Metadata of the chapter that will be visualized online

Book Title	Encyclopedia of Computer Vision	
Book Copyright Year	2013	
Copyright Holder	Springer Science+Business Media, LLC	
Title	Shock Graph	
Author	Degree	Prof.
	Given Name	**Sven J.**
	Particle	
	Family Name	**Dickinson**
	Suffix	
	Phone	
	Fax	
	Email	sven@cs.toronto.edu
Affiliation	Division	Department of Computer Science
	Organization	University of Toronto
	Street	6 King's College Rd.
	City	Toronto
	State	ON
	Country	Canada
Author	Degree	Dr.
	Given Name	**Ali**
	Particle	
	Family Name	**Shokoufandeh**
	Suffix	
	Phone	
	Fax	
	Email	ashokouf@cs.drexel.edu
Affiliation	Division	Department of Computer Science
	Organization	Drexel University
	Street	3141 Chestnut St.
	Postcode	19104
	City	Philadelphia
	State	PA
	Country	USA

Author	Degree	Dr.
	Given Name	**Kaleem**
	Particle	
	Family Name	**Siddiqi**
	Suffix	
	Phone	
	Fax	
	Email	siddiqi@cim.mcgill.ca
Affiliation	Division	School of Computer Science
	Organization	McGill University
	Street	3480 University Street
	Postcode	H3A 2A7
	City	Montreal
	State	PQ
	Country	Canada

S

Shock Graph

Sven J. Dickinson[1], Ali Shokoufandeh[2] and Kaleem Siddiqi[3]
[1]Department of Computer Science, University of Toronto, Toronto, ON, Canada
[2]Department of Computer Science, Drexel University, Philadelphia, PA, USA
[3]School of Computer Science, McGill University, Montreal, PQ, Canada

Related Concepts

▸Grassfire Flow; ▸Graph Matching; ▸Medial Axis/Skeleton; ▸Many-to-Many Graph Matching; ▸Object Categorization

Definition

The shock graph is obtained from the 2D Blum medial axis by incorporating properties of the radius function along the skeleton. The direction in which the radius function increases, or equivalently, the direction of the grassfire flow, is used to order groups of skeletal points and to derive parent-child relationships. This results in a directed acyclic graph whose nodes represent skeletal points and whose edges represent adjacency relationships. A variant of this construction associates skeletal points with edges, with the nodes representing the adjacencies.

Background

When Blum conceived of the medial axis or skeleton, his goal was to use it as a means to categorize objects from their projected (2D) outlines [4]. Specifically, by associating the direction of increasing radius value along a skeletal branch, or equivalently the direction of propagation of singularies of the grassfire flow, he proposed the concept of an axis-morphology or *a-morph* by which to achieve object categorization. His basic insight was that this could lead to a decomposition that reflected the qualitative part structure of the object. As an example, ignoring their detailed boundary geometry, outlines of hands would have similar a-morphs and these would be quite distinct from those of outlines of humans, fish or other object classes. In fact, he drew upon these later examples towards the end of his classic paper [4], where he also sketched possible extensions to 3D.

Whereas much has been written about medial or skeletal representations over the years (see [23] and also the medial axis/skeleton entry in this encyclopedia) the idea that an a-morph was essentially a directed graph which could be used for object recognition caught on only in the early 1990s. One likely reason is that it took the image analysis and computer vision communities many years to develop robust algorithms for skeleton computation. Since this time, however, a variety of successful approaches to view-based recognition using shock-graphs have been proposed and have been validated on large databases. Several of these are described in the present entry. There also exist more recent variants of the shock graph, such as Macrini et al.'s bone graph [14], which attempt to

K. Ikeuchi (ed.), *Encyclopedia of Computer Vision*, DOI 10.1007/978-0-387-31439-6,

mitigate the representational instability of the Blum medial axis. In fact, the mapping of the Blum skeleton to a graph-based representation, of which the shock graph is the most widely researched example, remains an active area of investigation.

