Shock Graphs and Shape Matching

Kaleem Siddiqi† Ali Shokoufandeh§ Sven J. Dickinson§ Steven W. Zucker†

†Yale University
Center for Computational Vision & Control
{siddiqi-kaleem, zucker-steven}@cs.yale.edu

§Rutgers University
Department of Computer Science & RuCCS
{shokoufa, sven}@cs.rutgers.edu

Abstract

We have been developing a theory for the generic representation of 2-D shape, where structural descriptions are derived from the shocks (singularities) of a curve evolution process, acting on bounding contours. We now apply the theory to the problem of shape matching. The shocks are organized into a directed, acyclic **shock graph**, *and complexity is managed by attending to the most significant (central) shape components first. The space of all such graphs is highly structured and can be characterized by the rules of a* **shock graph grammar**. *The grammar permits a reduction of a shock graph to a unique rooted shock tree. We introduce a novel tree matching algorithm which finds the best set of corresponding nodes between two shock trees in polynomial time. Using a diverse database of shapes, we demonstrate our system's performance under articulation, occlusion, and changes in viewpoint.*

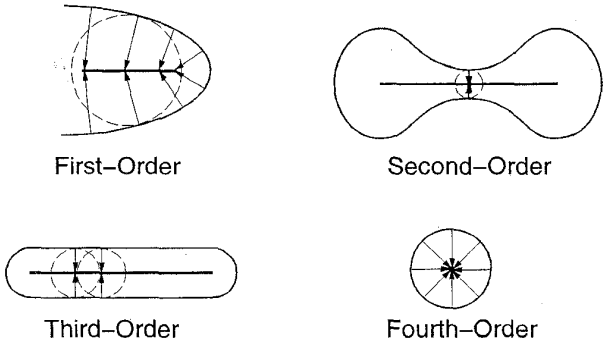

Figure 1: A coloring of shocks into four types. A 1-shock derives from a *protrusion*, and traces out a curve segment of 1-shocks. A 2-shock arises at a *neck*, and is immediately followed by two 1-shocks flowing away from it in opposite directions. 3-shocks correspond to an annihilation into a curve segment due to a *bend*, and a 4-shock an annihilation into a point or a *seed*. The loci of these shocks gives Blum's medial axis.

1 Introduction

Upon entering a room, one first notices the presence of a particular object, such as a dog, before realizing it is either a Siberian Husky or that it is "Loki", a particular Siberian. This example, modified from important studies by Rosch [15], suggests that there is an organization to our object memory, and that this organization facilitates recognition. Initially, particular instances are not recognized; rather, objects are first categorized generically at a "basic level of abstraction" [15]. The object is recognized as belonging to the category—dog—before more detailed, or subordinate levels, are refined. This motivating example is at the heart of this paper: we seek a technique for object recognition based on such entry-level, generic descriptions.

In recent work on this subject, Sclaroff and Pentland have used a modal representation corresponding to a shape's generalized axes of symmetry[16]; Zhu and Yuille have designed a 2-D shape matching system based on a decomposition into connected mid-grained skeletal parts [19]; Pauwels *et al.* have proposed the use of semi-differential invariants for planar shape recognition under affine distortions [13]; Basri *et al.* have proposed various models for measuring the cost of deforming one contour into another [3]; and François and Medioni have used a connection hierarchy of parts [6]. Whereas these efforts, together with a large body of literature on 2-D shape, have contributed a positive set of desiderata, no technique exists that satisfies all of them. Thus, we seek a representation that is viewpoint dependent to start; through which a notion of equivalence classes of (qualitatively similar) shapes emerges; that is applicable to natural as well as man-made objects; that is reliably and stably computable; and that supports efficient (e.g., polynomial-time) recognition in the presence of occlusion and noise. We build our representation on the singularities of a curve evolution process, described next.

In the application of curve evolution theory to visual

shape analysis, Kimia, Tannenbaum, and Zucker studied the following evolution equation, acting on simple closed curves in the plane [9]:

$$\begin{array}{ll} \mathcal{C}_t & = (1+\alpha\kappa)\mathcal{N} \\ \mathcal{C}(s,0) & = \mathcal{C}_0(s). \end{array} \quad (1)$$

Here $\mathcal{C}(s,t)$ is the vector of curve coordinates, $\mathcal{N}(s,t)$ is the inward normal, s is the path parameter, and t is the evolutionary time of the deformation. The constant $\alpha \geq 0$ controls the regularizing effects of curvature κ. When α is large, the equation becomes a geometric heat equation; when $\alpha = 0$, the equation is equivalent to Blum's grassfire transformation. In this paper, we shall only be interested in the latter case, under which the evolution equation is hyperbolic and *shocks* [10], or entropy-satisfying singularities, can form. Here we shall ignore the dynamics of the shock formation process, and will consider only the static picture obtained in the limit: the locus of shock positions gives Blum's medial axis. However, even in this static limit, the shocks provide additional information beyond that available from their loci: consider a "coloring" of the shocks according to the local variation of the radius function along the medial axis (see Figure 1). The colored description provides a much richer foundation for recognition than that obtained from an unlabeled (Blum) skeleton.

