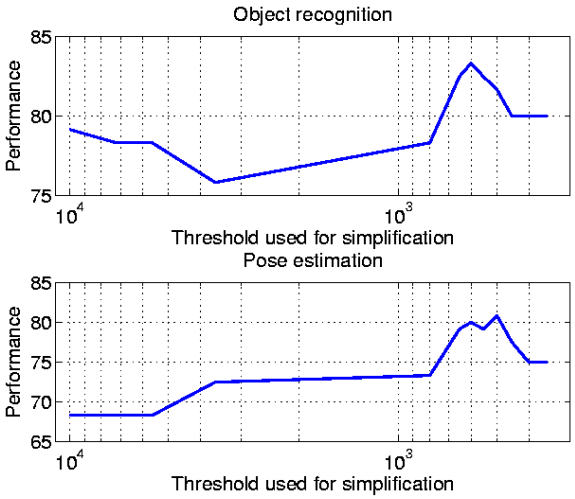


Figure 5. Results on unperturbed queries showing improved performance due to structural simplification for object recognition (top) and for pose estimation (bottom). The far left of each plot represents no structural smoothing (cost function reflects only reconstruction error), while the far right represents large structural smoothing (cost function is dominated by branch reduction). See text for discussion.

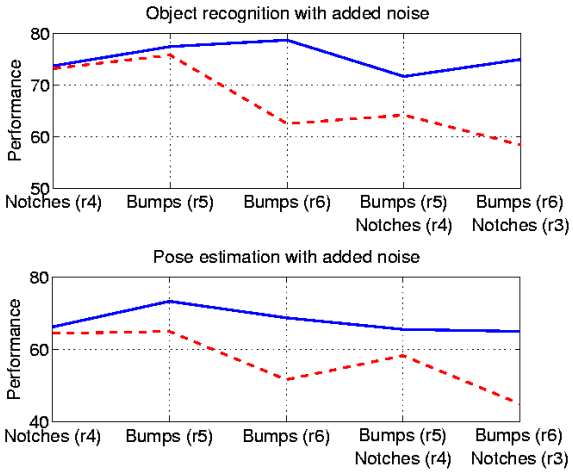


Figure 6. Results on perturbed queries showing improved performance due to structural simplification for object recognition (top) and for pose estimation (bottom). The red curves represent no simplification, while the blue curves represent simplification. See text for discussion.

database are structurally simplified prior to recognition.

References

[1] J. August, K. Siddiqi, and S. Zucker. Ligature instabilities in the perceptual organization of shape. *CVIU*, 76(3):231–243, 1999.

[2] J. August, A. Tannenbaum, and S. Zucker. On the evolution of the skeleton. In *ICCV(1)*, pages 315–322, 1999.

[3] H. Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38:205–287, 1973.

[4] S. Bouix, K. Siddiqi, A. Tannenbaum, and S. Zucker. Medial axis computation and evolution. In H. Krim and J. A. Yezzi, editors, *Statistics and Analysis of Shapes*. Springer, 2006.

[5] L. Costa and R. Cesar. *Shape analysis and classification*. CRC Press, 2001.

[6] J. Damon. Determining the geometry of boundaries of objects from medial data. *IJCV*, 63(1):45–64, 2005.

[7] P. J. Giblin and B. B. Kimia. On the local form and transitions of symmetry sets, medial axes, and shocks. *IJCV*, 54(1-3):143–156, 2003.

[8] J. A. Goldak, X. Yu, and A. K. L. Dong. Constructing discrete medial axis of 3-d objects. In *JCGA*, volume 3, pages 327–339, 1991.

[9] D. Macrini. Indexing and matching for view-based 3-d object recognition using shock graphs. Master's thesis, University of Toronto, 2003.

[10] R. Ogniewicz. Skeleton-space: A multiscale shape description combining region and boundary information. In *CVPR*, pages 746–751, Seattle, WA, June 1994.

[11] R. Ogniewicz. Automatic medial axis pruning by mapping characteristics of boundaries evolving under the euclidean geometric heat flow onto voronoi skeletons. Technical Report 95-4, Harvard Robotics Laboratory, 1995.

[12] R. L. Ogniewicz and O. Kubler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28:343–359, 1995.

[13] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing shock graphs. *PAMI*, 26(5):550–571, May 2004.

[14] A. Shokoufandeh, D. Macrini, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing hierarchical structures using graph spectra. *PAMI*, 27(7), July 2005.

[15] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. Zucker. The hamilton-jacobi skeleton. In *ICCV*, pages 828–834, 1999.

[16] K. Siddiqi and B. B. Kimia. A shock grammar for recognition. In *CVPR*, San Francisco, USA, 1996.

[17] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *IJCV*, 35(1):13–32, 1999.

[18] M. Styner, G. Gerig, S. Joshi, and S. Pizer. Automatic and robust computation of 3d medial models incorporating object variability. *IJCV*, 55(2-3):107–122, 2003.

[19] A. Telea, C. Sminchisescu, and S. Dickinson. Optimal inference for hierarchical skeleton abstraction. In *ICPR*, pages 19–22, Cambridge, U.K., August 2004.

[20] A. Telea and J. J. van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *Proc. IEEE VisSym '02*, pages 251–260. IEEE Press, 2002.