Recognition of Shapes by Editing Shock Graphs

Thomas B. Sebastian
Div. of Engineering
Brown University
Providence RI 02912
tbs@lems.brown.edu

Philip N. Klein
Dept. of Computer Science
Brown University
Providence RI 02912
klein@cs.brown.edu

Benjamin B. Kimia
Div. of Engineering
Brown University
Providence RI 02912
kimia@lems.brown.edu

Abstract

This paper presents a novel recognition framework which is based on matching shock graphs of 2D shape outlines, where the distance between two shapes is defined to be the cost of the least action path deforming one shape to another. Three key ideas render the implementation of this framework practical. First, the shape space is partitioned by defining an equivalence class on shapes, where two shapes with the same shock graph topology are considered to be equivalent. Second, the space of deformations is discretized by defining all deformations with the same sequence of shock graph transitions as equivalent. Shock transitions are points along the deformation where the shock graph topology changes. Third, we employ a graph edit distance algorithm that searches in the space of all possible transition sequences and finds the globally optimal sequence in polynomial time. The effectiveness of the proposed technique in the presence of a variety of visual transformations including occlusion, articulation and deformation of parts, shadow and highlights, viewpoint variation, and boundary perturbations is demonstrated. Indexing into two separate databases of roughly 100 shapes results in 100% accuracy for top three matches and 99.5% for the next three matches.

1 Introduction

We present a novel approach to object recognition that is based on finding the best deformation of one shape to another, and defining the distance between two shapes as the extent of this deformation. There has been extensive work in the area of shape comparison for object recognition and it is well known that the underlying representation of the shape of objects can have a significant impact on the effectiveness of a recognition strategy. Shapes have been represented by their outline curves [1, 3, 14, 21], point sets, feature sets [2, 5], and by their medial axis [11, 13, 15, 18, 23], among others.

In curve-based representations, the matching is typically based on either aligning feature points by an optimal similarity transformation, or by finding a mapping from one
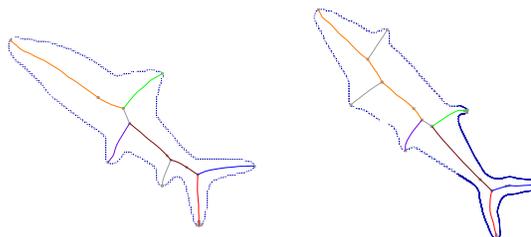

Figure 1: The matching result between two fishes based on their shock graphs. Same colors indicate matching shock branches, and grey colored branches in the shock graphs have been spliced. This color scheme is used throughout the paper.

curve to the other that minimizes an "elastic" performance functional, penalizing the "stretching" and "bending" energies of deformation [1, 3, 14, 21]. The curve outline-based matching methods often suffer from one or more of the following drawbacks: asymmetric treatment of the two curves, sensitivity to sampling, lack of rotation and scaling invariance, and sensitivity to articulations and deformations of parts.

Another type of shape representation models the shape outline as a point set and matches the point set using an assignment algorithm. Gold *et al.* [5] use graduated assignment to match image boundary features. In a recent approach, Belongie *et al.* [2] use the Hungarian method to match unordered boundary points, using a coarse histogram of the relative location of the remaining points as features. These methods have the advantage of not requiring ordered boundary points, but the match does not necessarily preserve the coherence of shapes in that the relationship among portions of shape in the process of matching may not be preserved.

Shapes have also been represented by their medial axis which can then be used for matching. Zhu and Yuille [23] have proposed a framework (FORMS) for matching animate shapes by comparing their skeletal graphs using a branch and bound strategy. The inherent instabilities of their skeletal representation are accounted for by using an a priori defined model graph. The applicability of their
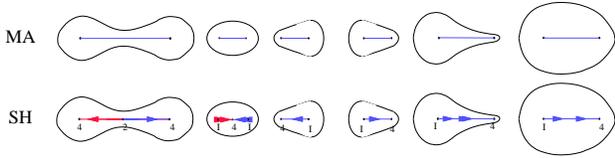
Figure 2: The dynamic interpretation of a medial axis as a flowing singularity (shock) gives rise to a more refined partitioning.

approach to inanimate objects is limited due to the choice of primitives used in modeling. Liu and Geiger [11] use the A* algorithm to match shape axis trees, which are defined by the locus of midpoints of optimally corresponding boundary points. They deal with articulations and occlusion by allowing graph topology changing operations. However, their algorithm does not preserve the ordering of edges at nodes, which can result in matches that do not preserve the coherence of the shape. In addition, the applicability of this method to large datasets of shapes is yet to be established.

A variant of the medial axis is the shock structure which is obtained by viewing the medial axis as the locus of singularities (shocks) formed in the course of wave propagation (grass-fire) from boundaries [7, 17, 19]. This dynamic interpretation of the shock trajectory associates a direction of flow, and an instantaneous velocity to each shock point, Figure 1a. The shock segments, which are defined as medial axis segments with monotonic flow, give a more refined partition of the medial axis. The resulting shock graph is a richer descriptor of shape than the medial axis graph, since its graph topology is more in accord with our perceptual notions of shape [16].

