

# Shock-Based Indexing into Large Shape Databases

Thomas B. Sebastian<sup>1</sup>, Philip N. Klein<sup>2</sup>, and Benjamin B. Kimia<sup>1</sup>

<sup>1</sup> Division of Engineering,  
Brown University,  
Providence, RI, USA

{tbs, kimia}@lems.brown.edu

<sup>2</sup> Department of Computer Science,  
Brown University,  
Providence, RI, USA  
klein@cs.brown.edu

**Abstract.** This paper examines issues arising in applying a previously developed edit-distance shock graph matching technique to indexing into *large* shape databases. This approach compares the shock graph topology and attributes to produce a similarity *metric*, and results in 100% recognition rate in querying a database of approximately 200 shapes. However, indexing into a significantly larger database is faced with both the lack of a suitable database, and more significantly with the expense related to computing the metric. We have thus (i) gathered shapes from a variety of sources to create a database of over 1000 shapes from forty categories as a stage towards developing an approach for indexing into a much larger database; (ii) developed a *coarse-scale* approximate similarity measure which relies on the shock graph topology and a very coarse sampling of link attributes. We show that this is a good first-order approximation of the similarity metric and is two orders of magnitude more efficient to compute. An interesting outcome of using this efficient but approximate similarity measure is that the approximation naturally demands a notion of *categories* to give high precision; (iii) developed an exemplar-based indexing scheme which discards a large number of non-matching shapes solely based on distance to exemplars, coarse scale representatives of each category. The use of a coarse-scale matching measure in conjunction with a coarse-scale sampling of the database leads to a significant reduction in the computational effort without discarding correct matches, thus paving the way for indexing into databases of tens of thousands of shapes.

**Keywords:** Similarity metric, object recognition, shape matching, shape retrieval, categorization, exemplars.

## 1 Introduction

The recognition of objects from their images is challenging and involves the integration of multiple cues like shape, color, and texture, which typically provide

complementary information. Shape is an important cue since humans can often recognize objects from line drawings as effectively as from images with color and texture information [8]. Indexing by shape content is therefore an important aspect of retrieving images from large databases. While retrieval based on shape can be described in terms of the underlying representation of shape, such as feature sets, point clouds, curves, medial axis, eigenshapes, among others, we present previous work in two classes based on the approach to organizing and querying the database, namely, whether the shape representation is eventually reduced to a finite set of attribute dimensions, or whether it is based on relative distances to other shapes.

In the first class, the “Euclidean Space” approach, each shape is reduced to a finite set of “features” and the distance between a pair of shapes is defined as the Euclidean distance between the features, *e.g.*, using moments, Fourier descriptors [17], invariants [22], *etc.* A good example of this class is the “active shape model” which represents shapes using a mean shape plus a set of linearly independent variation modes that are derived using a training set, *e.g.*, as in face recognition [1]. That each shape is summarized as a point in a low-dimensional Euclidean space allows for an organization and efficient querying of the database based on feature dimensions. Data structures such as KD-tree [5], Quadtree [23], X-Tree [6], and others rely on this Euclidean structure to iteratively divide the database elements into small clusters, some of which are not searched when querying the database [11,14].

In the second class, the “Metric Space” approach, each shape is only relatively represented by distances to other shapes. In this approach, differences in the basic representation of shape, *e.g.*, point clouds, curves, skeletons *etc.*, are summarized in the quality of the metric, and database organization, querying, and matching of shapes are based solely on distances between shapes. This distinction between representations of shape at the two levels is significant. For example, the measurement of shape context based on a point cloud representation [4] gives an alignment and finally a distance. Since at the matching stage only distances are used, this method falls into the metric space class. Also, approaches where an outline curve is elastically deformed into correspondence with another [3,12,33], and those that rely on matching medial axis representations [34,18,26,28,21,30] fall into this class.

Unfortunately, unlike the first class of representations where the structure of the underlying Euclidean space is used to organize and query the database, it is not clear how the metric space representations can be used to organize and query a database. The projection of the metric space to a low-dimensional Euclidean space can be problematic if shape variations occur along an infinite number of dimensions, each with a wide range of extents. These variations are not sparse and thus the mapping to a low-dimensional space will necessarily exclude certain dimensions, variations along which will not lead to robust results. Thus, the problem is one of organizing a metric space which can at best be approximated as a very high-dimensional Euclidean space.