To illustrate the coloring, imagine traversing a path along the medial axis. At a 1-shock the radius function varies monotonically, as is the case for a protrusion. At a 2-shock the radius function achieves a strict local minimum such that the medial axis is disconnected when the shock is removed, e.g., at a neck. At a 3-shock the radius function is constant along an interval, e.g., for a bend with parallel sides.[1] Finally, at a 4-shock the radius function achieves a strict local maximum, as is the case when the evolving curve annihilates into a single point or a seed.

With the above picture in mind, the coloring can be formalized as follows. Let X be the open interior of a simple closed curve, and $Me(X)$ its medial axis (the set of points reached simultaneously by two or more fire fronts). Let $B(x,\epsilon)$ be an open disk of radius ϵ centered at $x \in X$, and let $R(x)$ denote the radius of the largest such disk contained in X. Let $N(x,\epsilon) = Me(X) \cap B(x,\epsilon) \backslash \{x\}$ define a "punctured" ϵ-neighborhood of x, one that does not contain x itself. A medial axis point $x \in Me(X)$ is

1. **type 4** if $\exists \epsilon > 0$ s.t. $R(x) > R(y)\ \forall y \in N(x,\epsilon)$;
2. **type 3** if $\exists \epsilon > 0$ s.t. $R(x) = R(y)\ \forall y \in N(x,\epsilon)$ and $N(x,\epsilon) \neq \emptyset$;
3. **type 2** if $\exists \epsilon > 0$ s.t. $R(x) < R(y)\ \forall y \in N(x,\epsilon)$ and $N(x,\epsilon) \neq \emptyset$ and $N(x,\epsilon)$ is not connected; **and**
4. a **1-shock** otherwise.

The relationship between the above coloring of the medial axis and an Arnold classification of singularities [2] remains to be investigated. In Figure 4 we provide several numerical examples of colored medial axis descriptions. As we shall now show, the *coloring* coupled with a measure of *significance* derived from the time of shock formation, is the key to abstracting a representation that supports generic shape matching.

2 The Shock Graph

We shall now abstract the system of shocks derived from the curve evolution process into a graph, which we call the *Shock Graph*, or **SG**. This construction is inspired by Blum's classic work on axis-morphologies [4]; the shock types will label each vertex in the graph and the shock formation times will direct edges to provide an ordering for matching, and a basis for subgraph approximation.

By the Jordan Curve Theorem, any simple closed curve divides the plane $\mathcal{R}^2$ into exactly two components, one bounded and the other unbounded. We are interested in the bounded interiors of Jordan curves.

Definition 1 *A 2-D shape $\mathcal{O}$ is the bounded interior of a simple closed (Jordan) curve.*

From the coloring of shocks into four types in the previous section, it can be seen that 2-shocks and 4-shocks are isolated points, whereas 1-shocks and 3-shocks are neighbored by other shocks of the same type. To build the shock graph we shall group together shocks of the same type that form a connected component, denoting the groups with labels $\tilde{1}, 2, \tilde{3}$ and 4, and breaking apart the $\tilde{1}$'s at branch-points.[2] Let each shock group be indexed by a distinct integer i, and let t_i denote its time (or times) of formation, corresponding to the radius function evaluated at the shocks in the group. Hence, t_i will be an interval for a $\tilde{1}$; for 2's, $\tilde{3}$'s and 4's it will be a single number. Finally, let # denote a *start* symbol and ϕ a *terminal* symbol. The **SG** is a connected graph, rooted at a vertex labeled #, such that all other (non-terminal) vertices are shock groups, and directed edges to non-terminal vertices indicate the genesis of new shock groups.

Definition 2 *The Shock Graph of a 2-D shape, $\mathbf{SG}(\mathcal{O})$, is a labeled graph $G = (V, E, \gamma)$, with:*

[1] This "parallel" condition reflects the non-genericity of 3-shocks.

[2] The ˜ symbol is used to denote a curve segment. A branch-point, where the maximal inscribed disc "touches" the boundary at more than two points, will be shared by all $\tilde{1}$'s that overlap at it.

- **vertices** $V = \{1, ..., n\}$;
- **edges** $(i, j) \in E \subseteq V \times V$ *directed from vertex* i *to vertex* j *if and only if* $i \neq j$, $t_i \geq t_j$, *and* $i \cup j$ *is connected in the plane;*
- **labels** $\gamma : V \rightarrow l$, *with* $l \in \{\tilde{1}, 2, \tilde{3}, 4, \#, \phi\}$; *and*
- **topology** *such that,* $\forall j \in V$ *with* $\gamma(j) \neq \#$, $\exists i \in V$ *with* $(i, j) \in E$.