Several recognition approaches have been based on comparing shock graphs. Sharvit *et al.* [15, 6] considered pairwise assignment of shock graph nodes and the cumulative similarity of nodes and links in a graduated assignment approach [5] to find the optimal match. While this leads to fairly good matches, the errors point out a fundamental flaw in that coherence of a shape is not necessarily preserved in the match process, *e.g.*, the hierarchical relationships among parts of the shape can at times be violated. Siddiqi *et al.* convert the shock graphs to rooted trees and match them based on subgraph isomorphism [18], or by finding maximal cliques [13], which perform well in some shape matching and indexing tasks. However, the choice of the oldest shock as the root of the tree is arbitrary and can lead to erroneous matches. In addition, all the above shock graph matching methods [6, 15, 13, 18] do not explicitly model the instabilities of the symmetry-based representations, which can be problematic when dealing with visual transformations like occlusion, view point variation, and articulation.

We now present a brief overview of the proposed approach for comparing 2D shapes which addresses some of these issues. The main idea is to treat each shape as a point in a shape space and define the distance between shapes as the minimum cost of the deformation path connecting
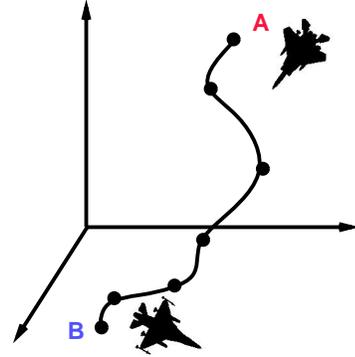

Figure 3: Every pair of shapes in the shape space is related by an infinite number of deformation paths, one of which is shown here. Each deformation path continuously deforms shape A to shape B, but can be effectively characterized by a sequence of transitions (dots).

one shape to the other. Since, there are an infinite number of dimensions (and infinite extents along each) in which a shape can be deformed, the space of shapes and of deformations must be partitioned to reduce the dimensionality of the search to practical limits. We address this issue by first defining an equivalence relation where all shapes with the same shock graph topology are equivalent, and then by defining as equivalent all the deformation paths which have the same set of transition points (boundaries between shape equivalence classes), Figure 3. These shock transitions have been formally classified recently and their complete list is now known [4]. Each transition in the graph domain is represented by an "edit" operation on the shock graph. A cost is associated with each edit operation and individual edit costs are summed to represent the cost of a deformation sequence. A graph edit distance algorithm was developed in [8, 9] based on string edit distance [20] and traditional tree edit distance [22]. The algorithm finds the globally optimal path in polynomial time which allows for a practical implementation of this framework.

2 Partitioning the Shape Space

The key bottleneck in a deformation-based approach to shape matching is the high-dimensionality of the underlying space of deformations. We define equivalence classes of shapes and deformations to reduce the dimensionality such that it is practical to search this space for an optimal deformation path between two shapes. First, observe that generally as a shape is deformed, the shock graph topology remains unaltered, but only the attributes of the graph, *e.g.*, curvature and acceleration functions stored in links, are altered, Figure 4a. However, while this describes a local neighborhood for most shapes, Figure 4a, near certain "transition" shapes, an infinitesimal change in the shape can cause a large (abrupt) change in the shock graph topology, Figure 4b. These shapes are precisely the instabilities of
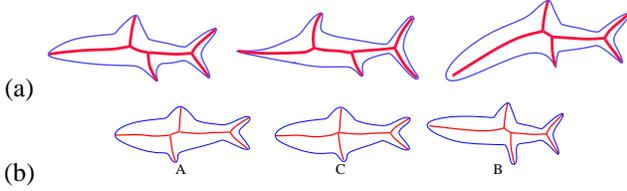
Figure 4: (a) A few examples of shapes belonging to the same shape cell, *i.e.*, having the same shock graph topology. (b) This figure illustrates one of the transitions (contract). A and B are the original shapes, and the C is the degenerate shape at the transition.
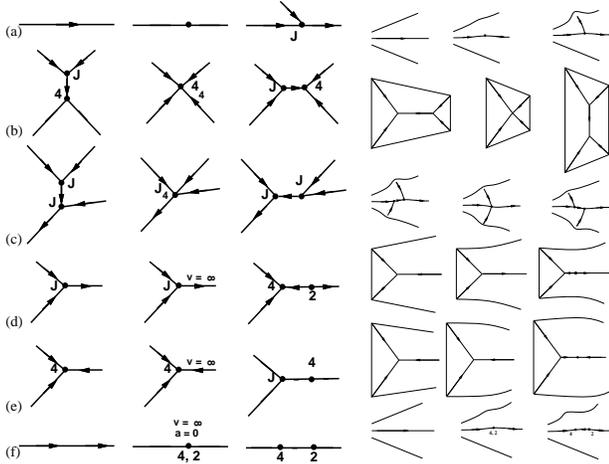

