Application of Planar Shape Comparison to Object Retrieval in Image Databases

Longin Jan Latecki and Rolf Lakämper

Dept. of Applied Mathematics, University of Hamburg,

Bundesstr. 55, 20146 Hamburg, Germany

Emails: {latecki, lakaemper}@math.uni-hamburg.de

December 30, 1999

Abstract

A similarity measure for silhouettes of 2D objects is presented, and its properties are analyzed with respect to retrieval of similar objects in image databases. To reduce influence of digitization noise as well as segmentation errors the shapes are simplified by a new process of digital curve evolution. To compute our similarity measure, we first establish the best possible correspondence of visual parts (without explicitly computing the visual parts). Then the similarity between corresponding parts is computed and summed. Experimental results show that our shape matching procedure gives an *intuitive* shape correspondence and is stable with respect to noise distortions.

Keywords: polygonal boundary representation, shape similarity measure, visual parts, discrete curve evolution.

1 Introduction

With the recent increase in image and multimedia databases, there has been an acceleration of research in developing and applying shape similarity measures, e.g., for shape-based retrieval of similar objects (see Forsyth et al. [4]). In computer vision there is a long history of work on shape representation and shape similarity. However, most of the existing methods have only a very limited possible application to distributed image databases, since the shape of objects must be restricted and known a priori. These methods are based on the close word assumption, which means that the application domain must be explicitly known, since prior knowledge of the application domain is necessary for parameter adjustment. Moreover, many of the existing approaches are very sensitive to noise. To systematize our discussion, we first suggest some necessary requirements for shape similarity measures that are used for retrieval of similar objects in distributed image databases. Then, we briefly review some of the existing approaches from the perspective of these requirements.

A shape similarity measure useful for shape-based retrieval in image databases should be in accord with our visual perception. This basic property leads to the following requirements:

(1) A shape similarity measure should permit recognition of perceptually similar objects that are not mathematically identical.

(2) It should not be effected by distortions (e.g., digitization noise and segmentation errors).

(3) It should preserve significant visual parts of objects.

(4) It should not depend on scale, orientation, and position of objects.

If we want to apply a shape similarity measure to distributed image databases, where the object classes are generally unknown a priori (e.g., in the Internet), it is necessary that:

(5) A shape similarity measure is universal, in the sense that it allows us to identify or distinguish objects of arbitrary shapes, i.e., no restrictions on shapes are assumed.

Since requirements (1), (2), and (3) are of a cognitive nature, their satisfaction should be justified by cognitive experiments. Requirements (4) and (5) are of a purely mathematical nature and their satisfaction should be shown by mathematical arguments. A motivation for requirement (3) is given in Siddiqi et al. [22],

> "*Part-based representations allow for recognition that is robust in the presence of occlusion, movement, growth, and deletion of portions of an object, and play an important role in theories of object categorization and classification.*"

There is a strong evidence for part-based representations in human vision, see e.g., [22, 7]. Hoffman and Richards [6] provide strong evidence that contours are psychologically segmented into visual parts at negative curvature minima. However, computation of negative minima of curvature as well as other extremal points is not robust in real digital images, since they are obtained by local computation. Although remarkable progress has been made on this matter, the robust computation of extremal points in real digital images is an open problem.

Due to requirement (5), methods based on vectors of shape parameters like area, perimeter, elongation (major axis / minor axis), etc. are excluded. Vectors of shape parameters may be very useful for shape classification, but not as a basis for shape similarity measures, since, as argued in Mokhtarian et al. [15],

> *"common shapes need hundreds of parameters to be represented explicitly"*,

and most of these parameters are probably unknown. An argument in this direction is also given in Sclaroff [19]:

> *"In the last few years, researchers have made some progress toward automatic shape indexing for image databases. The general approach has been to calculate some approximately invariant statistic like shape moments, and use these to stratify the image database ...*
>
> *One problem with this general approach is that it discards significant perceptual and semantic information. While indexing methods provide a means to*

quickly narrow a search to a more manageable subset, they often do not provide a method for closer, direct comparison how they are related. Rather than discarding useful similarity information by employing only invariants, we believe that one should use a decomposition that preserves as much semantically meaningful and perceptually important information as is possible, while still providing an efficient encoding of the original signal [18]. "

Also parametric methods that describe certain object classes e.g., B-spline surface patches of car prototypes, are excluded by requirement (5), since they require explicit assumptions about the category of curves to be represented.