The **SG** is built by "reversing" the grassfire evolution, analogous to growing a shape by adding lumps of material onto its seeds. The children of the unique vertex labeled #, at which the graph is rooted, are the last shock groups to form. Vertices with label ϕ are leaves of the **SG**, whose parents are the first shock groups to form. This reverse-time dependency is important because the last shocks to form correspond to the most significant (central) shape features.

Proposition 1 *Any 2-D shape* $\mathcal{O}$ *has a unique corresponding shock graph* $\mathbf{SG}(\mathcal{O})$.

PROOF: The proof appears in [18].

2.1 The Shock Graph Grammar

The notion of entry-level categories for shape that we seek is intimately connected to the topological structure of the shock graph. This structure is highly constrained because the events that govern the birth, combination, and death of shock groups can be abstracted into a small number of rewrite rules, shown in Figure 2. In analogy to Leyton's Process Grammar [11], the rules have been grouped according to the semantic processes that they characterize, although the alphabet of shock types that they operate on is quite different from boundary-based codons.

Definition 3 *The Shock Graph Grammar,* **SGG**, *is a quadruple* $G = (V, \Sigma, R, S)$, *with*

1. $V = \{\tilde{1}, 2, \tilde{3}, 4, \#, \Phi\}$, *the alphabet;*
2. $\Sigma = \{\Phi\}$, *the set of terminals;*
3. $S = \#$, *the start symbol; and*
4. $R = \{R_1, ..., R_{10}\}$, *the set of rules given in Figure 2.*

The rewriting system emphasizes the generative process of growing a shape by placing seeds, adding protrusions, forming unions, and so on. It operates by beginning at the start symbol and repeatedly replacing the left-hand side of a rule by the corresponding right-hand side until no further replacements can be made. It is the **SGG** that captures the beauty of shock graphs, because the rules embody constraints from the domain of curve evolution. In particular,

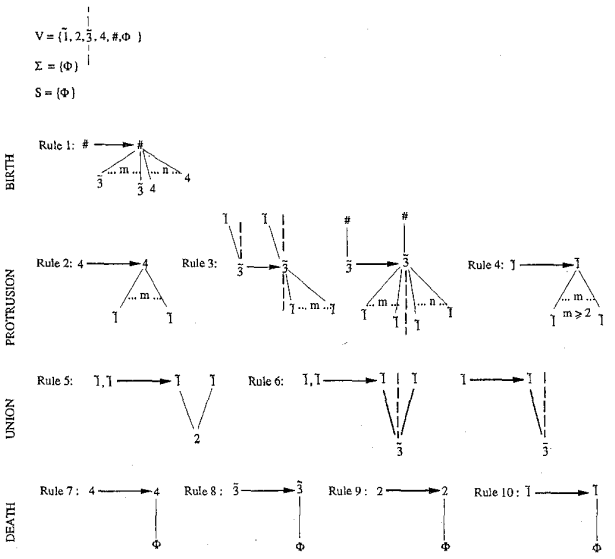

Figure 2: The Shock Graph Grammar, *SGG*. Dashed lines partition distinct ends of a $\tilde{3}$. The rules are grouped according to the different semantic processes (on the left) that they characterize. Note that the grammar is not context-free, e.g., rule 3 indicates that a $\tilde{1}$ can only be added onto an end of a $\tilde{3}$ that has no parent $\tilde{1}$.

Proposition 2 *The rewrite rules of the* **SGG** *are sufficient to derive the shock graph* $\mathbf{SG}(\mathcal{O})$ *of any 2-D shape* $\mathcal{O}$.

PROOF: A constructive proof appears in [18]. The strategy is to derive the rules by enumerating all legal parents and children for each vertex type.

We can now make several observations. First, since the same shock cannot be born at two distinct times, *the* **SG** *is a directed acyclic graph.* This has important consequences for object matching because the problem of searching directed acyclic graphs is computationally much simpler than that of searching arbitrary graphs. Second, since there exist rules in the **SGG** whose left-hand sides do not consist of single nonterminals, *the* **SGG** *is not context-free.* Third, the rewrite rules indicate that a 2-shock and a 4-shock can *only* be added by rules 5 and 1, respectively, and that equivalent rules exist for a $\tilde{3}$ (rules 6 and 1). Hence, *a 2-shock and a 4-shock are each semantically equivalent to a* $\tilde{3}$ *in a specific context.*

The **SG**'s for a variety of shapes are shown in Figure 5. All the graphs were generated *automatically* from the output of the shock detection process [17] displayed in Figure 4. Following the third observation, only label types $\tilde{1}$ and $\tilde{3}$ have been explicitly assigned. A $\tilde{3}$ with a parent $\tilde{1}$ at each end acts as a "neck", and

a $\tilde{3}$ with a # as a parent acts as a "seed". In the next Section we show that a shock graph can be reduced to a unique rooted shock tree, which in turn implies a hierarchical ordering of shape information (shock vertices). We then develop a formal approach to *significance-based matching*, where the key idea is to defeat complexity (when the database of shapes is diverse and large) by attending to most significant components first, via a depth-first search of the underlying shock trees.