Theory

The Blum medial axis or skeleton of a 2D outline is homotopic to it and is comprised of three types of skeletal points: endpoints of skeletal curves, interior points and branch points. The branch points are generically of degree 3, i.e., three skeletal curves are connected at a branch point. A formal classification is presented in [11]. The shock graph takes the 2D skeleton of a simple closed curve as input (one without holes) and labels each skeletal point according to whether the radius function at it is increasing monotonically (a 1-shock), is a local minimum (a 2-shock), is constant (a 3-shock) or is a local maximum (a 4-shock). Groups of adjacent 1-shocks are considered together, as are groups of 3-shocks. Given this interpretation, a directed acyclic graph is obtained by considering the skeletal points with the largest radii, which are the last to form in the grassfire flow, as the children of a dummy root node. The children are then placed, recursively, in order of decreasing radius value. This process of reversing the grassfire flow and adding 1-shock groups or 3-shock groups as children, is governed by the rules of a grammar, as shown in [24].

Rather than provide all the details of the grammar in this entry, the reader is referred to the examples in Fig. 1, which show the construction of the shock graphs of two brush shapes. The medial axis of each object is shown in the bottom row, with distinct groups of shocks being given a unique color (3-shocks are shown in yellow). In the labeling, the shock type appears first, followed by a unique identifier. The associated shock graphs are shown in the top row. It is clear that each shape is abstracted by a single root node (the 3-shock group describing the elongated portion of the brush), with its children being additional protrusions (1-shock groups). One of these protrusions has a 3-shock group as a child, which describes the handle of each brush. From this example it is evident that the shock graph is a formalization of Blum's a-morph, with the advantage that it lends itself to the use of graph-based methods for object categorization, as detailed below.

It is also important to point out that there is a variant of the shock graph where the representation places the skeletal points at edges of the graph, with the nodes representing connections. This variant is described in detail in [18, 19]. This representation has lead to different but equally successful methods for object recognition, based on a notion of the edit-distance between two graphs. The results of using this approach are also briefly described below.

Shock Graph-Based Object Categorization

An object categorization system based on shock graphs consists of two components: (1) an indexing component, which takes an input shock graph and returns, from a large database of model shock graphs, a small number of candidate shock graphs that might account for the input; and (2) a matching component, which takes one of the candidates and the input, and computes a similarity (or distance), along with a set of node correspondences. Under ideal conditions, the input shock graph would contain no artifacts due to noise, occlusion, or clutter, and would be isomorphic to one of the model shock graphs (provided that the input object represents one of the model objects). However, such conditions are highly unlikely, for in addition to noise, occlusion, and scene clutter, ligature-induced instabilities [1] often lead to spurious nodes/edges as well as medial branch oversegmentation. Formulating the problem as graph isomorphism, subgraph isomorphism, or even largest isomorphic subgraph will not lead to a meaningful solution, for large, or even significant isomorphisms may simply not exist between two shock graphs that represent instances of the same category. The shock graph indexing and matching problems are therefore inexact graph indexing/matching problems.

Indexing Shock Graphs

Given an input shock graph, the goal of the indexing module is to quickly (sublinearly) retrieve a small number of candidate model database shock graphs among which the input is likely included. As mentioned above, the input shock graph may be corrupted in a number of ways, precluding a simple global (based on the entire input) indexing framework. For example: (1) occlusion may remove part of the input shock graph and replace the missing part with a shock graph (or subgraph) belonging to a different object; (2) shadows or poor ilumination may simply delete some portion of the input shock graph; (3) scene clutter may embed the object shock graph (or portion thereof) in a much larger "scene" shock graph; and (4) ligature-based instability may introduce spurious nodes or may overpartition other nodes in the input shock graph. These factors require a part-based indexing framework that can operate in the presence of noise, occlusion, clutter, and ligature-based instability.

One such indexing framework that is applicable to not only shock graphs but any hierarchical, graph-based representation (specifically, any directed acyclic graph-based representation) was introduced by Shokoufandeh et al. [22], originally for the purpose of shock graph indexing. The key concept behind the approach is to capture the abstract shape of a graph (or subgraph) with a low-dimensional vector, yielding an efficient indexing mechanism. Capturing the abstract shape of a graph is important so that the index is invariant to noise and minor within-class shape deformation. Indexing at the part level is important in the presence of occlusion and scene clutter. Mapping a discrete graph structure to a low-dimensional point facilitates a simple nearest-neighbor search in a geometric space for similar model parts which, in turn, can vote for those model objects that contain those parts. Those model objects receiving the largest votes represent those candidate objects passed to the shock graph matching module for a more detailed analysis.