Figure 5: The first three columns show a schematic description of the six possible transitions of the shock structure, while the last three columns illustrate corresponding examples of shape deformations [4]. The central column represents the transition in the deformation from the left to right, or right to left columns. In the notation of [4] *(a)* is the $A_1 A_3$ transition, *(b)* and *(c)* are the two types of A_1^4, *(d)* and *(e)* are the two types of A_1^3 with infinite velocity, and *(f)* is the A_1^2 point with infinite velocity and zero acceleration. The graph operations to make the right and left columns equivalent are: splice, contract (two types), and merge (three types).

the medial axis/shock graph. All generic shock transitions have been formally enumerated and classified along a one-parameter family of deformations [4], Figure 5. At these transitions the local topology breaks down, creating gaps at the transition shape when using the skeletal representation, Figure 6a. This motivates an explicit embedding of the transitions, the "seams" of the shape space, in the definition of the notion of a neighborhood:

Definition 1 *A* shape cell *is a collection of shapes which have identical shock graph topology.*

Definition 2 *A* shape deformation bundle *is the set of one-parameter families of deformations passing through an identical sequence of shock transitions.*

Observe that this equivalence relation partitions the shape space. The grouping of a set of deformation paths into a
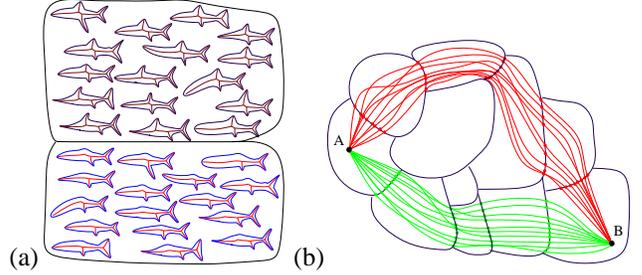

Figure 6: (a) Gaps in the topology of the shape space created by shock transitions. (b) This figure sketches two deformation path bundles between shapes A and B. Each of the two groups of deformations represent two distinct bundles of equivalent deformations.
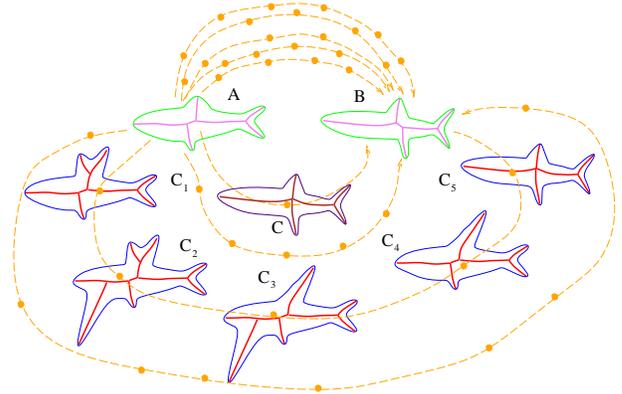
Figure 7: A deformation path where features are first added and then removed is clearly not an optimal path and need not be considered. In order to avoid such paths, we consider each deformation path as a pair of deformation paths from A and B leading to a common shape C where each sequence "simplifies" the shape.

bundle, Figure 6b, is a discretization of a rather high dimensional space of shapes and their deformation.

We note further that not all deformations paths are of interest to recognition. Consider the deformation path relating one shape to another in Figure 7. Clearly, this deformation sequence cannot be the optimal sequence since features are created and then removed. Thus, the collection of candidate paths should exclude those which unnecessarily venture into more complicated shapes. We ensure this by considering a deformation path from A and B to a "simpler" common shape C, Figure 8a, thus ensuring that no unnecessary complications arise.