3 Shock Graph Matching

3.1 Problem Formulation

Given two shock graphs, one representing an object in the scene (V_2) and one representing a database object (V_1), we seek a method for computing their similarity. Unfortunately, due to occlusion and clutter, the shock graph representing the scene object may, in fact, be embedded in a larger shock graph representing the entire scene. Thus we have a largest subgraph isomorphism problem, which can be formulated as a $\{0,1\}$ integer optimization problem. The optimal solution is a $\{0,1\}$ bijective mapping matrix M, which defines the correspondence between the vertices of the two graphs G and H, and which minimizes an appropriately defined distance measure between corresponding edge and/or node labels in the two graphs. More formally, we seek the matrix M, the global optimizer of the following [5]:

$$\begin{aligned} \min \quad & -\frac{1}{2}\sum_{u\in V_1}\sum_{v\in V_2} M(u,v)||u,v|| \\ \text{s.t.} \quad & \sum_{u'\in V_2} M(u,u') \leq 1,\ \forall u \in V_1 \\ & \sum_{v\in V_1} M(v,v') \leq 1,\ \forall v' \in V_2 \\ & M(x,y) \in \{0,1\}, \forall x \in V_1,\ y \in V_2 \end{aligned} \tag{2}$$

where $||.||$ is a measure of the similarity between the labels of corresponding nodes in two shock graphs (see Section 3.4).

The above minimization problem is known to be NP-hard for general graphs [8]. However, polynomial time algorithms exist for finite rooted trees, e.g., see [14]. In fact, based on the grammar in Figure 2, it is easy to show that the shock graph can be reduced to a unique rooted tree [18]. Hence, we can purse a polynomial time solution to the problem of matching shock trees.

3.2 An Eigenvalue Characterization of a Shock Tree

The shock tree can be represented as a $\{0,1\}$ adjacency matrix, with 1's indicating adjacent nodes in the tree. Any shock subtree therefore defines a submatrix of the adjacency matrix. If, for a given shock subtree, we compute the eigenvalues of its corresponding submatrix, then the sum of the eigenvalues is invariant to any similarity transformation applied to the submatrix. This means that the eigenvalue sum is invariant to any consistent re-ordering of the subtrees! In terms of our largest subgraph isomorphism problem, finding the two shock subtrees whose eigenvalue sums are closest represents an approximation to finding the largest isomorphic subtrees.[3]

In order to efficiently compute the submatrix eigenvalue sums, we turn to the domain of semidefinite programming. A symmetric $n \times n$ matrix A with real entries is said to be positive semidefinite, denoted as $A \succeq 0$, if for all vectors $x \in R^n$, $x^t A x \geq 0$, or equivalently, all its eigenvalues are non-negative. We say that $U \succeq V$ if the matrix $U - V$ is positive semidefinite. For any two matrices U and V having the same dimensions, we define $U \bullet V$ as their inner product, i.e., $U \bullet V = \sum_i \sum_j U_{i,j} V_{i,j}$. For any square matrix U, we define $\text{trace}(U) = \sum_i U_{i,i}$. Let I denote the identity matrix having suitable dimensions. The following result, due to Overton and Womersley [12], will characterize the sum of the first k largest eigenvalues of a symmetric matrix in the form of a semidefinite convex programming problem:

Theorem 1 *For the sum of the first k eigenvalues of a symmetric matrix A, the following semidefinite programming characterization holds:*

$$\begin{aligned} \lambda_1(A) + \ldots + \lambda_k(A) = \quad & \max \quad A \bullet U \\ & \text{s.t.} \quad \text{trace}(U) = k \\ & \qquad 0 \preceq U \preceq I. \end{aligned}$$

Before applying the above theorem, we must first convert our shock trees to adjacency matrices. Given a bounded degree, rooted tree $G = (V, E)$ with $|V| = n$ and $|E| = m$, we define the adjacency matrix A of G to be a $n \times n$ symmetric, $\{0,1\}$ matrix with its (i,j)-th entry $A_{i,j}$ equal to 1 if $(i,j) \in E$, and 0 otherwise. For each vertex $v \in G$, let $\delta(v)$ be the degree of v, and let $\delta(G)$ be the maximum degree over all vertices in G. For every vertex $u \in G$, we define $\chi(u)$ to be a vector in $R^{\delta(G)-1}$, obtained through the following procedure:

> For any child v of u in G, construct the adjacency matrix A_v of the induced subtree rooted at v, and for A_v, compute the quantity $\lambda_v = \lambda_1(A_v) + \ldots + \lambda_{\delta(v)}(A_v)$. Construct $\chi(u)$ as the vector formed by $\{\lambda_{v_1}, \ldots, \lambda_{v_{\delta(u)}}\}$ for which $\lambda_{v_1} \geq \ldots \geq \lambda_{v_{\delta(u)}}$.

[3] This analysis considers only the topological structure of the shock graph. Later, we will factor in geometric information associated with its vertices.