The graph-based shape abstraction is computed at every non-leaf node, and captures the abstract "shape" of the underlying subgraph rooted at that node. Therefore, each non-leaf node (with only four shock graph node types, leaf nodes are far too uninformative/ambiguous) "votes" for those objects that share its substructure; the root of the graph would therefore vote at the object level, and would be meaningful only if the object were unoccluded and not embedded in a larger scene. Mapping the structure of a rooted subgraph to a vector assigned to the subgraph's root is based on a spectral analysis of the graph's structure. The eigenvalues of a graph's adjacency matrix (whose values are 0,1,−1) capture important properties of the degree distribution of the graph's nodes. The eigenvalues can be combined to yield a low-dimensional abstraction of the graph's shape in terms of how and where the edges are distributed throughout the graph. Moreover, such a spectral "signature," called the *topological signature vector*, is proven to be stable under minor perturbations of graph structure due to noise. Details of the approach are found in [22], while an application of the same indexing framework to a different hierarchical graph, specifically a 3-D medial surface graph, can be found in [25].

Matching Shock Graphs

Given two shock graphs, e.g., one representing the input and one representing a model candidate, the matching component needs to return not only a similarity or distance measure that can be used to rank order the candidates, but also an explicit correspondence that defines which model nodes correspond to which input nodes. Such correspondence is necessary, for in the case of a cluttered scene, those nodes found to match a given model would be removed, and another candidate model matched to the remaining nodes. Moreover, the correspondence need not be one-to-one, for in the case of ligature-induced medial branch oversegmentation, node correspondence many be *many-to-many*.

Siddiqi et al. [24] developed a matching algorithm for shock graphs which, like the indexing framework of Shokoufandeh et al. [22] discussed above, can be applied to the matching of any directed acyclic graph structure, provided that a domain-dependent node similarity function is given. The algorithm is based on the same spectral graph theoretic abstraction that forms the heart of the indexing component described above. The algorithm fomulates the matching of two graphs as finding a maximal matching in a bipartite graph over the two nodes sets (input and model). The edge weights (each spanning one input shock graph node and one model shock graph node) in the graph have two components: (1) the distance between the two nodes' respective topoligical similarity vectors, defining the similarity of their underlying graph structures (rooted at the two nodes); and (2) a node similarity function (the only domain-dependent component of the algorithm) that defines the similarity of the node attributes (for shock graphs, this encodes the geometric similarity between the two skeletal branches corresponding to the two nodes).

At first glance, the matching algorithm would seem to throw out all the important hierarchical structure in the two graphs (absent in the bipartite graph); nodes in one graph are matched to nodes in the other graph, but the edges in the two original graphs appear to play no role. However, the key contribution of the algorithm is that the hierarchical edge structure is brought back via the topological signature vector similarity term. For the bipartite matching algorithm to match two nodes (i.e., select that edge in the matching), both their geometric similarity and their topological similarity must be high. In other words, the contents of the two nodes must be similar and the subgraphs rooted at the two nodes must be similar. The algorithm iterates by computing a matching, selecting the best edge from the matching (having maximum similarity), adding it to the solution set, and recursively continuing the process on the remaining graphs (after removing the pair of matching nodes defined by the best edge). Details of the approach are found in [24], while its application to other shape matching problems is described in [21] (multiscale blob and ridge graphs), [25] (medial surface graphs), and [8, 26] (curve skeleton graphs).

The above algorithm eventually yields a one-to-one node correspondence between the two graphs. However, because the algorithm generates the node correspondence in a coarse-to-fine manner, stopping the algorithm at the level of a coarse node-to-node correspondence defines an explicit many-to-many correspondence between the nodes in the subgraphs rooted at the coarse nodes. Moreover, since the topological signature vectors are stable under small amounts of additive graph noise, similarity can remain high even though the two subgraphs may have different numbers of nodes. As the cardinalities of the two graphs' node sets begin to differ more dramatically, for example due to heavy under- or over-segmentation, the method breaks down and more powerful many-to-many graph matching must be employed.