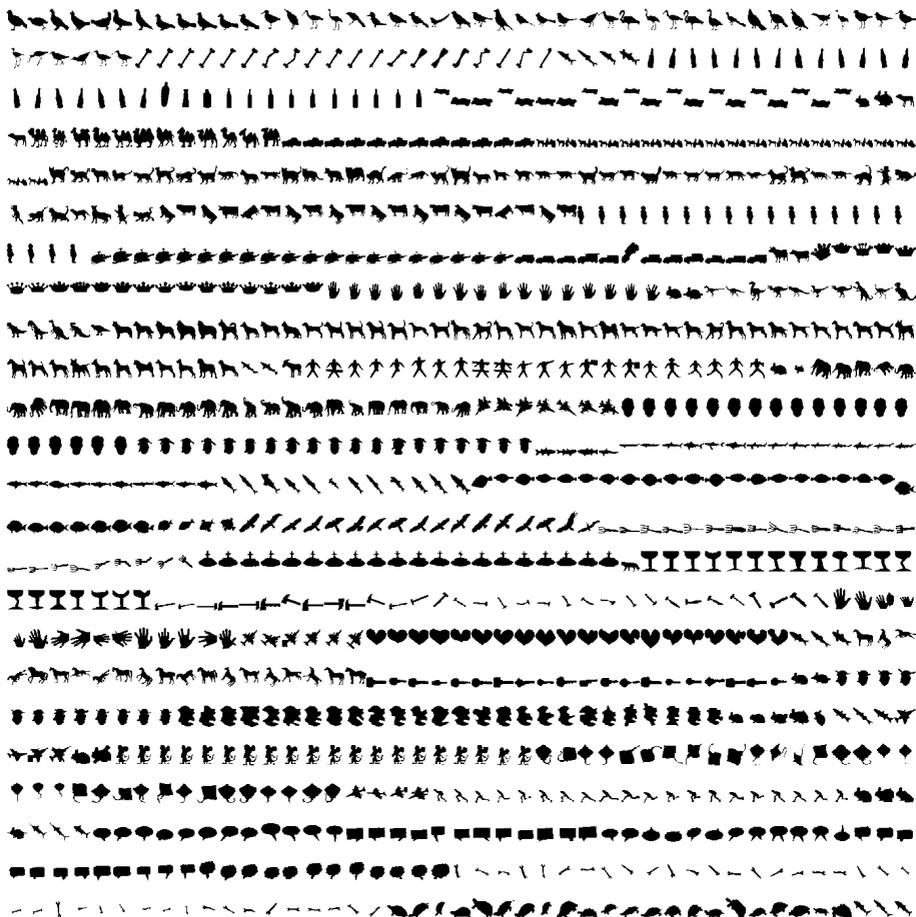
Distance-based nearest neighbor search techniques [10,31,32,9] have been used for searching metric spaces. These methods typically use the triangle inequality to avoid computing the distances of the query to all elements in the database. The basic idea is to select certain elements as *pivots*, group the remaining elements into clusters based on their distances to the pivots, and use the triangle inequality to prune out certain clusters during the search of the nearest neighbor of a query. However, the performance of these methods deteriorate as the dimensionality of the space increases (*curse of dimensionality*), principally because pairwise distances between elements in a high-dimensional space tend to fall in a narrow range, and the triangle inequality can only eliminate a few elements [9].

In this paper, we examine approaches to make indexing into large databases by matching shock graphs practical. In this approach the edit distance between shock graphs [25,16,15] is used as a *metric* of similarity between shapes. It is obtained by exhaustively searching for the optimal deformation path between two shapes, and using the cost of this path as the distance between two shapes. This approach is robust to various transformations like boundary perturbations, articulation, deformations of parts, occlusion, *etc.*, and gives intuitive results for indexing into shape databases. Results of two databases of 99 and 216 shapes show 100% correct recognition rate for the top three rank-ordered matches. However, this robustness to a range of variations is at the cost of a *computationally expensive metric*, and hence it is not practical for the direct querying of *large* databases, without employing additional steps.

A first step to reducing the computational burden is to devise an computationally inexpensive but approximate measure for comparing two shapes [20]. This “coarse” measure can then be used to quickly rule out highly distant shapes, retaining a few candidates for the “fine” comparison stage. Clearly, a reduction in accuracy of the distance measure implies that if only a very small margin of error is allowed much of the database has to be considered for the fine-scale comparison. This defeats the purpose of developing a coarse measure. However, observe that extremely high precision can be achieved if a reduction in accuracy in the matching is accompanied by a grouping of each shape with a set of neighboring shapes leading to the concept of *categories*.

A second step to making indexing practical involves the use of exemplars to represent a category. The key observation is that shapes in a category appear tightly clustered when viewed in the context of all shapes, and a category can thus be represented by a few exemplars. The membership of a query in a category is then determined by the set of distances to exemplars, but not all members of the category. The experiments in this paper indicate that the use of exemplars leads to significant savings in computational time without significantly affecting the recognition rate. Note that the use of exemplars is analogous to the use of pivots in distance-based indexing schemes.

A third step is required to evaluate indexing into large databases, namely the formation of a large shape database. A key contribution of this paper is the development of such a database with 1032 shapes organized into forty categories,



**Fig. 1.** The database of shapes used in the experiments. There are a total of 1032 shapes from 40 categories. The shapes were collected from a variety of sources. We have included shapes from the MPEG-7 test database, Farzin Mokhtarian (fish, sea animals), Mike Tarr (greebles), and Stan Sclaroff (tools).

Figure 1. The rest of the paper is organized as follows. Section 2 reviews the edit-distance shock graph approach to computing a shape comparison techniques by focusing on details which are needed to devise a coarse-scale measure. Section 3 discusses how this measure is defined, and describes the results of experiments with it which lead to the need for a notion of categories. Section 4 discusses the use of exemplars for further increasing the efficiency of indexing.

## 2 Recognition of Shapes by Matching Shock Graphs

In this section we review a framework for matching shapes by editing their shock graphs [25,16,15]. The basic idea is to view a shape as a point in a shape space where the distance is defined as the minimum-cost deformation path connecting one shape to the other. It is immediately clear, however, that there are an infinite number of ways a shape can deform. The practical search for an optimal path requires a discretization of the space of shapes and their deformations. Observe that generally as a shape deforms, the shock graph topology does not change, but only the attributes of the shock graph edges change. The identification of shapes by shock graph topology leads to a notion of a *shape cell*, the equivalence class of shapes with the same shock graph topology. This results in a discretization of the shape space. However, near the “boundaries” of each shape cell, or “transition” shapes, an infinitesimal change in the shape causes an abrupt change in the shock graph topology. These shapes are the instabilities of the medial axis/shock graph, which any recognition strategy must deal with. We have previously enumerated and classified these transitions [13]. As the shock graph topology changes at these transitions, matching strategies must explicitly identify shapes on either side of the transition as being close. Towards this end, we use the shock transitions to annotate deformation paths. Thus, as a second discretization, we define a *shape deformation bundle* which is the set of one-parameter families of deformation paths passing through the same set of shock transitions.

