Robust and Efficient Skeletal Graphs

Pavel Dimitrov   Carlos Phillips   Kaleem Siddiqi

McGill University
School of Computer Science &
Center for Intelligent Machines
{pdimit,carlos,siddiqi}@cim.mcgill.ca

Abstract

There has recently been significant interest in using representations based on abstractions of Blum's skeleton into a graph, for qualitative shape matching. The application of these techniques to large databases of shapes hinges on the availability of numerical algorithms for computing the medial axis. Unfortunately, this computation can be extremely subtle. Approaches based on Voronoi techniques preserve topology, but heuristic pruning measures are introduced to remove unwanted edges. Methods based on Euclidean distance functions can localize skeletal points accurately, but often at the cost of altering the object's topology. In this paper we introduce a new algorithm for computing subpixel skeletons which is robust and accurate, has low computational complexity, and preserves topology. The key idea is to measure the net outward flux of a vector field per unit area, and to detect locations where a conservation of energy principle is violated. This is done in conjunction with a thinning process applied in a rectangular lattice. We illustrate the approach with several examples of skeletal graphs for biological and man-made silhouettes.

1. Introduction

In recent years there has been significant interest in using graph-based abstractions of Blum's skeleton for qualitative shape recognition [10, 6, 11, 8, 14]. The application of such methods to large image databases hinges on the availability of robust and efficient algorithms for computing the medial axis, or approximations to it. Unfortunately, this is a subtle numerical problem. Methods based on Voronoi techniques preserve topology, but heuristic pruning measures are introduced to remove unwanted edges. Methods based on Euclidean distance functions can localize skeletal points accurately, but often at the cost of altering the object's topology. In this paper we introduce a new algorithm for computing subpixel skeletons which is robust and accurate, has low

computational complexity, and preserves topology. The key idea is to measure the net outward flux of a vector field per unit area, and to detect locations where a conservation of energy principle is violated. This is done in conjunction with a thinning process applied in a rectangular lattice. We illustrate the approach with several examples of skeletal graphs for biological and man-made silhouettes.

2. Divergence-Based Skeletons

Consider Blum's grassfire flow [3]

$$\frac{\partial \mathcal{C}}{\partial t} = \mathcal{N} \qquad (1)$$

acting on a 2D closed curve $\mathcal{C}$, such that each point on its boundary is moving with unit speed in the direction of the inward normal $\mathcal{N}$. In recent work, we have shown that this formulation leads to a Hamilton-Jacobi equation on the Euclidean distance function to the initial curve [12]. In physics, such equations are typically solved by looking at the evolution of the phase space of the equivalent Hamiltonian system. Since Hamiltonian systems are conservative, the locus of skeletal points coincides with locations where a conservation of energy principle is violated. This loss of energy can be used to formulate a natural criterion for detecting singularities of the distance function.

More specifically, let D be the Euclidean distance function to the initial curve $\mathcal{C}_0$. The magnitude of its gradient, $\| \nabla D \|$, is identical to 1 in its smooth regime. With $\mathbf{q} = (x, y)$, $\mathbf{p} = (D_x, D_y)$, the Hamiltonian system is given by

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}} = (0,0), \qquad \dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}} = -(D_x, D_y), \quad (2)$$

with an associated Hamiltonian function $H = 1 - (D_x^2 + D_y^2)^{\frac{1}{2}}$ [12]. It is straightforward to show that all Hamiltonian systems are conservative and hence divergence free. Conversely, when trajectories of the system intersect,

a conservation of energy principle is violated. This suggests a natural approach for detecting the skeleton: compute the divergence of the gradient vector field $\dot{\mathbf{q}}$ and detect locations where it is not zero. The divergence is defined as the net outward flux per unit area, as the area about the point shrinks to zero:

$$\mathrm{div}(\dot{\mathbf{q}}) \equiv \lim_{\Delta a \to 0} \frac{\int_L < \dot{\mathbf{q}}, \mathcal{N} > \mathrm{dl}}{\Delta a} \qquad (3)$$