Requirement (2) that a universal shape similarity measure must be unaffected by noisy distortions and irrelevant shape features excludes shape representations based on global transformations like boundary curve description by Fourier coefficients in Zahn and Roskies [23] and moment features in Hu [8], since as argued in Nishida [17]:

"However, there are the following drawbacks in using these features:

(1) High-order features, i.e., high order-coefficients of Fourier expansions of closed curves or high-order moments of two-dimensional images, are required for shape classifications, leading to expensive computations.

(2) High-order features are sensitive to noise, whereas only low-order ones are robust and stable.

(3) Even if image patterns can be classified with these features, point correspondences or shape transformation parameters cannot be obtained explicitly."

In this paper we present a shape similarity measure that satisfies requirements (1)-(5). We demonstrate this by theoretical considerations as well as by experimental results. We present here only a small fraction of our experimental results. We develop a system for retrieval of similar objects based on our measure, where a user query can be given either

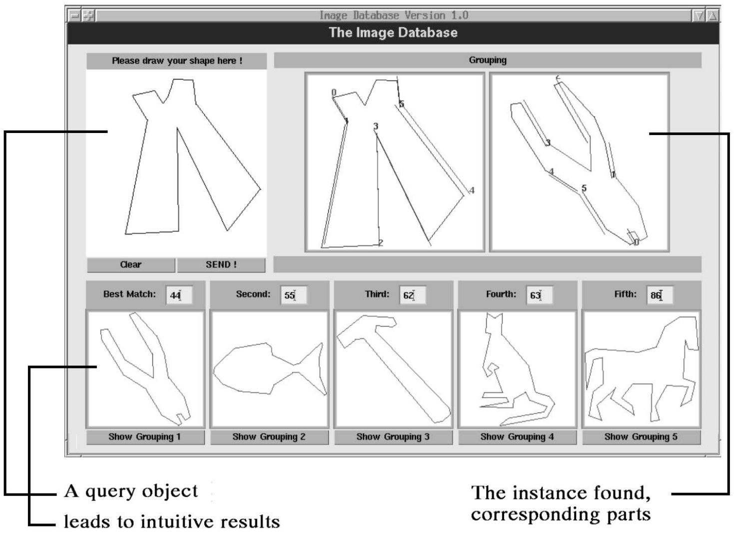


Figure 1: Retrieval of similar objects based on our similarity measure.

by a graphical sketch or by an example silhouette, see Figure 1. We apply our system to retrieve similar objects in various databases of object contours.

In Sections 3 and 4 our shape similarity measure is defined for object contours. Our approach of defining a shape similarity measure is related to that of Arkin et al. [1], where comparison of polygonal curves is based on L_2 distance of their turn angle representations (which we call tangent functions). A more detailed comparison is given at the beginning of Section 6. The main difference is that our shape similarity measure is based on a subdivision of objects into parts of visual form.

In Sections 6 and 7 we give a cognitive justification that our measure satisfies requirements (1), (2), and (3). We compare in Section 6 the performance of our measure to well-known shape similarity measures that have been justified by cognitive experiments. In Section 7 we describe further experiments that yield results corresponding to intuition.

Requirements (4) and (5) are of a purely mathematical nature and their satisfaction can be shown by simple arguments. The satisfaction of requirements (5) follows from the fact that we represent object boundaries as simple closed polygonal curves and that our shape similarity measure allows us to compare any two such curves. We simply obtain the polygonal curves from the boundary chain code (without any smoothing or other preprocessing) of segmented objects in digital images. Thus, every object contour in a digital image, can be represented as a simple closed polygonal curve (with a possibly large number of vertices) without loss of information and without any additional computation.

In Section 2 we overview our discrete curve evolution. Since the content of Sections 3–7 does not depend on Section 2, the reader not interested in shape simplification, can continue with the definition of our shape similarity measure in Section 3.