While the discretization of the shape space and their deformation paths drastically reduce the underlying dimensions of search, they do not avoid considering paths that venture into completely unrelated areas of the shape space: a shape can first be completely changed and these changes can be undone in a different manner to reach the second shape. Thus, to avoid unnecessarily complicating the shape, as a third measure, we consider each deformation path as a pair of simplifying paths, one for each shape, reaching a common third shape which is “simpler” than both.

With these three measures, the search space is significantly reduced, but it is still large. To efficiently search the discretized space we have developed an edit distance approach [16,15]. In the graph domain, each shock transition is represented by an “edit” operation, and four types of edit operations are defined based on shock transitions [13]: (i) the *splice* operation deletes a shock branch and merges the remaining two; (ii) the *contract* operation deletes a shock branch connecting two degree-three nodes; (iii) the *merge* operations combine two branches at a degree-two node; (iv) we also define a *deform* edit to relate two shapes with the same shock graph topology but with different attributes. We find the minimum-cost sequence of edits by an efficient polynomial-time tree edit distance algorithm developed in [16,15].

The costs of edit operations are assigned so that they are consistent with each other and also with our perceptual similarity metrics. Our approach is to first define the *deform cost*, and then derive all other edit costs as limits of the deform cost when the shape moves to the boundary of the shape cell (transition shape) [25]. The deform cost is defined by summing over local shape differences, and is based on an extension of an intrinsic curve comparison metric [24]. Ob-

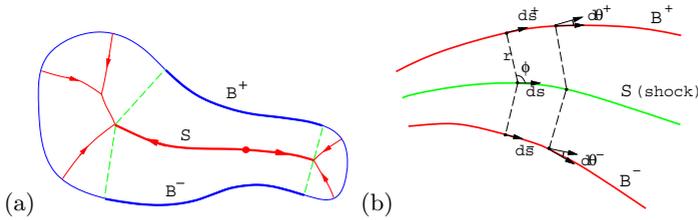


Fig. 2. The shock  $S$  and the corresponding shape boundary segments  $B^+$  and  $B^-$ .

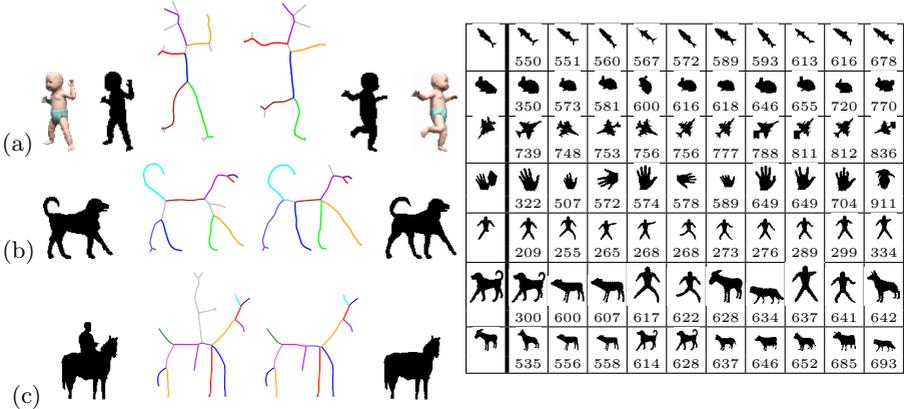


Fig. 3. Left: This figure from [25] illustrates that the shock graph matching algorithm works well in presence of articulation and deformation of parts (a), view-point variation (b), and occlusion (c). Same colors indicate matching shock branches, and grey colored branches have been pruned. The matching is intuitive in all cases. Observe, in particular, how the rider on the horse is pruned when matched against horse by itself. Right: Indexing results for a database of 99 shapes (9 categories and 11 shapes in each category) [25]. The 10 nearest neighbors for a few shapes in the database ordered by edit distance are shown. We rate the performance based on the number of times the 10 nearest neighbors are from the same category as the query. The results in proportions of 99 are: (99,99,99,98,98,98,96,95,94,86).

serve that each edge in the shock graph corresponds to two boundary segments, Figure 2a, and the problem of deforming a shock edge  $S$  to another shock edge can be viewed as one of deforming the corresponding boundary segments,  $B^+$  and  $B^-$ , representing a “joint curve matching problem”. Let the edges of the shock graphs being compared,  $S$  and  $\hat{S}$ , be parameterized by  $s$  and  $\hat{s}$  respectively. Let the boundary segments corresponding to  $S$ ,  $B^+$  and  $B^-$ , be parameterized by  $s^+$  and  $s^-$ , and let  $\theta^+(s^+)$  and  $\theta^-(s^-)$  be their orientations. Let  $r(s)$  and  $\phi(s)$  denote the width and relative orientation of  $B^+$  and  $B^-$ . (See Figure 2b.) The properties of  $\hat{S}$  are similarly defined. The local shape differences are measured in terms of differences in boundary lengths, boundary curvatures, width and relative orientation, and expressed as the following functional

$$\begin{aligned} \mu[\mathcal{S}, \hat{\mathcal{S}}; \alpha] = & \int \left( \left| \frac{ds^\pm}{d\xi} - \frac{d\hat{s}^\pm}{d\xi} \right| + R \left| \frac{d\theta^\pm}{d\xi} - \frac{d\hat{\theta}^\pm}{d\xi} \right| + 2 \left| \frac{dr}{d\xi} - \frac{d\hat{r}}{d\xi} \right| \right) d\xi \\ & + 2|\hat{r}_0 - r_0| + 2 \int R \left| \frac{d\phi}{d\xi} - \frac{d\hat{\phi}}{d\xi} \right| d\xi + 2R|\hat{\phi}_0 - \phi_0|. \end{aligned} \quad (1)$$