One such method for many-to-many graph matching of medial axis-based graphs was proposed by Demirci et al. [9, 10]. Their algorithm transforms the graphs into a finite dimension metric space in which an approximate solution to the many-to-many matching problem becomes tractable. The embedding step will result in a set of points, each representing a vertex of the original graph. Their proposed embedding has the additional property that pairwise distances between points in the target metric space closely resemble the shortest-path distances between the corresponding nodes in the graphs. Matching two graphs can then be formulated as the problem of matching their two embeddings. The many-to-many matching of the two embeddings then can be computed by solving a transportation problem using the Earth Mover's Distance algorithm [7]. The solution of this latter problem computes the mass which flows from one weighted point set to another that minimize the total transportation cost. The computed flows, in turn, define the many-to-many node correspondences between the original graphs.

The problem of matching shock graphs has also been studied in the context of edit-distance methods [18, 29]. These algorithms estimate the cost of matching as a function of edit operations, including node relabelings, additions and deletions, and edge contraction that transform one graph into another. A fundamental issue in devising algorithms based on edit-distance is the choice of cost of each operation. Torsello and Hancock [29] use the heuristic proposed by Bunke [5] for the cost associated with their edit operations. For example, the cost of relabeling elements is less than the cost of performing a deletion followed by inserting a new node with a new label. In contrast, Sebastian et al. [18] propose a multi-step heuristic to derive their edit costs. Their overall heuristic is centered around the notion of a shape cell, i.e., a collection of shapes which have identical shock graph topology. They define the cost of the deformation operation as a function of the discrepancy between matching shock attributes of shapes within a given cell. The cost associated with other edit operations is derived as the limit of the deformation cost when a shape moves to the boundary a shape cell.

Caelli and Kosinov [6] show how inexact matching can be utilized for measuring shape similarity between shock graphs. Their method establishes correspondence between sets (clusters) of vertices of two given graphs and as such can be viewed as a many-to-many matching approach. Their algorithm can be viewed as a generalization of the approach of Scott and Longuet-Higgins [17]. The actual matching is established using the renormalization of projections of vertices into the eigenspaces of graphs combined with a form of relational clustering. Similar to other inexact matching algorithms, their eigenspace renormalization projection clustering method is able to match graphs with different numbers of vertices.

Experimental Results

This section presents some examples of shock graphs and their matchings using the approaches described above. Figure 1(top) illustrates two shock graphs, describing different views of a brush, computed by the algorithm of Siddiqi et al. [24]. The underlying shocks, along with the computed matchings between segments (nodes), are shown in Fig. 1(bottom). Figure 2 represents the ability of the algorithm to compare objects based on their prototipical or coarse shape. Here, columns 2 through 10 denote the prototype views for each of nine object classes. The similarity between the prototypes and some of the objects in the database is reflected in the rows of this table. For each row, a box has been placed around the most similar shape. Demirci et al. [9] also evaluated the effectiveness of their matching algorithm for shape retrieval based on shock graphs from the Rutgers Tool Database [24]. Figure 3 shows some examples of the many-to-many feature matching results obtained from the algorithm for some of the objects in the Rutgers Tools Database. Finally, Fig. 4 shows the results obtained from applying the edit-distance algorithm of Sebastian et al. [18] to the matching of shock segments. Note that their edit distance algorithm will also produce a sequence of intermediate shock graphs that identify the steps of the transformation of one input shock graph to another.