2 Discrete Curve Evolution

Since contours of objects in digital images are distorted by digitization noise and segmentation errors, it is desirable to neglect the distortions while at the same time preserving the perceptual appearance at a level sufficient for object recognition. Therefore, in the context of image databases our similarity measure is applied to contours whose shape has been previously simplified by a discrete curve evolution. This allows us (see Figure 2)

- to reduce the influence of noise and
- to simplify the shape by removing *irrelevant* shape features without changing *relevant* shape features,

which contributes in a significant way to the fact that the similarity measure satisfies requirements (1) and (2). Observe that our discrete curve evolution is context sensitive, since whether shape components are relevant or irrelevant cannot be decided without context. Our discrete curve evolution is briefly presented in this section (more detailed presentations are given in [11, 12]).

Our curve evolution method does not require any control parameters to achieve the task of shape simplification, i.e., there are no parameters involved in the process of the discrete curve evolution. However, we clearly need a stop parameter, which is the number of iterations the evolution is to be performed. This parameter is automatically determined in accord with our visual perception by the procedure described in Section 5.

Since any digital curve can be regarded as a polygon without loss of information (with possibly a large number of vertices), it is sufficient to study evolutions of polygonal shapes.

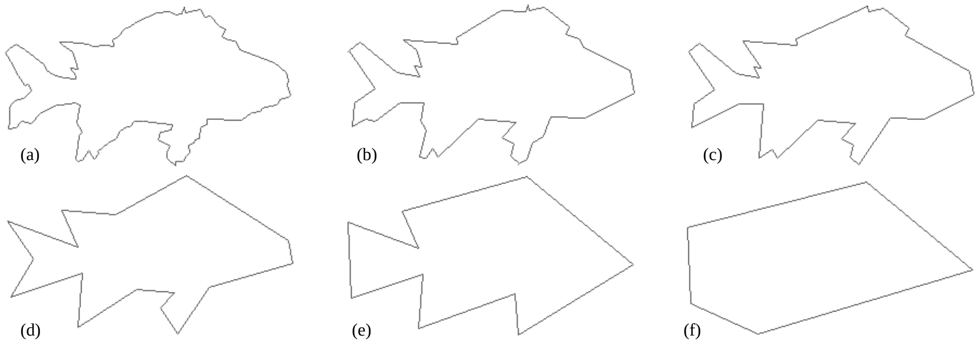


Figure 2: A few stages of the discrete curve evolution. (a) is a distorted version of the contour in WWW-page http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html.

The basic idea of the proposed evolution of polygons is very simple:

- In every evolution step, a pair of consecutive line segments s_1, s_2 is replaced by a single line segment joining the endpoints of $s_1 \cup s_2$.

The key property of this evolution is the order of the substitution. The substitution is done according to a relevance measure K given by

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)}, \tag{1}$$

where $\beta(s_1, s_2)$ is the turn angle at the common vertex of segments s_1, s_2 and l is the length function normalized with respect to the total length of a polygonal curve C. The main property of this relevance measure is the following

- The higher the value of $K(s_1, s_2)$ the larger is the contribution to the shape of the curve of arc $s_1 \cup s_2$.

A cognitive motivation of this property is given in [11], where a detailed description of our discrete curve evolution can also be found. Online demonstrations can be viewed on our www-site [14].

It is a simple and natural observation that maximal convex parts of objects determine visual parts. The fact that visual parts are somehow related to convexity has been mentioned in the literature, e.g., Basri et al. [2] state

> "*Parts generally are defined to be convex or nearly convex shapes separated from the rest of the object at concavity extrema, as in Hoffman and Richards [6], or at inflections, as in Koenderink and Doorn [10].*"

Although the observation that visual parts are *"nearly convex shapes"* is very natural, the main problem is to determine the meaning of *"nearly"* in this context. Many significant visual parts are not convex in the mathematical sense, since a visual part may have small concavities, e.g., small concavities caused by fingers in the human arm. Thus, a natural and simple idea is to compute significant convex parts while neglecting small concavities.