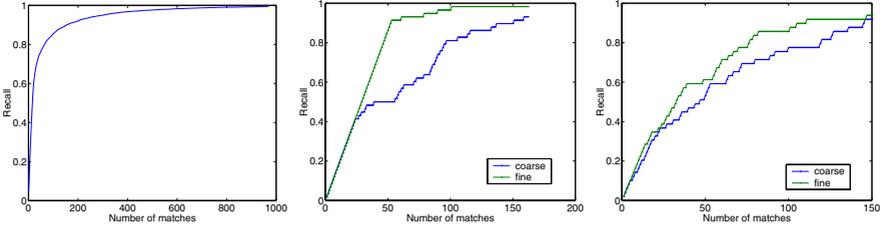
The alignment is expressed in terms of an alignment curve  $\alpha$  to ensure symmetric treatment of the two shock edges [24]. Finally, the deform cost of the shock edges  $\mathcal{S}$  and  $\hat{\mathcal{S}}$  is defined in terms of the alignment that minimizes the above functional, which is also found by a dynamic-programming method [24].

The shock graph matching technique works well in the presence of a variety of visual transformations like articulation, deformation of parts, boundary noise, viewpoint variation and occlusion, and gives excellent recognition rates for two distinct shape databases of roughly 100 and 200 shapes each, Figure 3. However, the running time of each match, which is 3-5 minutes on a typical workstation prohibits its use in organizing and querying databases with large number of shapes. We now detail two ideas to render this task practical.

### 3 Coarse-Scale Matching of Shock Graphs

Observe that in a large shape database, most of the shapes are typically very dissimilar to the query shape, and can possibly be eliminated as potential matches without using detailed and expensive comparisons. A close comparison of two shapes is warranted only if they are fairly similar. For example, if the query is a “camel”, it is desirable to eliminate all “fish” shapes *without* using the computationally expensive comparison of all shock graph attributes. A detailed match at a later stage will make fine distinctions among remaining shapes. This coarse-to-fine matching strategy can significantly reduce the computational time for indexing into databases. In this section, we develop a *coarse-scale version* of the similarity metric, and show that it is effective in identifying the top matches of the query efficiently.

Note that in matching two shock graphs as described in Section 2, most of the computational effort goes towards computing the cost of the edit operations, in particular, computing the deform costs for each pair of paths in the two shock graphs. This requires the computation of  $O(n^2m^2)$  optimal alignments, where  $n$  and  $m$  are the number of nodes in the shock graphs. Furthermore, finding each optimal alignment uses dynamic programming, which is quadratic in the number of samples along the shock graph paths. Thus, reducing the cost of each alignment, *e.g.*, by reducing the number of samples along the shock paths, would significantly reduce the overall computational expense. One idea, therefore, is to completely avoid computing the optimal alignment, and instead compute the deform cost in terms of the end points of the shock graph paths. While this is clearly not always accurate, it does provide a first-order approximation to the similarity metric. As before, we compare differences in boundary lengths, total curvatures, width and relative orientation for each shock path. Specifically, we define the deform cost between shock edges  $\mathcal{S}$  and  $\hat{\mathcal{S}}$  as



**Fig. 4.** Left: Average *recall* as function of  $N$ , the size of the top ranking coarse matches, for 200 randomly selected shapes from the database in Figure 1. Middle and right: Two instances of recall as a function of  $N$  for a fish and a dog (shown in Table 2), respectively.

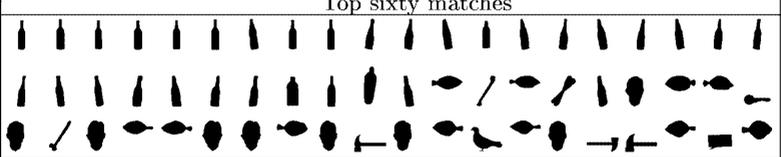
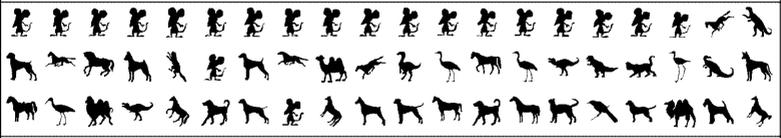
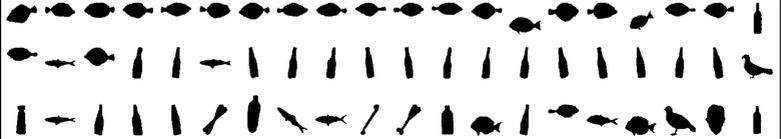
$$\mu[\mathcal{S}, \hat{\mathcal{S}}] = \left| \int d\hat{s}^\pm - \int ds^\pm \right| + R \left| \int d\hat{\theta}^\pm - \int d\theta^\pm \right| + 2 \left| \int d\hat{r} - \int dr \right| + 2|\hat{r}_0 - r_0| + 2R \left| \int d\hat{\phi} - \int d\phi \right| d\xi + 2R|\hat{\phi}_0 - \phi_0|. \quad (2)$$