The notion of simplicity used in this framework is derived from the transitions themselves. Note that a deformation from a shape to its neighboring transition shape restricts the shape by simplifying it: the first transition splices off branches, the second and third transition "symmetrize" the shape, the fourth transition removes parts at "necks", the fifth transition moves the shape to a rounder shape, and the last transition removes wavy patterns on the boundary. The main point is that a move from each shape to a neighboring transition shape removes one degree of freedom and leads
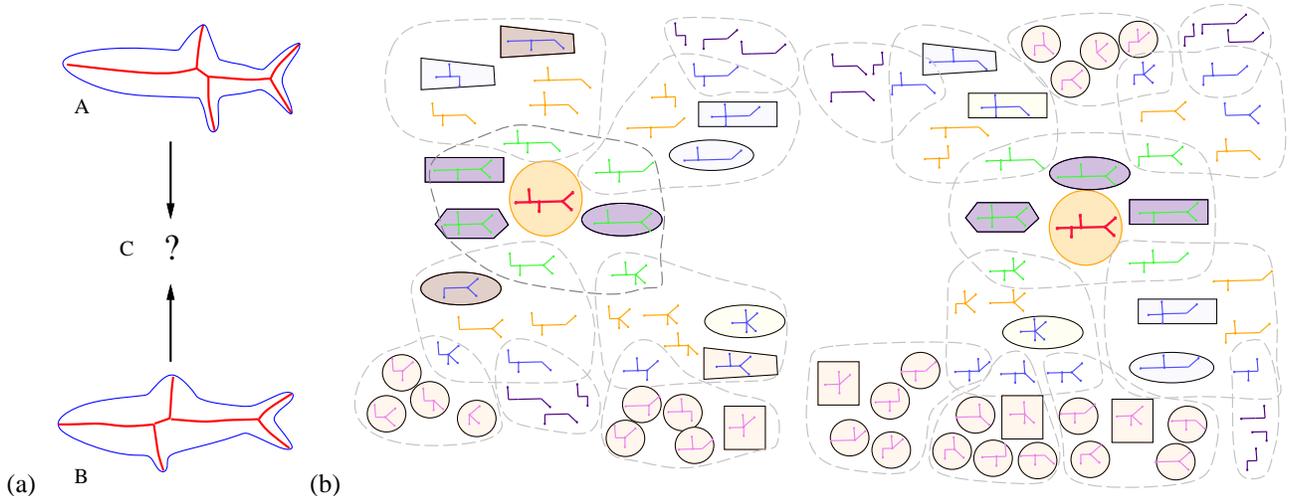
Figure 8: (a) The optimal path is obtained by searching all pairs of simplifying deformation paths leading to a common shape C. (b) A hand-drawn sketch of how the space of one-parameter family of deformations for each shape can be discretized using transition shapes. Each deformation segment is captured by a graph operation on the shock graph, namely, the "edit operation". Shape A the fish on the left (red) initially gives rise to seven shapes, each of which give rise to a finite number of other shapes enclosed in the dashed curves. A similar process applies to the second shape on the right. Common shapes with equivalent shock graph topology are detected, and marked by common icons with common shape/coloring to indicate the identification of a full deformation paths. The optimal path is the path with the least cost, which in this case, goes through the grey-hashed hexagonal surround.

to a shape cell of lower dimensionality. As this process is repeated, each shape is increasingly "simplified" until it resembles an elongated blob and finally a circle, Figure 8b. These transition shapes are reached by applying a shock graph operation, which we refer to as an "edit operation" in the graph domain.

Note that deforming the shapes A and B along all one-parameter families of "simplifying" deformations can only lead to a finite number of shapes indicated by the finite number of applicable transitions at each stage, $e.g.$, seven for the first application of transitions on the left fish in Figure 8b. In a second phase of the transition sequence, each of the seven shapes undergoes a similar application of deformations towards the next applicable transition. In a few steps, the shock graph reaches highly simplified shapes, $e.g.$, elongated approximations of the object (not shown). A complete path consists of a pair of deformation paths to a common shape cell.

3 Edit Distance Algorithm

Despite the above discretization and a tremendous reduction in dimensionality, Figure 8 illustrates that numerous paths remain to be considered, thus requiring an efficient algorithm to seek the optimal path among all possibilities. We have developed a polynomial time edit distance algorithm [8, 9] for comparing unrooted trees, which we review here. The notion of edit-distance was originally proposed to compare character strings [20], and applied to other domains, including some in computer vision [12]. In string matching applications there are three kinds of edit opera-

tions: deleting a character, inserting a character, and changing one character into another. Once costs are assigned to each edit operation, edit distance is defined to be the minimum cost of the sequence of operations required to convert one string to another, and is typically computed using dynamic programming. The notion of edit distance has been generalized to comparing ordered, rooted trees [22]. The edit operations for comparing trees are typically defined as: (i) *change* the label of an edge, (ii) *contract* an edge, and (iii) the inverse operation *uncontract* an edge. These are the natural edit operations in the domain of trees.

In applying the edit distance approach to comparing shock graphs, we derive the edit operations from the instabilities of the shock graph or shock transitions, Figure 5, which lead to four groups of edit operations: (i) the *splice* operation deletes a shock branch and merges the remaining two; (ii) the *contract* operations deletes a shock branch connecting two degree-three nodes; (iii) the *merge* operations combines two branches at a degree-two node; (iv) we also define a *deform* edit to relate two shapes in the same shape cell, $i.e.$, shapes with the same shock graph topology but with different attributes. We have developed a polynomial time algorithm to find the modified edit distance between two unrooted trees, and find the globally optimal sequence of transitions between two shock graphs. The algorithm details are presented in [8, 9].

A critical issue is how the costs are assigned to each operation in the modified edit distance in a manner that is not inconsistent with perceptual metrics of similarity. The basic approach is to first derive the cost of the deform edit, $i.e.$,
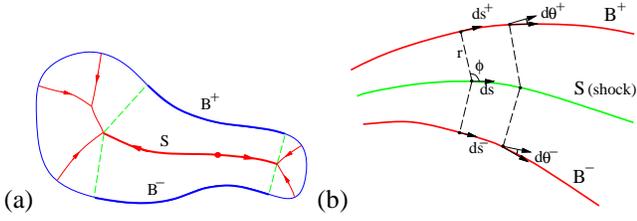
Figure 9: The shock $\mathcal{S}$ and the corresponding shape boundary segments $\mathcal{B}^+$ and $\mathcal{B}^-$.

the distance between two shapes within the same shape cell (identical shock topology), and then derive the cost of other edits as the limit of the deform cost as the shape moves to the boundary of the shape cell (transition). We derive the deform cost by summing over local shape differences cast in the language of differences between matching shock segment attributes. These are in turn defined based on a metric developed earlier for comparing curves [14].