Our solution is based on the discrete curve evolution method in which a significant visual part will become a convex part at some level of the evolution. If a significant visual part contains concavities, then the corresponding boundary arcs are concave arcs (with respect to the object). Since the relevance measures of these concave arcs are smaller than the relevance measure of the boundary arc of the significant visual part, the concavities will disappear in an earlier stage of the boundary evolution. Thus, there exists an evolution stage at which a significant visual part is a convex part, i.e., it is enclosed by a convex boundary arc (with respect to the object). A few stages of our curve evolution are illustrated in Figure 2. For example, the two small fins become convex in (d) and the fish tail becomes convex in (e). These parts are enclosed by maximal convex boundary arcs. Due to an important property of our curve evolution that the remaining vertices do not change their position, we can identify visual parts on the original contour as maximal convex boundary arcs obtained in the course of the evolution. In particular, this means that the position of the endpoints of a convex boundary arc obtained in the course of the evolution is exactly the same as the position of the endpoints on the original contour. We base our approach to shape decomposition into visual parts on the following rule:

- **Hierarchical convexity rule:**
  The maximal convex arcs (of the object) obtained at various stages of the contour evolution determine parts of the object boundary that enclose visual parts of the object.

The parts of boundaries obtained by the hierarchical convexity rule correspond for many objects to the parts obtained using points of minimal negative curvature (Hoffman and Richards [6]). This is the case when the endpoints of convex arcs are located near the points of minimal negative curvature. Also for many objects, the resulting parts of objects correspond to limbs and necks in the theory of Siddiqi and Kimia [20]. For boundaries of continuous objects, the endpoints of maximal convex arcs correspond to inflection points (e.g., inflection points are used for shape description in Freeman [5]). The correspondence of the endpoints of maximal convex arcs to different kinds of critical points (in the sense of differential geometry) is possible, since we work in a discrete space which does not exactly follow the rules of differential geometry. The correspondence of the visual parts obtained by our hierarchical convexity rule to the well-known visual parts, which are justified by many psychological experiments [6, 22, 7], gives cognitive motivation for the proposed rule.

In our approach, we do not need to decide which critical points have to be joined together in order to obtain object parts. The object parts obtained by joining two chosen negative curvature minima are called *part cuts* (Beusmans et al. [3]). As argued in Hoffman and Singh [7], a separate theory is necessary to determine the part cuts knowing the boundary points of minimal negative curvature. In our approach, the part cuts are simply the parts enclosed by maximal convex arcs. For further details see [11] or our www-page [14], where some online demos are also presented.

3 Shape Similarity Measure

In this section we define our shape similarity measure. In the context of image databases, this measure is applied to contours which have been previously simplified by the discrete curve evolution described in Section 2. The appropriate evolution stage is selected for each shape, and then the similarity is computed for the resulting instances of the shapes.

Our similarity measure profits from the decomposition into visual parts based on convex boundary arcs. The key idea is to find the right correspondence of the visual parts. We assume that a single visual part (i.e., a convex arc) of one curve can correspond to a sequence of consecutive convex and concave arcs of the second curve, e.g., part number 0 of the top-left fish contour in Fig. 3. This assumption is justified by the fact that a single visual part should match to its noisy versions that can be composed of sequences of consecutive convex and concave arcs, or by the fact that a visual part obtained at a higher stage of evolution should match to the arc it originates from. Since maximal convex arcs determine visual parts, this assumption guarantees preservation of visual parts (without explicitly computing visual parts).

In this section, we assume that polygonal curves are simple, i.e., there are no self-intersections, and they are closed. We assume also that we traverse polygonal curves in the counter clockwise direction.

Let $convconc(C)$ denote the set of all maximal convex or concave subarcs of a polygonal curve C. Then the order of traversal induces the order of arcs in $convconc(C)$.

Since a simple one-to-one comparison of maximal convex/concave arcs of two polygonal curves is of little use, due to the facts that the curves may consist of a different number of such arcs and even similar shapes may have different small features, we join together maximal arcs to form *groups*:

A **group** g of curve C is a union of a (non-empty) consecutive sequence of arcs in $convconc(C)$. Thus, g is also a subarc of C. We denote $groups(C)$ the set of all groups of C. We have $convconc(C) \subseteq groups(C)$.

A **grouping** G for a curve C is an ordered set of consecutive groups $G = (g_0, ..., g_{n-1})$ for some $n \geq 0$ such that

- $g_i \cap g_{i+1(mod\ n)}$ is a single line segment for $i = 0, ..., n-1$.

Since any two consecutive groups intersect in exactly one line segment, the whole curve C is covered by G. We denote the set of all possible groupings G of a curve C as $\mathcal{G}(C)$. Figure 3 shows sample groupings of the given contours, where each group is assigned a different number.