Here Δa is the area, L is its bounding contour and $\mathcal{N}$ is the outward normal at each point on the contour. Via the divergence theorem

$$\int_a \mathrm{div}(\dot{\mathbf{q}})\mathrm{da} \equiv \int_L < \dot{\mathbf{q}}, \mathcal{N} > \mathrm{dl}. \qquad (4)$$

In other words, the integral of the divergence of the vector field within a finite area gives the net outward flux through the contour which bounds it. Locations where the flux is negative, and hence energy is lost, correspond to sinks or skeletal points of the interior. Similarly, locations where the flux is positive correspond to sources or skeletal points of the exterior.[1]

Figure 1 illustrates the divergence-based computation on a panther silhouette. The gradient vector field of the Euclidean distance function is shown on the top, with the total outward flux on the bottom. Observe that the smooth regime of the vector field gives zero flux (medium gray), its sinks coincide with the skeleton of the interior (dark gray), and its sources with the skeleton of the exterior (light gray). Hence, a threshold on the divergence map yields a close approximation to the skeleton, as used in [12]. However, in general it is impossible to guarantee that the result is homotopic to the original object by simple thresholding. A high threshold may yield a connected set, but it is not thin and unwanted branches are present, Figure 2 (top). A low threshold yields a thin set, but it may be disconnected, Figure 2 (bottom). The solution, as we shall now show, is to introduce additional constraints to ensure that the topology of the object is preserved. The main idea is to incorporate a homotopy preserving thinning process in a rectangular lattice, where the removal of points is guided by their divergence values.

3. Homotopy Preserving Skeletons

3.1. Divergence-Ordered Thinning

There is a long history of approaches to computing the skeleton which are rooted in *thinning* operations [1]. These methods attempt to realize Blum's grassfire formulation by peeling away layers from the object, while retaining special

---

[1]Note that the classical definition of divergence as the sum of the partial derivatives of the vector field in its component directions cannot be used where it is singular, and these are precisely the points we are interested in.
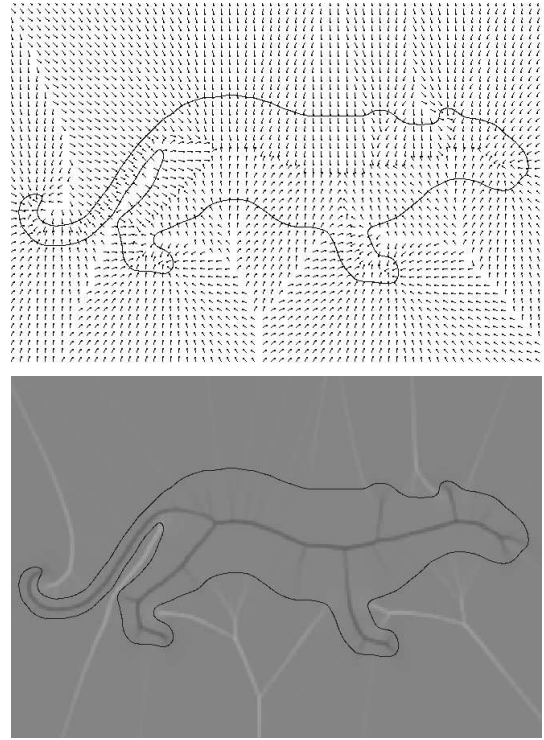