Challenges

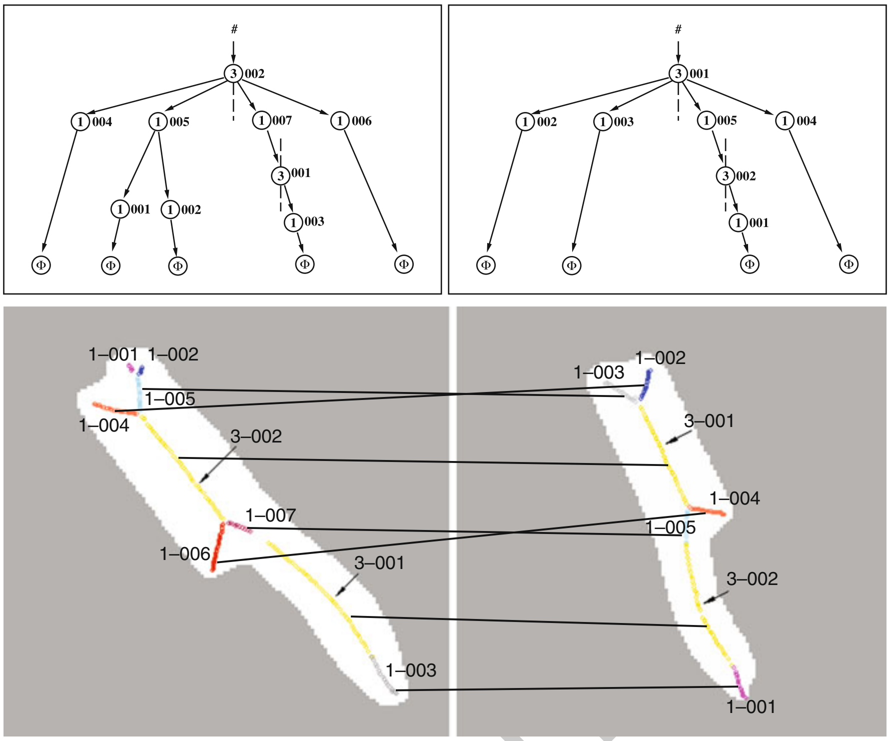
Symmetry is a powerful shape regularity that has formed the basis of many shape representations, including generalized cylinders [3], superquadrics [16], and geons [2]. Just as geons provide a qualitative and discrete shape abstraction of a generalized cylinder, shock graphs provide a discrete and qualitative shape abstraction of a medial axis. The resulting graph is ideally suited to shape categorization, for it is part-based, is stable under within-class deformation, and is stable under part articulation. However, the shock graph also faces some important challenges. First of all, it assumes that a closed contour has been recovered from an image, separating figure from background. While figure-ground segmentation remains an open research problem, it is important to note that in a categorization system, a perfect figure-ground separation may not be necessary. If a significant portion of the figure's boundary is correctly segmented, a significant portion of the resulting shock graph may be correct – enough to yield the correct candidate (among the list of returned candidates) during indexing. Still, while a shock graph does preserve locality of representation, significant figure-ground segmentation errors can propagate through the representation, disrupting it to a degree that prevents effective indexing. A recent attempt to recover a symmetric part decomposition from a cluttered scene has been reported by Levinshtein et al. [13], in which symmetric parts are detected locally (bottom-up) and then grouped to form an approximation to a medial axis.

The second challenge facing the shock graph is the ligature-based instability discussed earlier [1]. A number of approaches exist to try and regularize the medial axis through boundary smoothing, e.g., [12, 20, 27]; however, these methods do not effectively address the ligature structure. Other methods have sought to abstract the medial axis by regularizing out small internal branches, e.g., [28, 30]; however these methods don't explicitly target ligature structure. A recent promising approach to abstracting out ligature structure is proposed by Macrini et al. [14, 15], yielding a representation, called the *bone graph*, whose parts are the non-ligature medial branches that represent the salient parts and whose edges represent the "glue" (defined by the ligature branches) that binds the parts.

References

1. August J, Siddiqi K, Zucker S (1999) Ligature instabilities in the perceptual organization of shape. CVIU 76(3): 231–243
2. Biederman I (1985) Human image understanding: Recent research and a theory. Comput Vis Graph Image Process 32:29–73
3. Binford TO (1971) Visual perception by computer. In: Proceedings of the IEEE conference on systems and control, Miami
4. Harry Blum (1973) Biological shape and visual science. J Theor Biol 38:205–287
5. Bunke H (1997) On a relation between graph edit distance and maximum common subgraph. Pattern Recognit Lett 18(8):689–694
6. Caelli T, Kosinov S (2004) An eigenspace projection clustering method for inexact graph matching. IEEE Trans Pattern Anal Mach Intell 26(4):515–519