Given two curves C_1, C_2, we say that groupings $G_1 \in \mathcal{G}(C_1)$ and $G_2 \in \mathcal{G}(C_2)$ **corresponds** if there exists a bijection $f : G_1 \to G_2$ such that

1. f preserves the order of groups and

2. for all $x \in G_1$ $x \in convconc(C_1)$ or $f(x) \in convconc(C_2)$.

We call the bijection f a **correspondence** between G_1 and G_2. We denote the set of all corresponding pairs (G_1, G_2) in $\mathcal{G}(C_1) \times \mathcal{G}(C_2)$ by $\mathcal{C}(C_1, C_2)$. Two example correspondences are shown in Fig. 3. The condition that any f is a bijection means that both curves are decomposed into the same number of groups. Condition (2) means that at least one of the corresponding groups $x \in G_1$ or $f(x) \in G_2$ is a maximal (convex or concave) arc. The reason is that we want to allow mappings between one-to-many maximal arcs or many-to-one maximal arcs, but never between many-to-many maximal arcs. Since maximal convex arcs determine visual parts, condition (2) guarantees preservation of visual parts (without explicitly computing visual parts). Condition (2) implies also that every maximal (convex or concave) arc in a higher stage of abstraction, will match to the sequence of arcs it originates from.

A **similarity measure** for curves C_1, C_2 is defined as

$$S_c(C_1, C_2) = \min\{\sum_{x \in G_1} S_a(x, f_{(G_1,G_2)}(x)) : (G_1, G_2) \in \mathcal{C}(C_1, C_2)\}, \qquad (2)$$

where $f_{(G_1,G_2)}$ is the correspondence between G_1 and G_2 and S_a is a similarity measure for arcs that will be defined in the next section. To compute $S_c(C_1, C_2)$ means to find in the set $\mathcal{C}(C_1, C_2)$ of all corresponding groupings a pair of groupings for which the sum of the differences between the corresponding groups $S_a(x, f_{(G_1,G_2)}(x))$ is minimal.

The task of computing the similarity measure defined in (2) can be formulated as a global minimum problem:

Given a function

$$\mathcal{M}(X, Y) = \sum_{x \in X} S_a(x, f_{(X,Y)}(x))$$

that assigns a group matching value to every corresponding pair $(X,Y) \in \mathcal{C}(C_1,C_2)$ related by the correspondence $f_{(X,Y)}$, find a pair $(G_1,G_2) \in \mathcal{C}(C_1,C_2)$ for which $\mathcal{M}(G_1,G_2)$ is minimal, i.e., $\mathcal{M}(G_1,G_2) \leq \mathcal{M}(X,Y)$ for all elements $(X,Y) \in \mathcal{C}(C_1,C_2)$.

The similarity measure defined in (2) is computed using dynamic programming. Numerous experimental results show that it leads to intuitive arc correspondences, e.g., see Figure 3. We have applied a slightly modified version of the shape similarity measure in (2) to automatic object indexing and searching in image databases. The experimental results are described in Section 6.

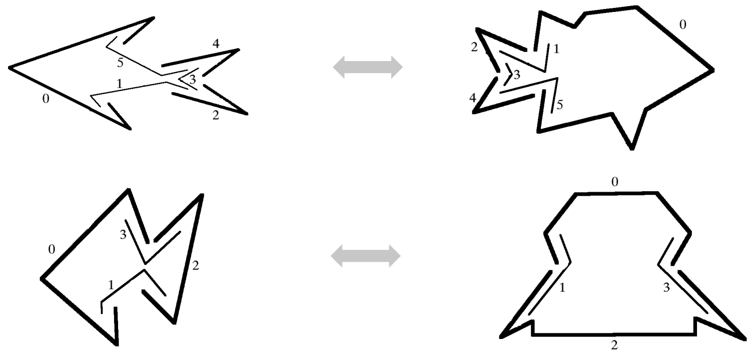


Figure 3: The corresponding arcs are labeled by the same numbers.

4 Similarity of Polygonal Arcs

The goal of this section is to define the similarity measure S_a for arcs that is part of the definition of the shape similarity measure in Section 3. As mentioned in the introduction, any digital curve C can be interpreted as a polygonal curve with a possibly large number of vertices without loss of information. We assign to a every polygonal curve a *tangent function*, which is a step function. We use the tangent function as a basis for the proposed similarity measure of simple polygonal arcs.