Figure 1: The gradient vector field of a signed distance function to the boundary of a panther shape (top), with the associated total outward flux (bottom). Whereas the smooth regime of the vector field gives zero flux (medium gray), its sinks correspond to the skeleton of the interior (dark gray) and its sources to the skeleton of the exterior (light gray).
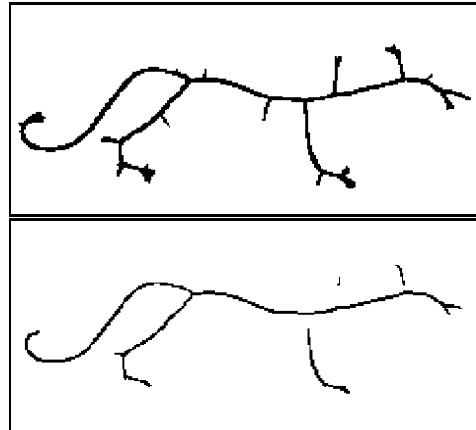

Figure 2: Thresholding the divergence map in Figure 1. A high threshold yields a connected set, but it is not thin, and unwanted branches are present (top). A low threshold yields a closer approximation to the desired skeleton, but the result is now disconnected (bottom).

points. It is possible to define erosion rules such that the topology of the object is preserved, but these methods are
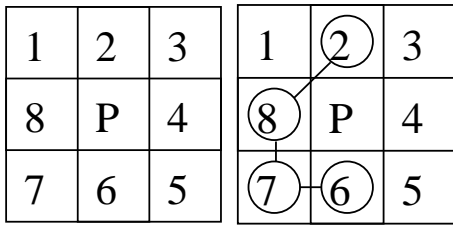
Figure 3: LEFT: A 3x3 neighborhood of a 2D digital point P in a rectangular lattice. RIGHT: An example neighborhood graph for which P can be removed. Note that there is no edge between neighbors 6 and 8 (see text).

quite sensitive to Euclidean transformations of the data and typically fail to localize skeletal points accurately. We shall exploit the interpretation based on divergence developed in the previous section to address these latter concerns.

To see how topology preserving erosion rules can be defined for a rectangular lattice let P be a 2D digital point which is contained in the object. The topology of the object can be altered if the removal of P either disconnects the object, or creates a hole in it. It is convenient to view this as a graph problem. Consider the 3x3 neighborhood of P, as shown in Figure 3 (left), and select those neighbors which are also contained within the object. Construct a neighborhood graph by placing edges between all pairs of neighbors (not including P) that are 4-adjacent or 8-adjacent to one another. If any of the 3-tuples $\{2, 3, 4\}$, $\{4, 5, 6\}$, $\{6, 7, 8\}$, or $\{8, 1, 2\}$, are nodes of the graph, remove the corresponding diagonal edges $\{2, 4\}$, $\{4, 6\}$, $\{6, 8\}$, or $\{8, 2\}$, respectively. This ensures that there are no degenerate cycles in the neighborhood graph (cycles of length 3). Now, observe that if the removal of P disconnects the object, or introduces a hole, the neighborhood graph will not be connected, or will have a cycle, respectively. Conversely, a connected graph that has no cycles is a tree. Hence, we have a simple criterion to decide whether or not P can be removed:

Proposition 1 *A 2D digital point P can be removed if and only if its 3x3 neighborhood graph, with cycles of length 3 removed, is a tree.*

A straightforward way of determining whether or not a graph is a tree is to check that its Euler characteristic $|V| - |E|$ (the number of vertices minus the number of edges) is identical to 1. Note that this check only has to be performed locally, in the 3x3 neighborhood of P. Figure 3 (right) shows an example neighborhood graph for which P can be removed.

The above proposition allows us to guarantee that a thinning process on a rectangular lattice preserves topology. However, the end result depends entirely on the order in which points are removed, so its relationship to the skeleton is unclear. In fact, most thinning methods give only a coarse

approximation to the skeleton unless additional constraints are introduced [2]. In the current context, we have derived a natural criterion for localizing skeletal points based on principles from physics. Specifically, since the total outward flux of the gradient vector field of the Euclidean distance function provides a continuous measure on the likelihood of a point being a sink or skeletal point, it is natural to *order* the removal of points by thinning according to this measure.