The above procedure yields a vector assigned to each vertex, whose elements are the individual eigenvalue sums corresponding to the node's (subtree's) adjacency submatrix. The power of this formulation is that for any rooted subtree, the vector coloring of the vertices is uniquely defined, and is invariant to a re-ordering of the subtrees rooted at each vertex. Furthermore, the λ_v function can be computed in polynomial time by solving the the equivalent semidefinite programming problem (Theorem 1) using a variant of the Interior Point method proposed by Alizadeh [1]. In section 3.4, we embed this procedure in our own algorithm for finding the largest isomorphic subtrees corresponding to two shock graphs. In addition, we factor in a measure of similarity between shock geometries, which we now discuss.

3.3 The Distance Between Two Vertices

The eigenvalue characterization introduced above applies to the problem of determining the topological similarity between two shock trees. Returning to the opening scenario, this, roughly speaking, defines an equivalence class of objects belonging to the same entry-level category. For example, a broad range of dogs will have very similar shock tree structures. On the other hand, when one is interested in discriminating between a short-legged Daschund and a Siberian Husky, geometric properties will play a significant role.

This geometry is encoded by information contained in each vertex of the shock tree. Specifically, recall that both $\tilde{1}$'s and $\tilde{3}$'s are curve segments of shocks. In the former case, the segment is directed, while in the latter there is a partial order but no preferred direction, since all the shocks were formed at the same time. Each shock in a segment is further labeled by its position, its time of formation (radius of the skeleton), and its direction of flow (or orientation in the case of $\tilde{3}$'s), all obtained from the shock detection algorithm [17]. In order to measure the similarity between two vertices u and v, we interpolate a low dimensional curve through their respective shock trajectories, and assign a cost $C(u,v)$ to an affine transformation that aligns one interpolated curve with the other. The technical details are presented in [18]. Intuitively, a low cost is assigned if the underlying structures are scaled or rotated versions of one another.

3.4 Algorithm for Matching Two Shock Trees

Our recursive algorithm for matching the rooted shock subtrees, G and H, accounts for both the topological similarity of the subtrees as well as the geometrical (shock) similarity of their corresponding nodes. Before stating our algorithm, inspired by Reyner [14], some definitions are in order. Let $G = (V_1, E_1)$ and $H = (V_2, E_2)$ be the two shock graphs to be matched, with $|V_1| = n_1$ and $|V_2| = n_2$. Define d to be the maximum degree of any vertex in G and H, i.e., $d = \max(\delta(G), \delta(H))$. For each vertex v, we define $\chi(v) \in R^{d-1}$ as the unique eigen-decomposition vector introduced in Section 3.2.[4] Furthermore, for any pair of vertices u and v, let $C(u,v)$ denote the shock distance between u and v, as described in Section 3.3. Finally, let $\Phi(G,H)$ (initially empty) be the set of final node correspondences between G and H representing the solution to our matching problem.

The algorithm begins by forming a $n_1 \times n_2$ matrix $\Pi(G,H)$ whose (u,v)-th entry has the value $C(u,v)||\chi(u)-\chi(v)||_2$, assuming that u and v are compatible in terms of their shock order, and has the value ∞ otherwise. Next, we form a bipartite edge weighted graph $\mathcal{G}(V_1, V_2, E_\mathcal{G})$ with edge weights from the matrix $\Pi(G,H)$.[5] Using the scaling algorithm of Goemans, Gabow, and Williamson [7], we then find the maximum cardinality, minimum weight matching in $\mathcal{G}$. This results in a list of node correspondences between G and H, called $\mathcal{M}_1$, that can be ranked in decreasing order of similarity.

From $\mathcal{M}_1$, we choose (u_1, v_1) as the pair that has the minimum weight among all the pairs in $\mathcal{M}_1$, i.e., the first pair in $\mathcal{M}_1$. (u_1, v_1) is removed from the list and added to the solution set $\Phi(G,H)$, and the remainder of the list is *discarded*. For the subtrees G_{u_1} and H_{v_1} of G and H, rooted at nodes u_1 and v_1, respectively, we form the matrix $\Pi(G_{u_1}, H_{v_1})$ using the same procedure described above. We then find the matching $\mathcal{M}_2$ in the bipartite graph defined by weight matrix $\Pi(G_{u_1}, H_{v_1})$, yielding another ordered list of node correspondences. The procedure is recursively applied to (u_2, v_2), the edge with minimum weight in $\mathcal{M}_2$, with the remainder of the list discarded.

This recursive process eventually reaches the leaves of the subtrees, forming a list of ordered correspondence lists (or matchings) $\{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$. In backtracking step i, we remove any subtrees from the graphs G_i and H_i whose roots participate in a matching pair in $\Phi(G,H)$ (we enforce a one-to-one correspondence of nodes in the solution set). Then, in a depth-first manner, we first recompute $\mathcal{M}_i$ on the subtrees rooted at u_i and v_i (with solution set nodes removed). As before, we choose the minimum weight matching pair, and recursively descend. Unlike a traditional depth-first search, we dynamically recompute the branches at each node

[4] Note that if the maximum degree of a node is d, then excluding the edge from the node's parent, the maximum number of children is $d-1$. Also note that if $\delta(v) < d$, then then the last $d - \delta(v)$ entries of χ are set to zero to ensure that all χ vectors have the same dimension.