Let C be a polygonal curve. We treat it as a function $C : [0,1] \to \mathbb{R}^2$, i.e., the length of C is rescaled to 1. The **tangent function** of C (which is also called a *turning function*) is a multi-valued function $T(C) : [0,1] \to [0, 2\pi]$ defined by $T(C)(s) = C'_{-}(s)$ and $T(C)(s) =$

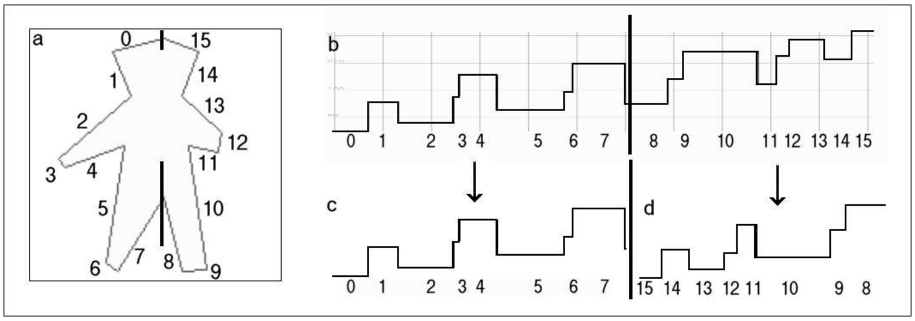


Figure 4: A polygonal curve (a) and its tangent function (b). The similarity of (c) and (d) shows that the curve (a) is symmetric.

$C'_+(s)$, where $C'_-(s)$ and $C'_+(s)$ are left and right derivatives of C. For example, see Figures 4(a) and (b). Clearly, only if $C(s)$ is a vertex of the polygon, $C'_-(s) \neq C'_+(s)$. The y-difference between two adjacent steps of the tangent function represents the turn angle of the corresponding pair of line segments.

Now we define the similarity measure for arcs. Let c, d be simple polygonal arcs that are parts of closed curves C, D. We denote by $T(c), T(d)$ their tangent functions uniformly scaled so that their projections on the x-axis $\pi_x(T(c))$ and $\pi_x(T(d))$ both have length one. The arc similarity measure is given by (e.g., see Figure 5)

$$S_a(c,d) = \left(\int_0^1 (T(c)(s) - T(d)(s) + \theta_0)^2 \, ds \right) \max(l(c), l(d)) \max(\frac{l(c)}{l(d)}, \frac{l(d)}{l(c)}) \qquad (3)$$

where l is the relative arclength of an arc with respect to the boundary length of the curve it is part of and θ_0 is defined below. The integral in (3) is weighted with the arc length penalized by the difference in length of the corresponding parts. For example, if $l(c) > l(d)$, then the scaling term is equal to $l(c)\frac{l(c)}{l(d)}$, where $l(c)$ scales the value of the integral by the relative arclength of arc c with respect to the length of curve C and $\frac{l(c)}{l(d)}$ is the penalty for the relative length difference of arcs c and d.

The constant θ_0 is a translation of $T(d)$ that minimizes the integral, i.e.,

$$\int_0^1 (T(c)(s) - T(d)(s) + \theta_0)^2 \, ds = \inf_{\theta \in [0, 2\pi)} \int_0^1 (T(c)(s) - T(d)(s) + \theta)^2 \, ds.$$

The constant θ_0 exists and is given by Lemma 3 in Arkin et al. [1].

Observe that we apply measure (3) with the restriction that $c \in convconc(C)$ or $d \in convconc(D)$.