The outline of our approach is now in place. We shall apply a homotopy preserving thinning process to the interior of the object, where candidate points for removal will be considered in order of decreasing divergence. We shall preserve end points that are *significant*, i.e., their total outward flux is below some selected threshold, but will thin the rest of the object until no further points can be removed. The result will be a thin set in a rectangular lattice (a set without any interior), that is homotopic to the object and by the divergence measure will be as close as possible to the "true" skeleton.

3.2. End Points, Curve Points and Branch Points

In the continuous case, the distinction between an end point, a curve point and a branch point of the skeleton follows from the number of times a disc centered at the point intersects the skeleton, Figure 4. Specifically: 1) if $\exists$ an interval $(0, a]$ such that for any ϵ in that interval a disc of radius ϵ centered at P intersects the skeleton once, then P is an end point, 2) if $\exists$ an interval $(0, a]$ such that for any ϵ in that interval a disc of radius ϵ centered at P intersects the skeleton twice, then P is a curve point, and 3) if $\exists$ an interval $(0, a]$ such that for any ϵ in that interval a disc of radius ϵ centered at P intersects the skeleton three or more times, then P is a branch point.

For a digital skeleton in a rectangular lattice, i.e., a 4-connected or 8-connected 1 pixel thick curve, a very similar characterization applies. The smallest ϵ disc is now the ordered set of neighbors $\{1, 2, 3, 4, 5, 6, 7, 8\}$, and one simply has to count the number of times the path through these neighbors intersects the digital skeleton. It is also easy to see that in the course of thinning, a candidate end point of a 4-connected or 8-connected digital curve may be characterized as follows:

Proposition 2 *A point P could be an end point of a 1 pixel thick digital curve if, in a 3x3 neighborhood, it has a single neighbor, or it has two neighbors, both of which are 4-adjacent to one another.*

We now have criteria to identify end points as the object is being thinned and also to detect branch points, once no more digital points can be removed.
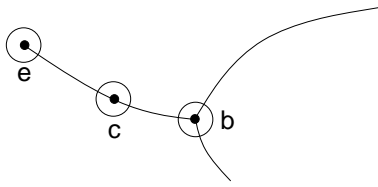
Figure 4: An end point *e*, a curve point *c* and a branch point *b* can each be distinguished from one another by determining the number of times a disc with infinitesimal radius centered at the point intersects the skeleton (see text).

3.3. The Algorithm

The thinning process can be made very efficient by observing that a point within the object which does not have at least one background point as an immediate neighbor cannot be removed, since this would create a hole. Therefore, the only potentially removable points are on the border of the object. Once a border point is removed, only its neighbors may become removable. This suggests the implementation of the thinning process using a heap. We now present the algorithm.

Algorithm 1 The divergence-ordered thinning algorithm

Part I: Total Outward Flux

Compute the signed distance transform of the object D [4].
Compute the gradient vector field ∇D.
Compute the net outward flux of ∇D using Eq. 4
    For each point P in the interior of the object
       $Flux(P) = \sum_{i=1}^{8} < N_i, \nabla D(P_i) >$,
       where P_i is an 8-neighbor of P and N_i is the
       outward normal at P_i of the unit disc centered at P.

Part II: Homotopy Preserving Thinning

For each border point P
if (P is removable)
  insert(P, Heap) with $Flux(P)$
  as the sorting key for insertion
While (Heap.size > 0)
P = HeapExtractMax(Heap)
  if (P is removable)
    if (P is not an end point) or ($Flux(P) > Thresh$)
      Remove P
      for all neighbors Q of P
        if (Q is removable)
          insert(Q, Heap)
    else mark P as a skeletal (end) point
    end { if }
  end { if }
end { while }

The complexity of the algorithm can be determined as follows. The computation of the distance transform [4], the gradient vector field and the total outward flux are all $\mathcal{O}(n)$
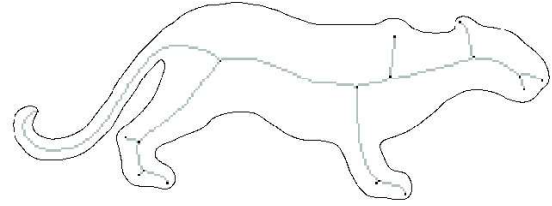