Note that this measure does not rely on finding an optimal alignment, and is thus very efficient to compute; it reduces the computational requirement of matching a pair of shock graphs from a few minutes to a few seconds. This is a *significant* speedup: the computation of pairwise distances and the distance from a query for a database of 1000 shapes is 500,000 and 1000 times the cost of computing the metric, respectively. For a database of 10,000 shapes these numbers are 50,000,000 and 10,000, respectively! The premise underlying the use of a coarse-scale metric is that it allows for the pruning of very dissimilar shapes, *e.g.*, if we could safely discard 90% of the database using a coarse-scale match, the overall cost is reduced to 10% of the fine-scale match, plus the coarse-scale matching cost. Since the cost of the latter is negligible compared to the cost of fine-scale matching of 10% of the database, this would result in approximately an order of magnitude speedup.

Therefore, the effectiveness of the coarse-scale matching is measured in terms of its ability to capture the correct match in a small fraction of the top rank-ordered matches. This is precisely the notion of *recall*<sup>1</sup>. Figure 4 shows the average recall for 200 queries. Observe that well over 90% of similar shapes are captured in the top 200 matches. Table 1 shows the top 60 matches for a few query shapes using coarse-scale shock-graph matching. Observe that the top matches (i) contain a large number of shapes in the correct category, and (ii) the correct category is predominant in the first few matches, *but* (iii) not all shapes in the correct category are retrieved, unless a very large portion of the database is considered. Thus, while for several queries *all* shapes in the same category are returned, for some others one or more instances are missing. This situation will get only worse as the number of shapes in the database increases.

<sup>1</sup> Precision and recall have been used to evaluate the performance of a shape retrieval scheme [7,19]. Precision is the ratio of the number of similar shapes retrieved to the total number of shapes retrieved, while recall is the ratio of the number of similar shapes retrieved to the total number of similar shapes in the database.

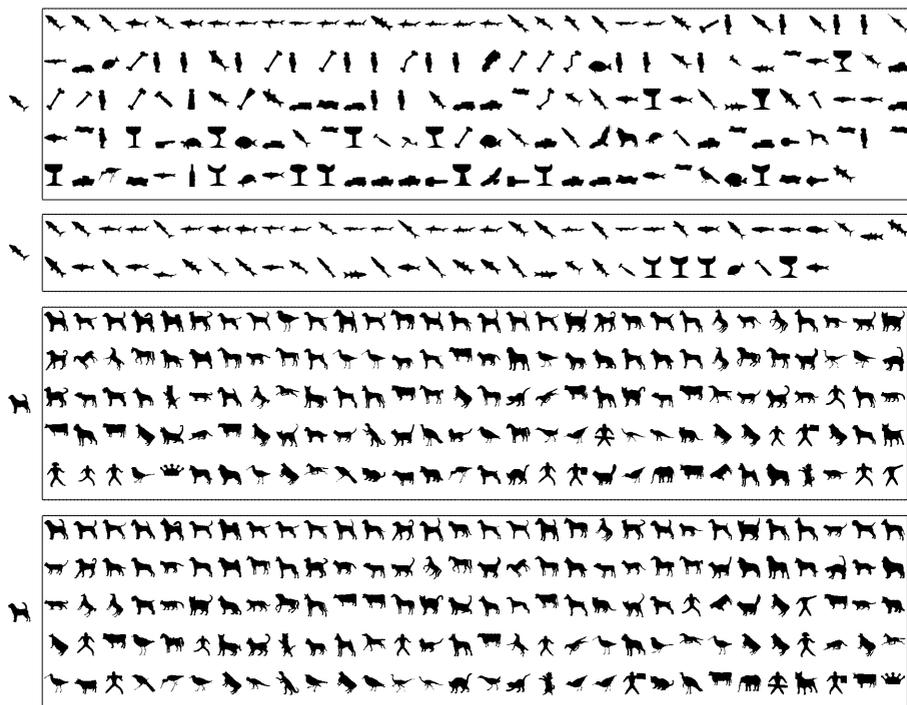
**Table 1.** This table shows that coarse-scale match is a good first-order approximation of the detailed shock graph matching. The query shape (left column), top 60 matches (middle column) and number of shapes from the correct category (right column) are shown. The matches are ordered from left to right and top to bottom. Observe that in most cases the top matches contain many shapes that are similar to the query.

Query	Top sixty matches	Correct
		32/33
		20/20
		24/32
		20/20
		29/49

Thus, if the correct ranking is required without much tolerance for error, an increasing number of top ranking coarse-scale matches must be presented for fine-scale comparison. This clearly defeats the purpose of employing a two-stage coarse-to-fine strategy. See Table 2 and Figure 4 for the interaction between coarse and fine stages.