This approach to comparing two curves consists of finding the minimum-cost deformation of one curve to the other where the cost is the sum of "stretching" and "bending" energies [21]. Specifically, we define the cost of matching infinitesimal segments on the two curves by length and curvature differences as $\mu = |d\hat{s} - ds| + R|d\hat{\theta} - d\theta|$, where R is a constant. The problem is then cast as minimizing a functional over all possible *alignments* between the two curves, using the notion of an alignment curve and dynamic programming [14].

To extend this idea to matching two shock-graph edges, we view the problem of deforming an edge in a shock graph, $\mathcal{S}$ in Figure 9a, to an edge in another shock graph by deforming the corresponding boundary segments, $\mathcal{B}^+$ and $\mathcal{B}^-$, representing a "joint curve matching problem". As in the case of curves, we penalize stretching and bending of the pair of boundary segments in terms of length and curvature differences. However, this notion of joint curve matching of shape boundary segment pairs is not enough, since changes in the relative pose of the two segments, namely, width of the shape and the relative orientation of the boundary segments, must also be taken into account.

Specifically, let the edges of the shock graphs being matched, $\mathcal{S}$ and $\hat{\mathcal{S}}$, be parameterized by s and $\hat{s}$ respectively. Let the boundary segments corresponding to $\mathcal{S}$, $\mathcal{B}^+$ and $\mathcal{B}^-$ be parameterized by s^+ and s^-. Let $\theta^+(s^+)$ and $\theta^-(s^-)$ be orientations of $\mathcal{B}^+$ and $\mathcal{B}^-$, Figure 9b. The boundary segments of $\hat{\mathcal{S}}$ are similarly defined. The cost of matching infinitesimal segments of the shock edges are defined by length differences $|d\hat{s}^+ - ds^+| + |d\hat{s}^- - ds^-|$, boundary curvature differences $|d\hat{\theta}^+ - d\theta^+| + |d\hat{\theta}^- - d\theta^-|$, width differences $2(|\hat{r}_0 - r_0| + |d\hat{r} - dr|)$, and relative orientation differences $2(|\hat{\phi}_0 - \phi_0| + |d\hat{\phi} - d\phi|)$,

$$\mu[\mathcal{S}, \hat{\mathcal{S}}; \alpha] = \int_0^{\tilde{L}} \left(\left| \frac{d\hat{s}^{\pm}}{d\xi} - \frac{ds^{\pm}}{d\xi} \right| + R \left| \frac{d\hat{\theta}^{\pm}}{d\xi} - \frac{d\theta^{\pm}}{d\xi} \right| + 2 \left| \frac{d\hat{r}}{d\xi} - \frac{dr}{d\xi} \right| \right) d\xi$$
$$+ 2|\hat{r}_0 - r_0| + 2 \int_0^{\tilde{L}} R \left| \frac{d\hat{\phi}}{d\xi} - \frac{d\phi}{d\xi} \right| d\xi + 2R|\hat{\phi}_0 - \phi_0|,$$
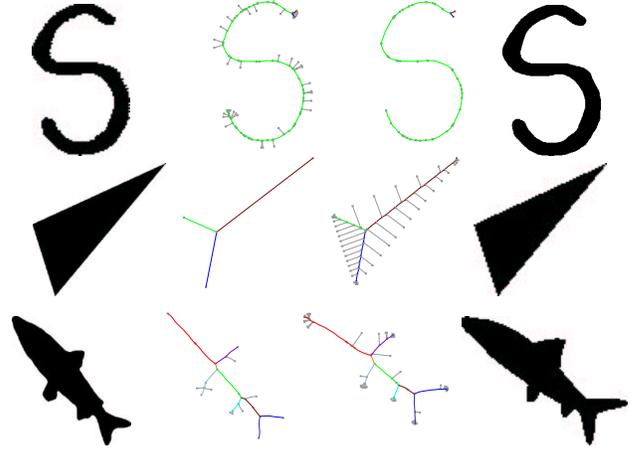
Figure 10: Shock matching in the presence of boundary noise where the same shape is represented by a coarser discretization (top and middle rows) or by a coarser discretization as well as a shape change (bottom row). Observe that shock graphs are matched correctly, pruning out all the spurious edges on the graph of the noisy boundary. Note that the match is intuitive: all the edges corresponding to the boundary noise are pruned, while matching salient parts, *e.g.*, heads, fins, and tips of tails of the two fishes.

where $\tilde{L}$ is the length of the alignment curve α, mediating the match [14]. Then, the deform cost of the shock edge $\mathcal{S}$ and $\hat{\mathcal{S}}$ is defined as the cost of the optimal alignment,

$$d(\mathcal{S}, \hat{\mathcal{S}}) = \min_{\alpha} \mu[\mathcal{S}, \hat{\mathcal{S}}; \alpha]$$

which is found by a dynamic-programming method [14].