Figure 5: The digital skeleton of the panther silhouette obtained by divergence-ordered thinning. The skeleton is shown in light gray, with its end points and branch points shown in black. Compare with Figures 1 and 2.

operations, where n is the total number of points in the 2D array. The worst case complexity of the thinning process is $\mathcal{O}(k log(k))$, where k is the number of points in the interior of the object. The reasoning is as follows. A point can be inserted into the heap only when one of its neighbors is removed, and this can happen at most 8 times. Hence, the while loop may go through all k points of the object at most 8 times. At each iteration, insertion into a heap as well as the extraction of its minimum, are both $\mathcal{O}(log(l))$ operations, done no more than 8 times. Here l is the number of elements in the heap. There cannot be more than k elements in the heap, because we only have a total of k points in the object. The worst case complexity for thinning is therefore $\mathcal{O}8k8(log(k)) = \mathcal{O}k log(k)$. Hence, the complexity of the entire method is $\mathcal{O}(n) + \mathcal{O}(k log(k))$. The algorithm converges when the heap is empty, at which stage the set of points which remain comprise the skeleton. The end points have been labelled, and the branch points and interior points can be detected via the procedure described in Section 3.2.

Figure 5 illustrates the output of the algorithm on the panther silhouette, with the digital skeleton shown in light gray and the end points and branch points shown in black. Observe that the skeleton corresponds closely to the sink points in the divergence map of Figure 1. However, in contrast to the earlier results obtained by thresholding (Figure 2), the result is now 1 pixel thin (without an interior) and is homotopic to the object. In fact, this property is ensured by Propositions 1 and 2. [2]

3.4. A Subpixel Skeletal Graph

It is straightforward to interpret the digital skeleton, with end points and branch points labelled, as a graph. Let the set of end points and branch points comprise the vertices, with

---

[2]The one exception is the case of a 2x2 square block, which can occur if there are branches at each of its corners. In order to preserve this symmetry, when we abstract the skeleton into a graph, each lattice point of the 2x2 square will be stored with its corresponding branch, and a new branch point will be placed in the middle.

an edge representing the ordered list of digital points connecting two vertices. The digital points can be ordered by applying a depth-first-search algorithm, starting at a vertex, where one pixel follows another if the two are 4-adjacent or 8-adjacent, but do not neighbor the same branch point. The search terminates when another vertex is reached.
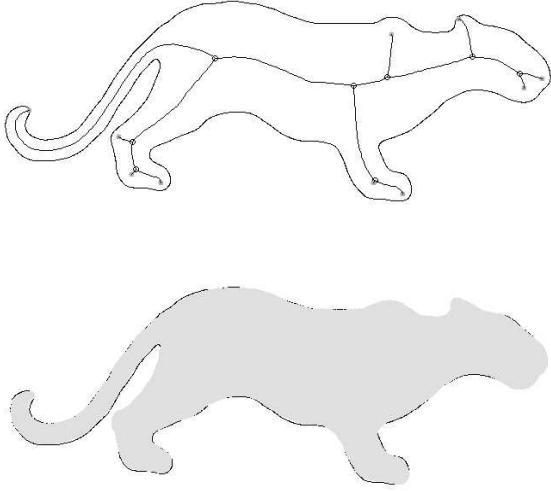

Figure 6: TOP: A subpixel skeletal graph, with branch points shown as empty circles and end points as filled circles. Compare with the digital skeleton, Figure 5, from which it was derived. BOTTOM: The reconstruction as the envelope of the maximal inscribed disks (grey) of the skeleton, overlayed on the original object.