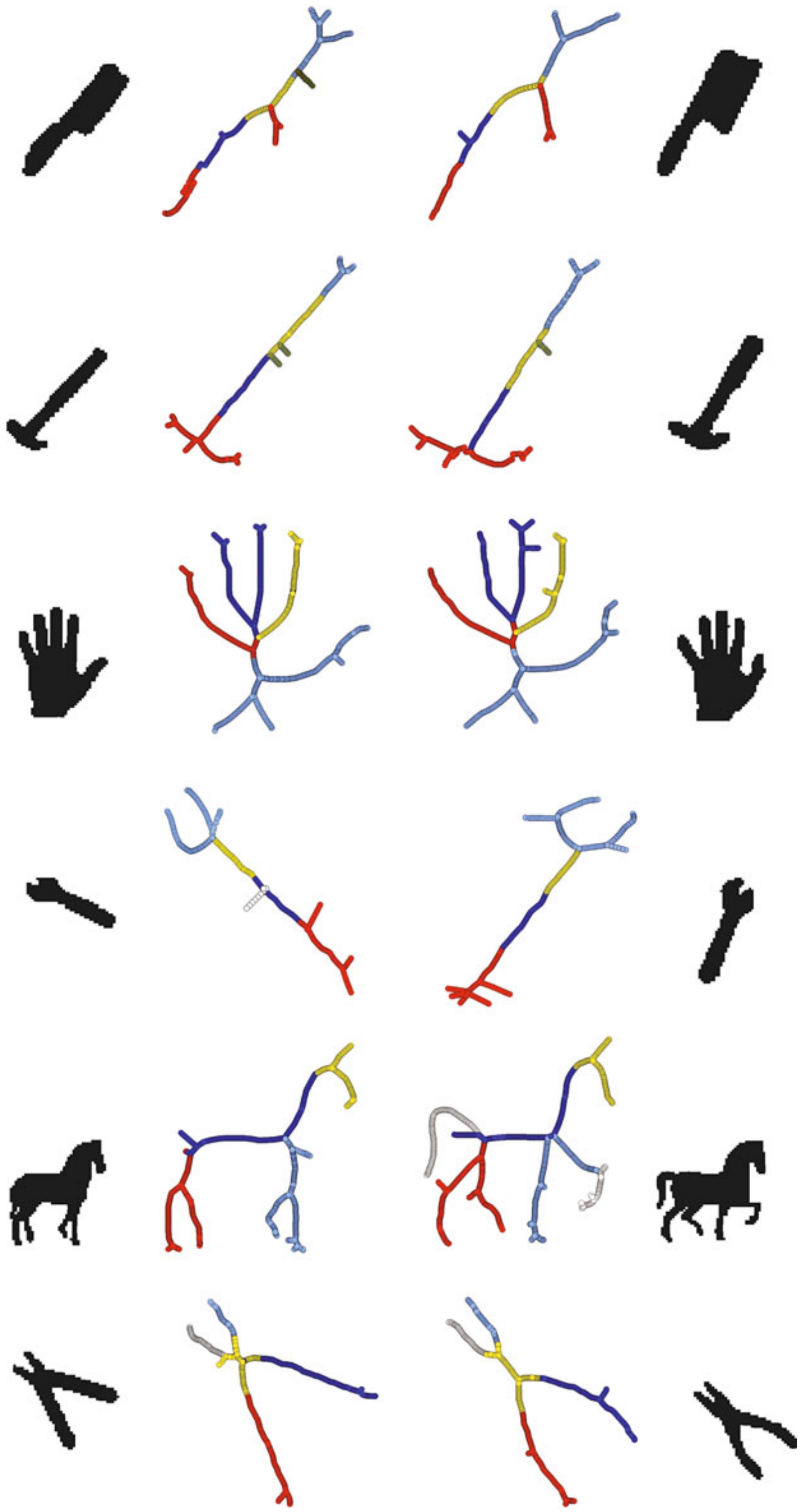
7. Cohen SD, Guibas LJ (1999) The earth mover's distance under transformation sets. In: Proceedings of the international conference on computer vision (ICCV), Kerkyra, pp 1076–1083
8. Cornea N, Demirci MF, Silver D, Shokoufandeh A, Dickinson S, Kantor P (2005) 3d object retrieval using many-to-many matching of curve skeletons. In: Proceedings of the international conference on shape modeling and applications (SMI), MIT, Cambridge, pp 368–373
9. Denirci MF, Shokoufandeh A, Keselman Y, Bretzner L, Dickinson S (2006) Object recognition as many-to-many feature matching. Int J Comput Vis 69(2):203–222
10. Demirci F, Shokoufandeh A, Dickinson S (2009) Skeletal shape abstraction from examples. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 31(5):944–952
11. Giblin Peter J, Kimia Peter J (2003) On the local form and transitions of symmetry sets, medial axes, and shocks. IJCV 54(1–3):143–157
12. Katz Robert A, Pizer Stephen M (2003) Untangling the blum medial axis transform. Int J Comput Vis 55(2–3): 139–153 .
13. Levinshtein A, Sminchisescu C, Dickinson S (2009) Multi-scale symmetric part detection and grouping. In: Proceedings of the international conference on computer vision (ICCV), Kyoto
14. Macrini D, Dickinson S, Fleet D, Siddiqi K (2011) Bone graphs: Medial shape parsing and abstraction. Comput Vis Image Underst (CVIU), special issue on graph-based representations, Special Issue on Graph-Based Representations, 115(7):1044–1061
15. Macrini D, Dickinson S, Fleet D, Siddiqi K (2011) Object categorization using bone graphs. Computer Vision and Image Understanding (CVIU) 115(8):1187–1206
16. Pentland A (1986) Perceptual organization and the representation of natural form. Artif Intell 28:293–331
17. Scott G, Longuet-Higgins H (1991) An algorithm for associating the features of two patterns. Proceedings of royal society of london B244:21–26
18. Sebastian T, Klein P, Kimia B (2001) Recognition of shapes by editing shock graphs. In: Proceedings of the international conference on computer vision (ICCV), Vancouver 755–762
19. Sebastian T, Klein P, Kimia B (2004) Recognition of shapes by editing shock graphs. IEEE Trans Pattern Anal Mach Intell 26:550–571
20. Doron Shaked, Bruckstein Alfred M (1998) Pruning medial axes. Comput Vis Image Underst 69(2):156–169
21. Shokoufandeh A, Bretzner L, Macrini D, Demirci MF, Jönsson C, Dickinson S (2006) The representation and matching of categorical shape. Comput Vis Image Underst 103(2):139–154
22. Shokoufandeh A, Macrini D, Dickinson S, Siddiqi K, Zucker SW (2005) Indexing hierarchical structures using graph spectra. IEEE Trans Pattern Anal Mach Intell 27(7):1125–1140
23. Siddiqi K, Pizer Stephen M (2008) Medial representations: mathematics, algorithms and applications. Springer, Dordrecht
24. Siddiqi K, Shokoufandeh A, Dickinson S, Zucker S (1999) Shock graphs and shape matching. Int J Comput Vis 30: 1–24
25. Siddiqi K, Zhang J, Macrini D, Shokoufandeh A, Bioux S, Dickinson S (2008) Retrieving articulated 3-d models using medial surfaces. Mach Vis Appl 19(4):261–275
26. Sundar H, Silver D, Gagvani N, Dickinson S (2003) Skeleton based shape matching and retrieval. In: Proceedings of the international conference on shape modelling and applications (SMI), Seoul, pp 130–142
27. Hüseyin Tek, Kimia Benjamin B (2001) Boundary smoothing via symmetry transforms. J Math Imaging Vis 14(3):211–223
28. Telea A, Sminchisescu C, Dickinson S (2004) Optimal inference for hierarchical skeleton abstraction. In: Proceedings of the international conference on pattern recognition, Cambridge, UK, pp 19–22
29. Torsello A, Hancock ER (2003) Computing approximate tree edit distance using relaxation labeling. Pattern Recognit Lett 24(8):1089–1097
30. van Eede M, Macrini D, Telea A, Sminchisescu C, Dickinson S (2006) Canonical skeletons for shape matching. In: Proceedings of the international conference on pattern recognition, Hong Kong, pp 64–69