The above describes the cost between the shock segments in an intrinsic manner. Other edit costs are considered to be limiting cases of a deform cost when one of the shock segments is shrunk to a point. We omit a discussion of these costs due to lack of space.

4 Recognition under Visual Transformations

This section first examines the performance of shock-graph matching in the presence of commonly occurring visual transformations, like articulation and deformation of parts, shadow and highlights, viewpoint variation, scale, boundary perturbations, occlusion. It then shows recognition results for two databases of roughly 100 shapes each.

Shock computation is sensitive to boundary perturbations, which can introduce spurious shock edges. While the traditional approach includes regularization in the detection process, one can view the regularization as part of the recognition process. Observe that the cost of a splice in this case is very low since corresponding changes required in

5
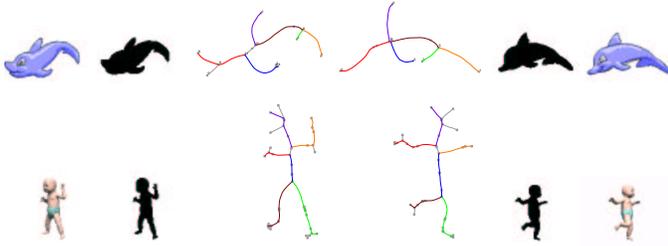
Figure 11: Effect of articulation and deformation of parts. Top row: Observe that the nose, fins, and tail of the dolphin in two different poses are matched intuitively. Bottom row: Matching of the two poses of the baby is intuitive. Observe, in particular, how the legs are matched correctly in spite of the self occlusion that merges part of the legs in the shape on the right.
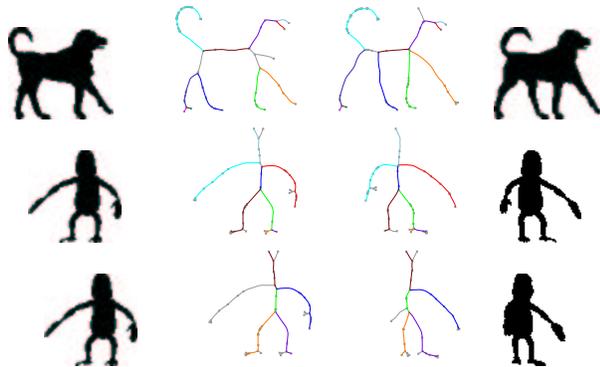
Figure 12: Effect of viewpoint variation. The 2D shapes were obtained by projecting a 3D model. Top row: Observe, in particular, how the hind legs of the retriever are matched correctly despite being partly merged in the shape on the left. Middle row: In both views of the cartoon model, the left and the right arms are visible, but despite a change in view the arms matched correctly. Bottom row: Note how the left arm of model in the right view has merged with the torso, thus changing the overall part structure. However, the correspondence given by shock-graph matching is intuitive as it is able to splice out the left hand in the left view.

the boundary model are slight. Thus, the cumulative cost of a large sequence of splices of spurious shocks is rather low and this typically forms the optimal sequence, Figure 10.

The shock graph of a shape inherently segments the shape into parts, and captures the hierarchical relationship between those parts. Thus, shock-graph matching implicitly involves matching the global hierarchy of parts in addition to matching the individual parts, making it robust to changes which may occur in some of the parts. Hence, shock matching is robust in the presence of articulation and deformation of parts of the shape, when the overall part hierarchy is not significantly altered, Figure 11.

Since 2D shapes are typically obtained from projections of 3D objects, robustness to viewpoint variations is critical to a 2D shape matching technique. When the viewpoint is varied gradually, the spatial location and the shape of
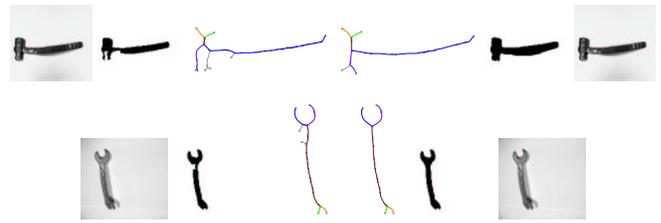
Figure 13: Shock matching when shadows and highlights affect the segmentation. The original images are from the tools database of Stan Sclaroff, Boston University. The tools were segmented using a region growing technique with manually selected thresholds. Note that parts of the hammer and the wrench are segmented as background due to highlights. Also, note the segmentation errors due to the shadow cast by the hammer (top right). The correspondence obtained by shock-graph matching is intuitive in both cases.
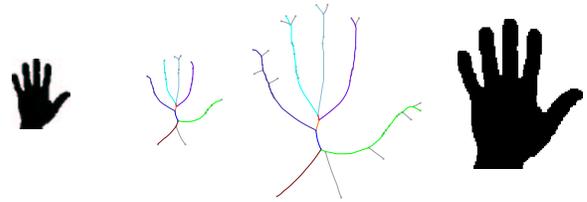
Figure 14: Shock matching in the presence of a scaling transform. The hand on the right was obtained by scaling the original hand by 225%. Observe that the correspondence is intuitive.