High precision can be achieved by observing that as the ability of the metric in precisely locating one shape with respect to others is diminished, there is an implicit implication of equivalence between a shape and its immediate *neighbors*. The explicit formulation of such an equivalence to improve precision implies connecting a shape to its neighboring shapes the basis of forming *categories*. To illustrate that the neighborhood topology tremendously improves precision, consider the extreme case of the dog example in Table 1 where only 29/49 are retrieved in the top 60 coarse matches, and where the complete retrieval of all shapes requires considering the top 260 matches. We observe that the missing 20/49 at the coarse-scale can be retrieved by *association*, *i.e.*, membership in a

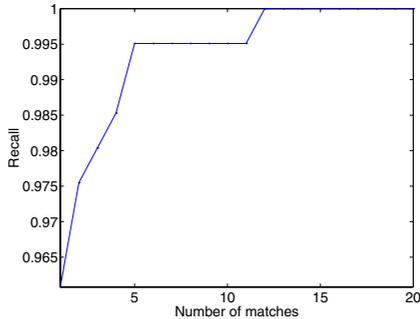
**Table 2.** This table illustrates the coarse-to-fine strategy for two examples. In the first example, using the coarse-scale match 158 shapes (first row) out of 1032 are selected for fine-scale matching. Note that 54/58 fish shapes are retrieved. These 158 shapes are compared to the query using fine-scale shock graph matching, and the top 61 matches contain the 54 fish shapes (second row). Also, note that the top 53 matches are fish shapes. In the second example, top 150 matches are needed to retrieve 47/49 dog shapes using coarse-scale matching (third row). In this case also, fine-scale matching (last row) does better than coarse-scale matching, but represents less of an improvement compared to most other cases. Most of the errors in fine-scale matching are due to the confusion between dog and other quadrupeds. In fact, the top 82 matches for fine-scale shock matching are quadrupeds. The recall in both cases is compared in Figure 4.



category. Thus, if the top ranking coarse-scale match is a dog, all shapes closely connected to it (all dogs, or some subsets of dogs) are automatically included as retrieved results. More precisely, define a *category-based ranking* of retrieval as the relative ranking of each category by the order of appearance of its first exemplar relative to those of other categories. For example, in the case of the fish in top row of Table 2, the categories are “fish”, “hammer”, “child” etc. Table 3 shows that in the category-based ranking, the top category is correct 96% of the time, and the top five categories are correct 100% of the time, see bottom of Table 3.

**Table 3.** (Top) This table shows the ranking by category, where each category is ranked by its first sample in the coarse-scale match. The top category for 200 randomly selected queries is correct 96% of the time. (Bottom) A summary of correctness for top  $N$  categories.

Query	1	2	3	4	5	6	7	8	9	10
										
glass	glass	key	bottle	brick	dude	hammer	flatfish	fish	ray	face
										
bird	bird	fountain	stef	textbox	flightbird	greeble	chopper	cat	fork	dog
										
camel	horse	camel	dog	cat	hand	bird	crown	elephant	cattle	dinosaur
										
car	car	key	child	fish	bird	dog	brick	heart	textbox	stef
										
carriage	carriage	dog	flightbird	dinosaur	rat	dude	horse	bird	fork	cattle
										
cattle	dog	bird	elephant	cat	cattle	crown	dude	horse	hand	dinosaur
										
chopper	chopper	bird	stef	dog	rabbit	flightbird	hand	horse	textbox	dude
										
crown	crown	dinosaur	bird	dog	elephant	camel	cat	rat	stef	turtle
										
dude	dude	dog	glass	fork	cat	dinosaur	bird	fish	rabbit	hand
Query	96%	98%	99%	99.5%	100%	100%	100%	100%	100%	100%



**Fig. 5.** This figure shows the category-based recall for 200 queries based on categories returned in top  $N$  matches.

A second observation is required before claiming that examining the categories of the top-ranking coarse-scale matches at the fine-scale is both sufficient and efficient. For this to be true, the top few categories must be packed into the top few matches by the coarse-scale measure. Figure 5 which plots category-based recall shows the first match gives 96% correct category, while the top six matches gives 99.5% and the top twelve matches give 100% correct recall. Thus, the top dozen matches result in top five categories which are required for a fine-scale comparison with zero-error recognition. The selection of  $N$ , the top matches to be considered, is a parameter of the system.

## 4 Exemplars for Indexing Into Databases

The approach developed in Section 3 reduces the number of fine-scale computations significantly. However, the coarse-scale comparison of a query is done against *all* shapes in the database. This also becomes unrealistic for significantly larger databases. For example, if we have a database of 10,000 shapes, and if the cost of a coarse-scale match is 2 seconds, the coarse-scale matching of each query would take about 20,000 seconds, *i.e.*, more than 5 hours! To alleviate this difficulty, observe that shapes are not randomly organized, but rather enjoy a neighborhood topology. Intuitively, when considering the collection of “fish” in the context of other shapes, they seem to be tightly clustered and densely packed around each other, and at some distances to other shapes. Thus, if we establish that our query *e.g.*, a “camel” is highly distant from one instance of a fish, “fish 1”, it is also likely to be distant from nearby neighbors, say “fish 2”. Formally, using the triangle inequality,  $d(\text{camel}, \text{fish } 2) > |d(\text{camel}, \text{fish } 1) - d(\text{fish } 1, \text{fish } 2)|$ , which establishes a lower bound for  $d(\text{camel}, \text{fish } 2)$  without having to compute it. This suggests that for a cluster of shapes distant from the query the actual distance needs to be computed only for a few samples. In other words, each category can be represented by a few representative samples, or *exemplars*.

This representation of a tightly packed, dense cluster of shapes by a few samples is consistent with the psychophysics data [29]. There has also been ample use of exemplars or prototypes in computer vision, *e.g.* [2]. From a computational perspective, with the use of exemplars the number of matches necessary is proportional to the number of categories, and not to the total number of shapes. The process of selecting optimal exemplars is a challenging task in itself, and is not discussed in this paper. For the experiments reported in this paper, we initially selected several representative exemplars to represent each category. We now describe our experiments in using exemplars to further reduce computational requirements in indexing a large database.