[5] $G(A,B,E)$ is a weighted bipartite graph with weight matrix $W = [w_{ij}]$ of size $|A| \times |B|$ if, for all edges of the form $(i,j) \in E$, $i \in A$, $j \in B$, and (i,j) has an associated weight $= w_{i,j}$.

in the search tree. Processing at a particular node will terminate when either subtree loses all of its nodes to the solution set.

We can now state the algorithm more precisely:

procedure isomorphism(G,H)
  $\Phi(G,H) \leftarrow \emptyset$
  $d \leftarrow \max(\delta(G), \delta(H))$
  for $u \in V_G$ compute $\chi(u) \in R^{d-1}$ (see Section 3.2)
  for $v \in V_H$ compute $\chi(v) \in R^{d-1}$ (see Section 3.2)
  call match(root(G),root(H))
  return(cost($\Phi(G,H)$))
end

procedure match(u,v)
  do
    {
    let $G_u \leftarrow$ rooted subtree of G at u
    let $H_v \leftarrow$ rooted subtree of H at v
    compute $|V_{G_u}| \times |V_{H_v}|$ weight matrix $\Pi(G_u, H_v)$
    $\mathcal{M} \leftarrow$ max cardinality, minimum weight
        bipartite matching in $\mathcal{G}(V_{G_u}, V_{H_v})$ with
        weights from $\Pi(G_u, H_v)$ (see [7])
    $(u', v') \leftarrow$ minimum weight pair in M
    $\Phi(G,H) \leftarrow \Phi(G,H) \cup \{(u', v')\}$
    call match(u',v')
    $G_u \leftarrow G_u - \{x | x \in V_{G_u}$ and $(x,w) \in \Phi(G,H)\}$
    $H_v \leftarrow H_v - \{y | y \in V_{H_v}$ and $(w,y) \in \Phi(G,H)\}$
    }
  while ($G_u \neq \emptyset$ and $H_v \neq \emptyset$)

In terms of algorithmic complexity, observe that during the depth-first construction of the matching chains, each vertex in G or H will be matched at most once in the forward procedure. Once a vertex is mapped, it will never participate in another mapping again. The total time complexity of constructing the matching chains is therefore bounded by $O(n^2\sqrt{n \log\log n})$, for $n = \max(n1, n2)$ [7]. Moreover, the construction of the $\chi(v)$ vectors will take $O(n\sqrt{n}L)$ time, implying that the overall complexity of the algorithm is $\max(O(n^2\sqrt{n \log\log n}), O(n^2\sqrt{n}L)$.

The above algorithm provides, in polynomial time better than $O(n^3)$, an approximate optimal solution to the minimization problem in 2. The matching matrix M in (2) can be constructed using the mapping set $\Phi(G,H)$. Our algorithm is particularly well-suited to the task of matching two shock trees since it can find the best correspondence in the presence of occlusion and/or noise in the tree.

4 Examples

We demonstrate our shape matching system with several examples. The database of shapes we used is shown in Figure 3. The shock-based descriptions of representative shapes, numerically computed using the

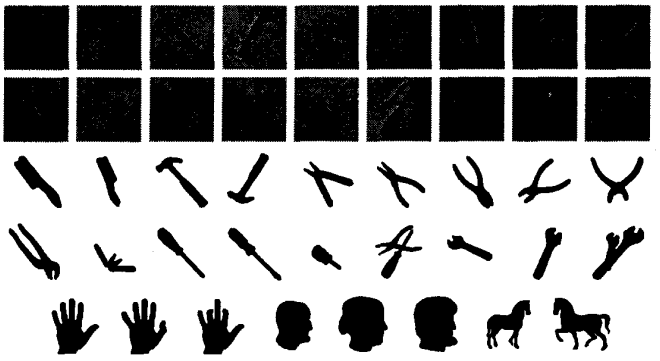

Figure 3: TOP: The 18 tool images used in our experiments. MIDDLE: The silhouettes were segmented using a curve evolution based active contour. BOTTOM: The database was supplemented with 8 biological shapes, adapted from a range image of a hand and various da Vinci sketches.

algorithms developed in [17], are shown in Figure 4, with the derived shock graphs in Figure 5. Notice how for each shape a hierarchy of components emerges, with the most significant components (e.g., the palm of the hand, and the neck of the pliers) placed closest to the root node. Similar descriptions were computed for each of the shapes in the database.

To evaluate our matcher's ability to compare objects based on their protypical or coarse shape, we chose as a prototype for each of our 9 object classes, that object whose total distance to the other members of its class was a minimum.[6] We then computed the similarity between each remaining object in the database and each of the class prototypes, with the results shown in Figure 6.