In our implementation we have used a subpixel contour tracer to obtain a piecewise circular arc representation of the boundary of the object [13]. In this case, all simulations can be based on an *exact* distance function, by determining the distance of each lattice point to the closest circular arc. As shown in Appendix A, it is then possible to move each point on the digital skeleton so that it is within arbitrary precision of a *true* skeletal point, leading to a subpixel skeletal graph. For curve points the idea is to search in a direction defined by the pan-normal from the object's boundary to the point, and to iteratively refine this search. For branch points the idea is to move the point till it is arbitrarily close to the center of a maximal inscribed disc which touches the contour at three or more points.

4. Examples

We now present examples of subpixel skeletal graphs. In each case, the same divergence threshold was used to determine which end points to preserve. Figure 6 (top) shows the subpixel skeleton for the panther silhouette, derived from the earlier digital skeleton in Figure 5. Branch points are depicted with empty circles and end points with filled circles. Figure 6 (bottom) shows the object reconstructed as the envelope of the maximal inscribed discs associated with each skeletal point. Figure 7 depicts subpixel skeletal graphs for a wide variety of 2D shapes representing a range of object categories. The results demonstrate the robustness of the framework under Euclidean transformations, as well as changes in scale.

5. Discussion and Related Work

We have introduced a novel algorithm for computing subpixel skeletal graphs which is robust, accurate, computationally efficient, invariant to Euclidean transformations, and homotopy preserving. The essential idea is to combine a divergence computation on the gradient vector field of the Euclidean distance function, which we introduced in earlier work [12], with a thinning process that preserves topology. The identification of end points and branch points allows the result to be interpreted as a graph. The digital skeleton is then shifted to be within arbitrary precision of the "true" skeleton. We have illustrated the approach on a variety of 2D shapes. It should be clear that framework applies to shapes with arbitrary topology. We have also extended the method to a framework for computing medial surfaces of 3D objects [5].

We note that in related work, Leymarie and Levine have utilized the magnitude of the gradient vector field to design a potential function to attract a snake moving in from the shape's boundary [7]. This leads to an intuitive simulation of the grassfire, however the interpretation of its singularities as a skeletal graph is not immediate. Geiger et. al have introduced a variational approach to computing symmetric axis trees, where portions of a curve are matched against others, incorporating constraints including co-circularity and parallelism [8]. Zhu has posed the computation of the medial axis as a statistical inference problem, leading to an approximation of the skeleton [16]. Tek and Kimia have introduced a promising approach for calculating symmetry maps, which is based on the combination of a wavefront propagation technique with the use of an exact (analytic) distance function [15]. Finally, Malandain and Fernandez-Vidal obtain two sets based on thresholding a function of two measures, ϕ and d, to characterize the singularities of the Euclidean distance function [9]. Whereas empirical results are good, the choice of appropriate thresholds for these measures, as well as strategies for combining them, are based on heuristics.