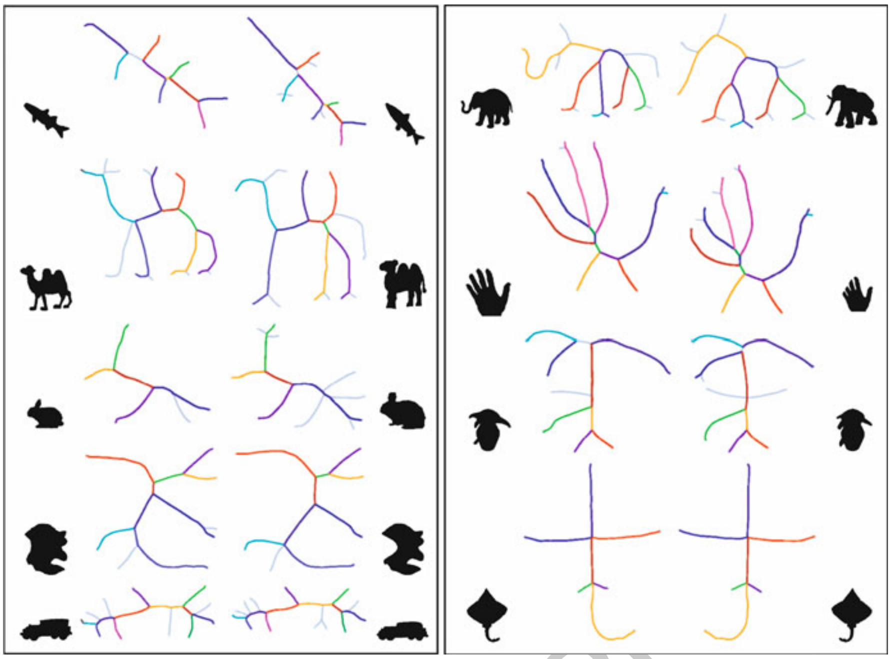
Shock Graph, Fig. 1 The shock graphs derived for two different views of a brush using the algorithm of Siddiqi et al. [24] are represented in the top row. The bottom row depicts the correspondences between nodes in the shock graphs computed by the matching algorithm