In our experiments the triangle inequality turns out to be a strong condition. An alternative approach is to consider the similarity of the query to each exemplar, and use it to define a fuzzy notion of membership in each category. Let  $C_i, i = 1, \dots, N$  be the categories in the database. Let  $E_i^k$  represent the  $k^{\text{th}}$  exemplar of category  $C_i$ , and denote  $\mathbf{E}_i = \{E_i^k | k = 1, \dots, N_i\}$  as the set of exemplars for category  $C_i$ . Let  $Q$  be the query shape, and let  $d(Q, E_i^k)$  be the distance between  $Q$  and  $E_i^k$ . The similarity of the query to an exemplar is then defined as

$$S(Q, E_i^k) = \exp\left(-\frac{d(Q, E_i^k)}{\min_{\substack{i=1, \dots, N \\ k=1, \dots, N_i}} d(Q, E_i^k)}\right). \quad (3)$$

Shepard [27] has shown that the use of exponential in relating perceptual distance and similarity is universal. We do not directly use the closest exemplar’s category because categories tend to have different variations; some tightly clustered,

**Table 4.** This table illustrates the top 5 matching categories ranked by the membership function using the coarse-scale distance measure for some representative query shapes. The closest matching exemplar is used to represent each category.

					
bird	dinosaur	bird	horse	dog	elephant
					
bone	bone	bottle	turtle	classic	child
					
car	car	classic	child	flightbird	fish
					
bottle	bottle	ray	face	flatfish	bone
					
camel	camel	dinosaur	bird	horse	dog
					
carriage	carriage	rat	bird	camel	dude
					
cat	horse	dog	cat	cattle	bird
					
cat	cat	cattle	horse	dog	bird
					
child	child	classic	flatfish	bone	fish
					
chopper	chopper	bird	key	textbox	dog
					
classic	classic	child	bottle	bone	flatfish
					
dog	horse	dog	dinosaur	camel	cat
					
elephant	elephant	dinosaur	bird	horse	dog
					
plane	plane	bird	elephant	dinosaur	dog

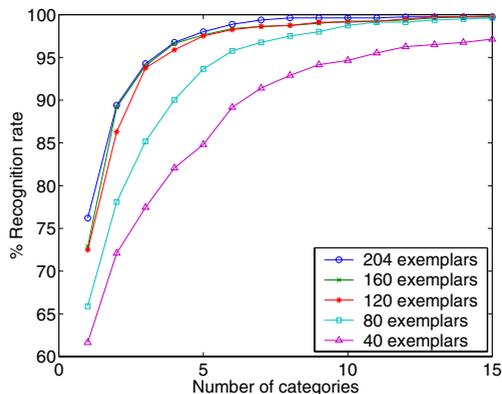
					
greeble	turtle	greeble	flatfish	dog	textbox
					
fish	bone	child	fish	hammer	flatfish
					
dinosaur	dinosaur	bird	cattle	dog	horse
					
glass	glass	key	bottle	ray	brick
					
hammer	hammer	key	glass	rabbit	turtle
					
plane	plane	bird	cattle	dude	dinosaur
					
horse	horse	dog	cattle	camel	cat
					
rabbit	rabbit	textbox	greeble	turtle	Heart
					
key	key	child	brick	bottle	dog
					
fish	fish	hammer	bone	flatfish	brick
					
ray	ray	flatfish	textbox	face	Heart
					
textbox	ray	textbox	hand	bird	bottle
					
turtle	turtle	cattle	dog	bone	flatfish
					
tool	tool	hammer	brick	bone	glass
					
fish	fish	child	bone	hammer	flatfish

others highly variable. Consider a query that is slightly closer to an exemplar in a tightly clustered category than to an exemplar in a largely variable category. It is intuitively clear that it is more likely that this query is a member of the second category. The statistical analysis of an appropriate measure of membership is beyond the scope of this paper. As a first approximation, we define a membership measure  $\nu$  for the query in each category  $C_i$  as the sum of similarities of the query to all exemplars of  $C_i$ .

$$\nu(Q, C_i) = \sum_{k=1, \dots, N_i} S(Q, E_i^k). \quad (4)$$

This membership measure is used to identify the closest categories to each query.

The overall strategy is as follows. For each query  $Q$ , we compute  $d(Q, E_i^k)$  for each exemplar,  $k = 1, \dots, N_i, i = 1, \dots, N$ . This is of significantly lower cost than computing  $d(Q, S)$  for all shapes  $S$  in the database. In our experiments we initially selected three to six exemplars per category, for a total of 204 exemplars. Thus, the cost of computing  $d(Q, E_i^k)$  is five times lower than  $d(Q, S)$ . We then used  $\nu(Q, C_i)$  to rank order likely categories and for these performed fine-scale matching. Table 4 shows the top matching categories for a few representative



**Fig. 6.** This figure illustrates the percent correct match found in the top  $N$  ranked categories for a range of exemplars per category. Observe that for about six exemplars the first match is right 78% of the time, while for one exemplar it is right only 62% of the time. Also, note that the lower the number of exemplars used, the more categories must be considered for each level of error tolerance. For example, allowing for only a 3% error tolerance, we must consider (15, 8, 5, 4, 4) categories for (1,2,3,4,6) exemplars per category, respectively. This suggests a hierarchical scheme: use the one exemplar and 15 categories to rule out 25/40 categories; then use a few, say four exemplars to prune further. We expect that these curves will become significantly distinct as the number of total shapes per database increases, thereby increasing the utility of a hierarchical scheme in larger databases.

query shapes using exemplar-based matching. The top one, five and ten categories are correct 78.5%, 98.5% and 99.5% of the time, respectively, Figure 6.