For each row in the table, a box has been placed around the most similar shape. We note that for the 15 test shapes, all but two are most similar to their class prototype, with the class prototype coming in a close second in the latter two cases. The recovered correspondences between nodes for the best matches in rows 4 and 15 are shown in Figure 7. Three very powerful features of our system our worth highlighting. First, the method is truly generic: the matching scores impose a partial ordering in each row, which reflects the qualitative similarity between structurally similar shapes. An increase in structural complexity is reflected in a higher cost for the best match, e.g., in the bottom two rows of Figure 6. Second, the procedure is designed to handle noise or occlusion, manifest as missing or additional vertices in the shock graph. Third, the depth-first search through subtrees is extremely efficient.

[6] For each of the three classes having only two members, the class prototype was chosen at random.

Query	Distance to Class Prototype								
	0.02	2.17	4.48	3.55	2.96	0.21	4.58	14.33	10.01
	2.39	0.10	5.97	15.90	3.98	0.14	26.12	17.28	28.94
	10.89	4.72	2.08	12.24	3.12	2.15	19.73	10.11	12.64
	7.15	6.42	1.19	1.35	5.10	3.38	10.58	11.11	11.11
	4.08	7.72	2.98	1.49	4.26	4.14	26.60	13.54	14.21
	14.77	6.72	5.69	0.36	2.30	5.90	10.58	16.25	19.10
	7.86	8.90	5.94	0.74	1.59	1.10	10.81	10.39	16.08
	2.66	4.23	3.23	6.47	0.62	1.48	11.73	15.38	15.15
	3.18	5.31	1.25	4.64	0.60	1.30	14.18	17.22	9.08
	4.55	0.76	1.32	2.86	1.49	0.11	21.38	15.35	13.04
	6.77	19.46	22.11	13.27	8.21	29.50	0.15	5.12	5.03
	8.73	23.14	31.45	24.41	10.16	31.08	0.18	8.45	7.05
	12.46	19.0	27.40	14.58	24.26	17.10	8.85	7.49	6.93
	13.86	23.07	12.81	11.24	17.48	23.23	6.02	6.92	3.06
	15.73	21.28	14.10	12.46	19.56	19.21	9.53	7.12	5.06

Figure 6: Experiment 1: similarity between database shapes and class prototypes. In each row, a box is drawn around the most similar shape (see the text for a discussion).

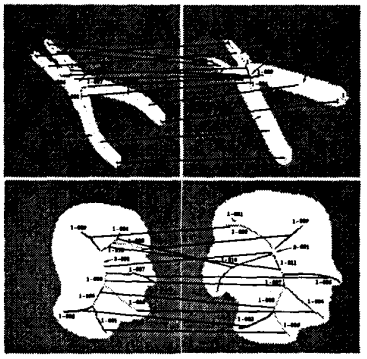

Figure 7: The computed correspondences between nodes for the best matches in rows 4 (top) and 15 (bottom) of Figure 6.

Query	Distance to Class Exemplars			
	8.89	0.38	4.70	5.94
	1.49	0.36	1.35	0.89
	5.21	9.95	0.57	10.01
	1.17	7.02	2.91	2.08

Table 1: Experiment 2: similarity between members of a class. Each row of the table highlights different aspects of matching invariance (in addition to translation): Rows 1 and 2: invariance to deformation, image rotation, and illumination; Row 3: invariance to deformation, scaling, and occlusion; and Row 4: invariance to deformation, scaling, image rotation, and illumination.

Query	Distance to Class Exemplars			
	0.09	0.29	0.16	5.42
	1.48	1.30	3.46	0.11
	6.53	0.79	5.24	0.37

Table 2: Experiment 3: similarity between members of a class. Each row of the table highlights different aspects of matching invariance (in addition to translation): Row 1: invariance to scaling, deformation (different taper), and occlusion; Row 2: invariance to scaling, image rotation, and slight rotation in depth; and Row 3: invariance to image rotation, scaling, and occlusion.

In Tables 1 and 2, we compare a number of objects to other members of their class as well as to a member from a different class. The objects have been chosen to illustrate the power of the matcher to deal with changes in image plane rotation, scale, deformation, occlusion, translation, and even slight rotation in depth. In both experiments, the results reflect the matcher's ability to compare shapes within the same class, at a finer scale.