Instance	Distance to Class Prototype								
									
	$\boxed{0.02}$	2.17	4.48	3.55	2.96	0.21	4.58	14.33	10.01
	2.39	$\boxed{0.10}$	5.97	15.90	3.98	0.14	26.12	17.28	28.94
	10.89	4.72	$\boxed{2.08}$	12.24	3.12	2.15	19.73	10.11	12.64
	7.15	6.42	$\boxed{1.19}$	1.35	5.10	3.38	10.58	11.11	11.11
	4.08	7.72	2.98	$\boxed{1.49}$	4.26	4.14	26.60	13.54	14.21
	14.77	6.72	5.69	$\boxed{0.36}$	2.30	5.90	10.58	16.25	19.10
	7.86	8.90	5.94	$\boxed{0.74}$	1.59	1.10	10.81	10.39	16.08
	2.66	4.23	3.23	6.47	$\boxed{0.62}$	1.48	11.73	15.38	15.15
	3.18	5.31	1.25	4.64	$\boxed{0.60}$	1.30	14.18	17.22	9.08
	4.55	0.76	1.32	2.86	1.49	$\boxed{0.11}$	21.38	15.35	13.04
	6.77	19.46	22.11	13.27	8.21	29.50	$\boxed{0.15}$	5.12	5.03
	8.73	23.14	31.45	24.41	10.16	31.08	$\boxed{0.18}$	8.45	7.05
	12.46	19.0	27.40	14.58	24.26	17.10	8.85	$\boxed{7.49}$	16.93
	13.86	23.07	12.81	11.24	17.48	23.23	6.02	6.92	$\boxed{3.06}$
	15.73	21.28	14.10	12.46	19.56	19.21	9.53	7.12	$\boxed{5.06}$

Shock Graph, Fig. 2 Similarity between database and class prototypes computed using the algorithm of Siddiqi et al. [24]. In each row, a *box* is drawn around the most similar shape



Shock Graph, Fig. 3 The results of matching skeleton graphs for some pairs of shapes in the Rutgers Tools Database using the algorithm of Demirci et al. [9] . Corresponding segments are shown using the same color. Observe that correspondences are intuitive in all cases



Shock Graph, Fig. 4 The matching results for a few shock graphs produced by the edit-distance algorithm of Sebastian et al. [18]. Matching shock branches are shown using the same color, while the *gray* colored edges in the shock graphs indicate that they are spliced or contracted