parts typically changes gradually. This is handled by the deform edit. Exceptionally, at certain views, there is a sudden appearance or disappearance of a part, which is handled by the splice transition. Also, changes in aspect are handled by contract operations. In general, a change in viewpoint constitutes a one-parameter family of deformations on shape, for which a well-defined neighborhood locally exists through the explicit embedding of transitions, Figure 12.

The presence of shadows and highlights can affect the segmentation of an object. Changes in the boundary caused by highlights tend to be small, but typically affect the shock-graph topology. On the other hand, changes in the shock graph caused by shadows are often more global in nature, and tend not to affect the shock-graph topology. Figure 13 shows that shock-graph matching is effective in the presence of a limited extent of segmentation errors.

We have examined the robustness of the proposed technique to changes in scale. Observe that the shock-graph topology is not changed when a global scaling transformation is applied. However, the shock edge comparison metric is not scale invariant (see Section 3). Nevertheless, shock-graph matching gives the intuitive correspondence in the presence of modest amounts of scaling, Figure 14. However, in the presence of large amounts of scaling (> 3 times), unintuitive correspondence *can* result if parts of unequal size are present, as the cost of matching equal-size but non-corresponding parts may become dominant. The
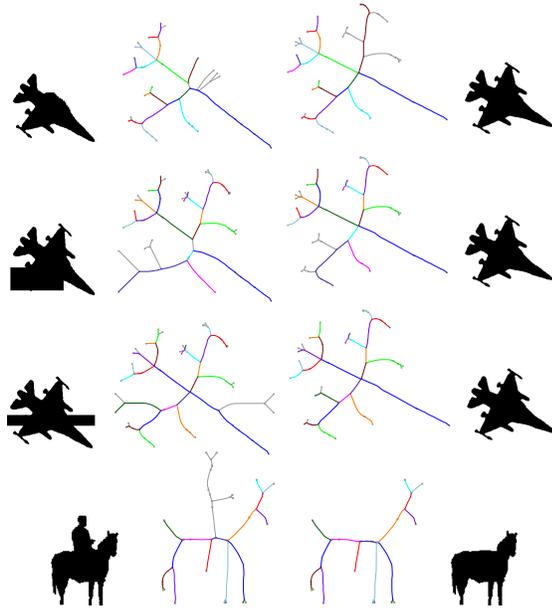
Figure 15: Shock graph matching in the presence of partial occlusion, both blending with the background and with the object. In all cases, the effects of occlusion on the match is confined to the occluded parts of the shape.

current implementation requires matching at several scales; the inclusion of scale in the matching process is one of our future goals.

Occlusion is a serious challenge for any recognition framework. We consider two types of occlusion, one where the background blends with part of the shape and second where part of the shape blends with the background. In both types shock-graph matching gives the intuitive correspondence by splicing out the occluded part, Figure 15. However, it may fail if a large part of the shape is occluded, in which case splice cost dominates the total cost.

Having demonstrated the performance of the recognition algorithm under various visual transformations, we now examine recognition rates for indexing into two databases. The first database was created from nine categories: fish, rabbit, airplane, "greeble" (obtained from Mike Tarr's collection), tool (obtained from Stan Sclaroff's database), hand, doll, four-legged animal, and sea-animal (obtained from Farzin Mokhtarian's database). We include eleven shapes in each category to allow for variations in form, as well as for occlusion, articulation, missing parts, *etc.*, for a total of 99 shapes. Each shape is matched against all others, and results were ordered by edit distance, Table 1. Since there are ten possible correct matches for each shape (within category) excluding the shape itself which is a perfect match, we measure the performance by a vector of 10 numbers which represents the proportion of correct matches. Five other shapes which show non-category matches are also shown. Recognition rates in % by this

measure are (100, 100, 100, 99, 99, 99, 97, 96, 95, 87); in other words the top three choices are always correct for this database.

We constructed a second database from samples of a very large database of shapes created for testing the compression rates for MPEG7, kindly provided by Latecki and Lakamper [10]. This database consists of 18 categories with 12 shapes in each category. Recognition rates in % are (100, 100, 100, 100, 100, 100, 99, 99, 97, 98, 95); in other words the top six choices are always correct for this database.

We aim to explore recognition rates on a much larger database. However, two developments are needed. First, the algorithm typically takes about 5 minutes on an SGI Indigo II (195 MHz) for the examples presented here. Improvements in efficiency are possible by rewriting the code and by revising the algorithm. Second, we have developed a notion of prototypes which can be included in the indexing scheme to reduce the computational burden. It is in the context of a much larger space of shapes that the effectiveness of this algorithm can truly be explored, these initial results are rather promising and support further exploration of this framework.