5 Conclusions

In this paper, we have abstracted a representation of shape based on singularities of a curve evolution process into a shock graph, and have introduced a shock graph grammar to characterize its structure. We have developed a shape matching algorithm that is *generic* and provides a powerful means for efficiently computing the best correspondence between two shock graphs in the presence of noise and occlusion, as illustated by several examples. In future work, we shall address the problem of indexing 2-D objects in a large database. Using a vector of eigenvalue sums computed on the subtrees of a shock tree, similar subtrees can be retrieved from a database via a simple vector norm. Further, building on ideas from aspect graphs we plan to extend our approach to a view-based strategy for generic 3-D object recognition. The intuitive idea is to concatenate the shock graphs associated with a collection of sufficiently distinct projected views of an object, and then use a similar matching algorithm. Whereas much work remains to be done on this front, empirical evidence

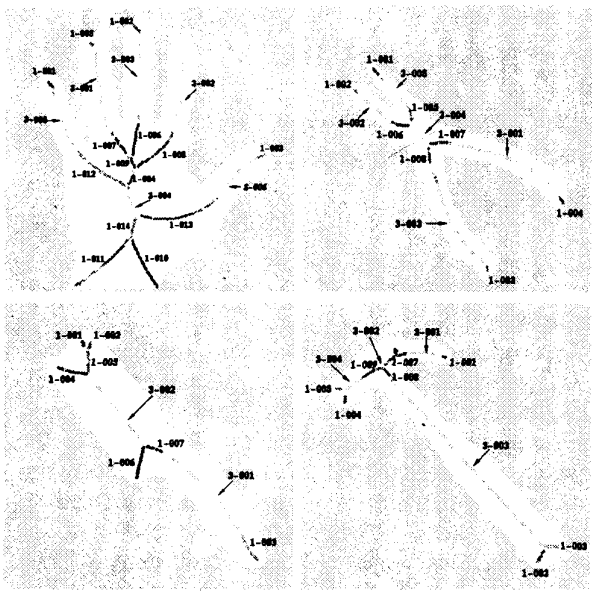

Figure 4: The shocks computed for a hand, a plier, a brush, and a hammer. The labels correspond to vertices in the derived shock graphs, as shown in Figure 5.

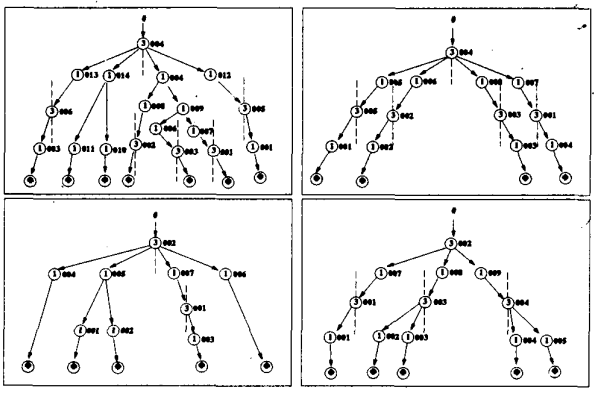

Figure 5: The shock graphs for the shapes in Figure 4. The vertices are labeled according to their type, with the arrows in the direction of shape growth. The distinct ends of a 3-shock are partitioned with a dashed line.

indicates that the topological structure of the shock graph is quite stable under small changes in viewpoint.

Acknowledgements We thank Jonas August for many helpful discussions. Kaleem Siddiqi and Steven Zucker were supported by grants from AFOSR, NSERC, and NSF. Sven Dickinson gratefully acknowledges the support of NSF CAREER grant IRI-9623913.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.

[2] V. Arnold. *The Theory of Singularities and Its Applications.* Lezioni Fermiane, Piza, Italy, 1991.

[3] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. In *Proceedings, ICCV Workshop on Physics-Based Modeling in Computer Vision*, pages 135–143, 1995.

[4] H. Blum. Biological shape and visual science. *J. Theor. Biol.*, 38:205–287, 1973.

[5] G. G. E. Mjolsness and P. Anandan. Optimization in model matching and perceptual organization. *Neural Computation*, 1:218–229, 1989.

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

[7] H. N. Gabow, M. X. Goemans, and D. P. Williamson. An efficient approximate algorithm for survivable network design problems. *Proc. of the Third MPS Conference on Integer Programming and Combinatorial Optimization*, pages 57–74, 1993.

[8] M. Garey and D. Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[9] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Shape, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. *IJCV*, 15:189–224, 1995.

[10] P. D. Lax. Shock waves and entropy. In E. H. Zarantonello, editor, *Contributions to Nonlinear Functional Analysis*, pages 603–634, New York, 1971. Acad. Press.

[11] M. Leyton. A process grammar for shape. *Artificial Intelligence*, 34:213–247, 1988.

[12] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Math. Programming*, 62(2):321–357, 1993.

[13] E. Pauwels, T. Moons, L. J. V. Gool, P. Kempenaers, and A. Oosterlinck. Recognition of planar shapes under affine distortion. *IJCV*, 14(1):49–65, January 1995.

[14] S. W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM J. Comput.*, 6:730–732, 1977.

[15] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8:382–439, 1976.

[16] S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE PAMI*, 17(6):545–561, June 1995.

[17] K. Siddiqi and B. B. Kimia. A shock grammar for recognition. Technical Report LEMS 143, LEMS, Brown University, September 1995.

[18] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. Technical report, Yale/Rutgers, Sept. 1997.

[19] S. Zhu and A. L. Yuille. Forms: a flexible object recognition and modelling system. *IJCV*, 20(3):187–212, 1996.