In contrast, our method is rooted in a physics-based analysis of the gradient vector field of the Euclidean distance function. This justifies the use of the divergence theorem to compute the total outward flux of the vector field, and to locate points where energy is absorbed. In this paper we
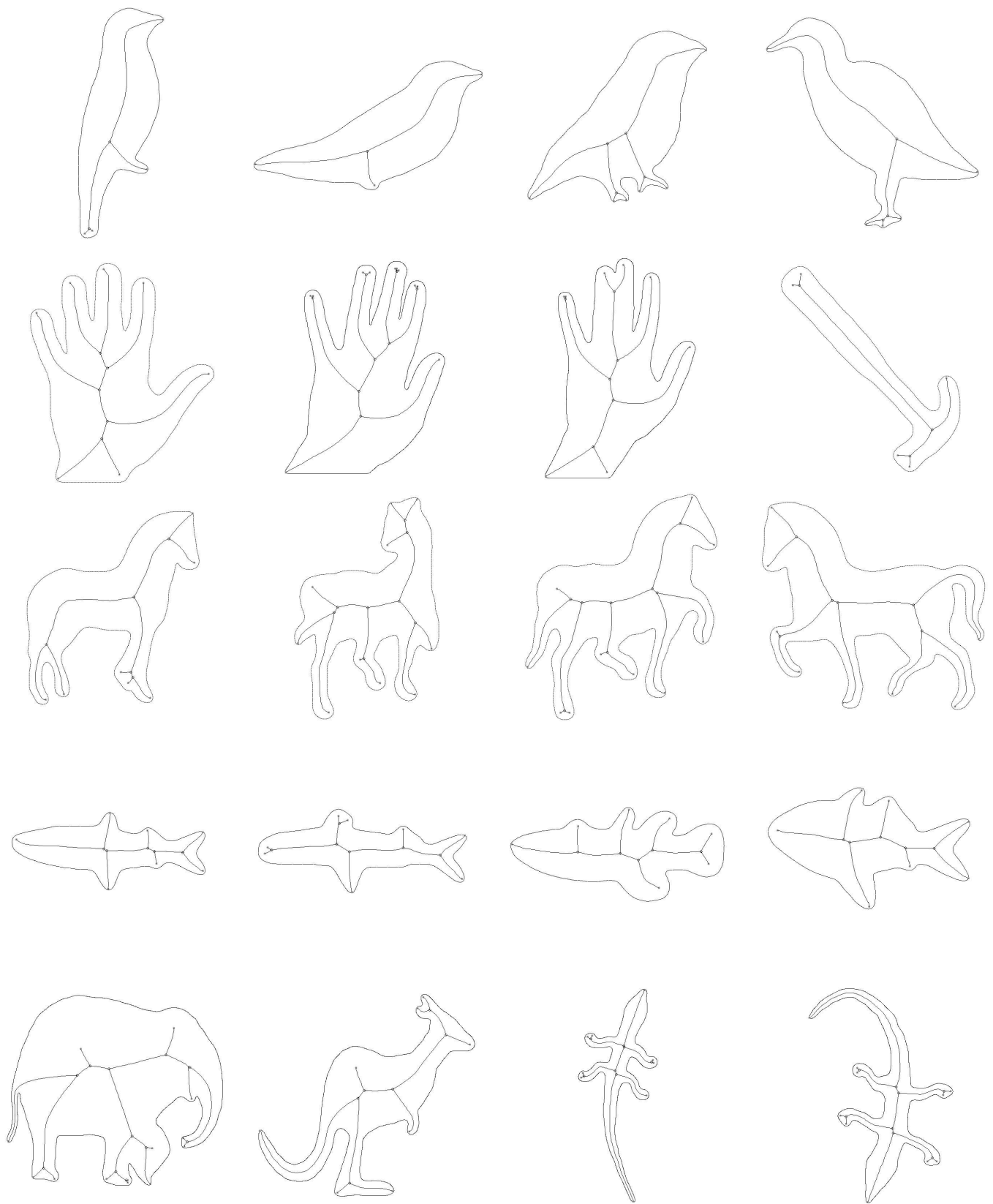
Figure 7: Subpixel skeletons for a wide variety of 2D shapes, obtained by divergence-ordered thinning.

have combined this measure with a homotopy preserving thinning algorithm in 2D, followed by a subpixel shift to efficiently obtain robust and accurate skeletal graphs.

A. Subpixel Shifting

We now show how points on a digital skeleton can be shifted to be within arbitrary precision of "true" skeletal points, by making use of an exact distance function. Consider a closed contour and its digital skeleton obtained by divergence-ordered thinning. For each end point of the digital skeleton find the closest point on the contour, and use the set of closest points to partition the contour. Following the partition, we can refer to the contour segments C_L and C_R associated with either side of a digital branch B.