References

[1] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.

[2] S. Belongie and J. Malik. Matching with shape contexts. *CBAIVL*, 2000.

[3] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *PAMI*, 21(12):1312–1328, 1999.

[4] P. J. Giblin and B. B. Kimia. On the local form and transitions of symmetry sets, and medial axes, and shocks in 2D. ICCV, pages 385–391, 1999.

[5] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *PAMI*, 18(4):377–388, 1996.

[6] B. Kimia, J. Chan, D. Bertrand, S. Coe, Z. Roadhouse, and H. Tek. A shock-based approach for indexing of image databases using shape. *SPIE* vol. 3229:288–302, 1997.

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

[8] P. N. Klein, T. B. Sebastian, and B. B. Kimia. Shape matching using edit-distance: an implementation. *SODA*, pages 781–790, 2001.

[9] P. N. Klein, S. Tirthapura, D. Sharvit, and B. B. Kimia. A tree-edit distance algorithm for comparing simple, closed shapes. *SODA*, pages 696–704, 2000.

[10] L. J. Latecki, R. Lakamper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. *CVPR*, pages 424–429, 2000.

Table 1: Left: A database of 99 shapes with 9 categories and 11 shapes in each category. Right: Each shape is matched against every other shape in the database. The 15 nearest neighbors for a few shapes in the database ordered by the edit distance between each pair, normalized by the sum of the arclengths of the shapes and multiplied by 1000 for clarity of presentation. As there are 11 shapes in each category, up to ten nearest neighbors can be from the same category. The next five matches are shown for completeness. Observe that in most case the top 10 matches are from the same category. We rate the performance based on the number of times the ten nearest neighbors are in the same category. The results in proportions of 99 are: (99,99,99,98,98,98,96,95,94,86).

550	551	560	567	572	589	593	613	616	678	809	812	828	836	838
350	573	581	600	616	618	646	655	720	770	793	824	860	860	869
739	748	753	756	756	777	788	811	812	836	932	932	933	937	946
322	507	572	574	578	589	649	649	704	911	939	942	955	956	957
209	255	265	268	268	273	276	289	299	334	650	679	697	714	714
300	600	607	617	622	628	634	637	641	642	643	643	649	650	654
535	556	558	614	628	637	646	652	685	693	702	702	714	720	738

Table 2: Left: A database of shapes selected from the MPEG test database created by Latecki and Lakamper [10], who kindly made this available to us. We selected 96 shapes with 8 categories and 12 shapes in each category. Right: The 15 nearest neighbors for a few shapes in the database. As there are 11 other shapes in each category, up to eleven nearest neighbors can be from the same category. Observe that in most cases the top 11 matches are from the same category: the results in proportions of 96 are as follows: (96,96,96,96,96,96,95,95,93,94,91).

203	368	371	372	384	407	408	422	423	425	431	515	538	552	553
097	100	107	114	120	130	148	149	152	166	192	444	453	481	507
285	296	296	315	330	332	352	352	358	369	402	481	500	523	526
216	225	226	232	241	257	262	264	268	273	313	404	404	405	417
279	281	306	308	322	323	325	327	342	349	350	470	479	493	499
275	316	324	349	382	390	411	417	419	432	444	452	468	475	481

[11] T. Liu and D. Geiger. Approximate tree matching and shape similarity. ICCV, pages 456–462, 1999.

[12] R. Myers, R. Wilson, and E. Hancock. Bayesian graph edit distance. *PAMI*, 22(6):628–635, 2000.

[13] M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *PAMI*, 21(11):1105–1120, 1999.

[14] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Alignment-based recognition of shape outlines. *IWVF*, pages 606-618, 2001.

[15] D. Sharvit, J. Chan, H. Tek, and B. Kimia. Symmetry-based indexing of image databases. *JVCIR*, 9(4):366–380, 1998.

[16] K. Siddiqi, B. Kimia, A. Tannenbaum, and S. Zucker. Shocks, shapes, and wiggles. *IVC*, 17(5-6):365–373, 1999.

[17] K. Siddiqi and B. B. Kimia. A shock grammar for recognition. *CVPR*, pages 507–513, 1996.

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

[19] H. Tek and B. B. Kimia. Symmetry maps of free-form curve segments via wave propagation. *ICCV*, pages 362–369, 1999.

[20] R. Wagner and M. Fischer. The string-to-string correction problem. *J. ACM*, 21:168–173, 1974.

[21] L. Younes. Computable elastic distance between shapes. *SIAM J. Appl. Math.*, 1996.

[22] K. Zhang and D. Sasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Computing*, 18:1245–1262, 1989.

[23] S. C. Zhu and A. L. Yuille. FORMS: A flexible object recognition and modeling system. *IJCV*, 20(3), 1996.