We have experimented with the use of fewer exemplars to represent each category. Figure 6 shows the result of discarding exemplars whose intra-category distance to all other exemplars in that category is the smallest. This process is repeated until a pre-defined number of exemplars remain. The results show that while fewer exemplars generally implies considering additional shapes for the fine-scale match, as expected, there is a trade-off between the two. Note that the use of approximately three exemplars per category for a total of 120 to represent the database of 1032 shapes results in an order of magnitude speedup to determine an object's category, Figure 6. The conclusion of these experiments is to follow a hierarchical representation at the exemplar level as well: with two or three *principal exemplars* we can safely rule out 75% of categories, while with three *auxiliary exemplars*, we can rule out 50% of the remaining categories, thus reducing the expensive fine-scale matching only to very few categories. A goal of our future work is to examine this type of hierarchical exemplar representation.

## 5 Conclusion

We have gathered a database of over 1000 shapes to examine the issues of computational cost and the need for database organization for indexing into large databases. We have presented two approaches which together significantly reduce computational requirements. First, we developed a coarse-scale distance measure which leads to 50-100 times speedup in distance computations. We showed that the need for high precision at the coarse-scale matching level naturally gives rise to the need to organize the database into categories. Second, we showed that a coarse-scale sampling of each category using exemplars does not significantly reduce categorization and indexing performance if a membership function is defined based on the similarity measure. Together, the coarse-scale measure and the coarse-scale sampling lead to significant reduction in computational requirements. We believe that together with a hierarchical organization of both categories and exemplars we can apply this approach to indexing into databases of tens of thousands of shapes.

**Acknowledgments.** Philip Klein acknowledges the support of NSF grant CCR-9700146 and Benjamin Kimia acknowledges the support of NSF grant BCS-9980091.

## References

1. A.Lanitis, C.J.Taylor, and T.F.Cootes. Automatic interpretation and coding of face images using flexible models. *PAMI*, 19(7):743–756, 1997.
2. R. Basri. Recognition by prototypes. *IJCV*, 19(2):147–167, August 1996.
3. R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.
4. S. Belongie, J. Malik, and J. Puzicha. Matching shapes. *ICCV*, pages 454–461, 2001.
5. J. L. Bentley and J. H. Friedman. Data structures for range searching. *ACM Computing Surveys*, 11(4):397–409, 1979.
6. S. Berchtold, D. A. Keim, and H.-P. Kriegel. The X-tree: An index structure for high-dimensional data. *VLDB*, pages 28–39, 1996.
7. L. Bergman and V. Castelli, editors. *Image Databases, Search and Retrieval of Digital Imagery*. John Wiley and Sons, 2002.
8. I. Biederman and G. Ju. Surface versus edge-based determinants of visual recognition. *Cognitive Psychology*, 20:38–64, 1988.
9. S. Brin. Near neighbor search in large metric spaces. *VLDB*, pages 574–584, 1995.
10. E. Chavez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273 – 321, 2001.
11. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petrkovic, and W. Equitz. Efficient and effective querying by image content. *J. Intelligent Information Systems*, 3:231–262, 1994.
12. 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.

13. 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.
14. W. I. Groski and R. Mehrota. Index-based object recognition in pictorial data management. *CVGIP*, 52:416–436, 1990.
15. P. Klein, T. Sebastian, and B. Kimia. Shape matching using edit-distance: an implementation. *SODA*, pages 781–790, 2001.
16. P. Klein, S. Tirthapura, D. Sharvit, and B. Kimia. A tree-edit distance algorithm for comparing simple, closed shapes. *SODA*, pages 696–704, 2000.
17. C. Lin and R. Chellappa. Classification of partial 2-D shapes using Fourier descriptors. *PAMI*, 9(5):686–690, 1987.
18. T. Liu and D. Geiger. Approximate tree matching and shape similarity. *ICCV*, pages 456–462, 1999.
19. E. Milios and E. Petrakis. Shape retrieval based on dynamic programming. *IEEE Trans. Image Processing*, 9(1):141–146, 2000.
20. G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. *CVPR*, pages I:723–730, 2001.
21. M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *PAMI*, 21(11):1105–1120, 1999.
22. E. Rivlin and I. Weiss. Local invariants for recognition. *PAMI*, 17(3):226–238, 1995.
23. H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2):187–260, 1984.
24. T. B. Sebastian, P. N. Klein, and B. B. Kimia. Alignment-based recognition of shape outlines. *IWVF*, pages 606–618, 2001. Springer.
25. T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing shock graphs. *ICCV*, pages 755–762, 2001.
26. D. Sharvit, J. Chan, H. Tek, and B. B. Kimia. Symmetry-based indexing of image databases. *JVCIR*, 9(4):366–380, 1998.
27. R. N. Shepard. Toward a universal law of generalization for psychological science. *Science*, pages 1317–1323, 1987.
28. K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *IJCV*, 35(1):13–32, November 1999.
29. M. J. Tarr and H. H. Bulthoff, editors. *Object Recognition in Man, Monkey, and Machine*. MIT Press/Elsevier, 1999.
30. A. Torsello and E. R. Hancock. Computing approximate tree edit-distance using relaxation labelling. *Workshop on Graph-based Representations in Pattern Recognition*, pages 125–136, 2001.
31. J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40:175–179, 1991.
32. P. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. *SODA*, pages 311–321, 1993.
33. L. Younes. Computable elastic distance between shapes. *SIAM J. Appl. Math.*, 58:565–586, 1998.
34. S. C. Zhu and A. L. Yuille. FORMS: A flexible object recognition and modeling system. *IJCV*, 20(3):187–212, 1996.