We now proceed to shift the points contained in the branch B. Let $p \in B$ and $a \in C_R$, $b \in C_L$ be such that $d(p,a) = \min_{x \in C_R} d(p,x)$ and $d(p,b) = \min_{x \in C_L} d(p,x)$. Without loss of generality, assume $d(p,a) \le d(p,b)$, see Figure 8. If we extend the line segment ap to a point u so that the triangle aub becomes isosceles, we must get that u is on the other side of the true skeleton with respect to p. This is true because the line segment ap is normal to the contour segment C_R at a, by the definition of a closest point, and thus a is closest to a skeletal point on the extension of the line segment. Since $d(p,a) \le d(p,b)$, u is at least on the skeleton if not further, by the definition of a skeletal point.

Now, given p and u as above, find the mid-point m between them. Update p to m if m is closer to C_R than to C_L, otherwise update u to m. When the segment pu is small enough, return one of p or u. Note that the actual precision of the shifting depends on the precision of the end points which were used to segment the contour.

Next, we consider the shifting of a digital branch point, using a slightly different approach. Here the essential idea is to move the point so that it is arbitrarily close to the center of a maximal inscribed disc which touches the boundary at at least three points.

Using the digital branch point as a first approximation, identify the closest contour points to it, one on each of the three closest contour segments. Then, shift the branch point to the center of the unique circle which passes through these three contour points. Repeat the procedure until the distances to the three closest points are within a specified tolerance of one another.

References

[1] C. Arcelli and G. Sanniti di Baja. A width-independent fast thinning algorithm. *IEEE PAMI*, 7(4):463–474, July 1985.
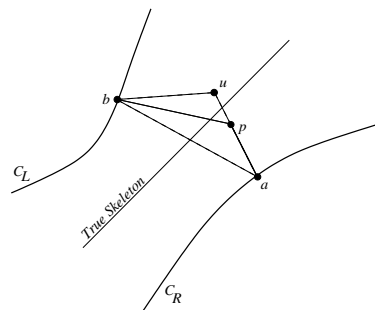
[2] G. Bertrand. A parallel thinning algorithm for medial surfaces. *Pattern Recognition Letters*, 16:979–986, 1995.

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

[4] G. Borgefors. Distance transformations in arbitrary dimensions. *CVGIP*, 27:321–345, 1984.

[5] S. Bouix and K. Siddiqi. Divergence-based medial surfaces. In *ECCV'2000*, Dublin, Ireland, June 2000.

[6] A. François and G. Medioni. Generic shape learning and recognition. In *International Workshop on Object Representation in Computer Vision*, April 1996.

[7] F. Leymarie and M. D. Levine. Simulating the grassfire transform using an active contour model. *IEEE PAMI*, 14(1):56–75, Jan. 1992.

[8] T.-L. Liu, D. Geiger, and R. V. Kohn. Representation and self-similarity of shapes. In *Sixth International Conference on Computer Vision*, 1998.

[9] G. Malandain and S. Fernandez-Vidal. Euclidean skeletons. *Image and Vision Computing*, 16:317–327, 1998.

[10] R. L. Ogniewicz. *Discrete Voronoi Skeletons*. Hartung-Gorre, 1993.

[11] D. Sharvit, J. Chan, H. Tek, and B. B. Kimia. Symmetry-based indexing of image databases. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, June 1998.

[12] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. The hamilton-jacobi skeleton. In *ICCV'99*, pages 828–834, Kerkyra, Greece, September 1999.

[13] K. Siddiqi, B. B. Kimia, and C. Shu. Geometric shock-capturing eno schemes for subpixel interpolation, computation and curve evolution. *Graphical Models and Image Processing*, 59(5):278–301, September 1997.

[14] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.

[15] H. Tek and B. B. Kimia. Symmetry maps of free-form curve segments via wave propagation. In *ICCV'99*, pages 362–369, Kerkyra, Greece, September 1999.

[16] S. C. Zhu. Stochastic computation of the medial axis in markov random fields. In *CVPR'98*, pages 72–79, Santa Barbara, CA, June 1998.

Figure 8: An illustration of a subpixel shift; see the text for a discussion.