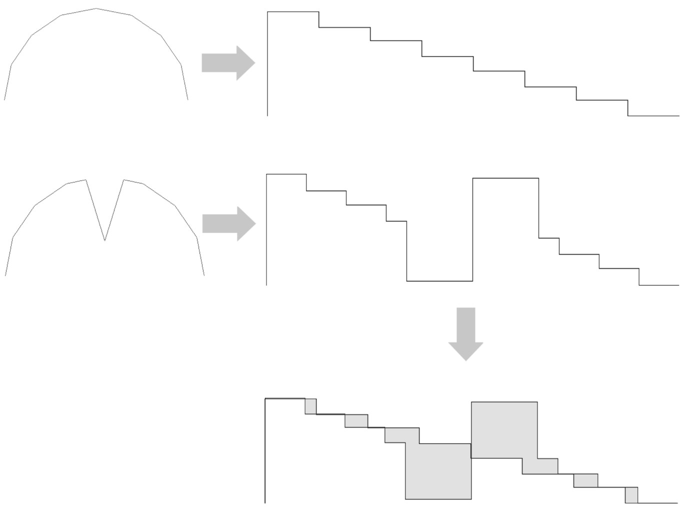


Figure 5: $D_{L_2}(T(c), T(d))$

5 An Optimal Evolution Stage

Now we describe the procedure that determines the stop parameter i.e., a stage at which the curve evolution halts. The evolved contours obtained at this stage are used as input to our shape similarity measure.

Let $P = P^0, ..., P^m$ be polygons obtained from a polygon P in the course of discrete curve evolution such that P^m is the first convex polygon.

For $i = m$ with a step -1 do:
The boundary of a high abstraction level P^i is segmented into maximal convex/concave parts. These parts are compared to their corresponding parts on the original polygon P^0, where the corresponding parts are the ones having the same endpoints. The comparison is done using the S_a-measure. If the comparison of a single part in P^i leads to a value higher than a given threshold s, the shape P^i is abstracted too much, and the previous abstraction level P^{i-1} is taken. The procedure repeats until the stage i is reached such that the comparison of all parts in P^i to all their corresponding parts is lower than s.

Figure 6 shows some examples of automatic level abstraction. Notice that the comparison of the boundary parts to their corresponding segments in different abstraction levels uses the property of the evolution process, that the set of vertices of a shape at abstraction level n is a subset of all sets of vertices of the shape at abstraction levels $0, .., n-1$, hence it is easy to find the corresponding boundary parts. The computation time of this procedure is not critical, since the level of abstraction is computed off-line for every shape in the database. Using a fixed threshold s results in an intuitive correct abstraction level in most experiments (a few hundred shapes of different complexity), but cannot be the final solution. Future work will take into account more statistical data for a dynamic threshold adjustment.

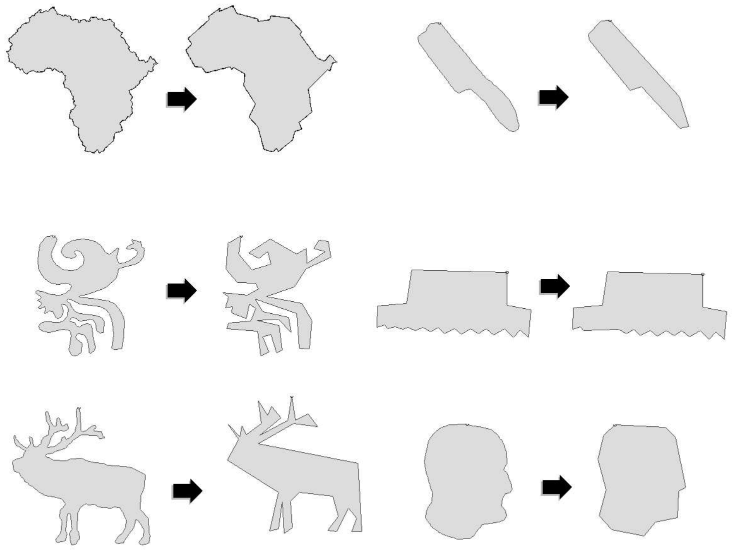


Figure 6: Some examples of the automatic abstraction level that is used as input to our shape similarity measure.

6 Comparison to Known Similarity Measures

We restrict our comparison to universal similarity measures that are translation, rotation, reflection, and scaling invariant. This excludes, for example, Hausdorff distance [9], which is universal but does not satisfy these requirements.

As stated in the introduction, our approach to defining a shape similarity measure is related to the one in Arkin et al. [1], where L_2 distance of tangent functions is used for comparing polygonal shapes. The main difference of our approach is that we use L_2

distance for comparing parts of polygonal shapes, which makes our approach more robust with respect to local distortions that result in non-uniform stretching of boundary parts. This is illustrated by comparing the two shapes in Figure 7.

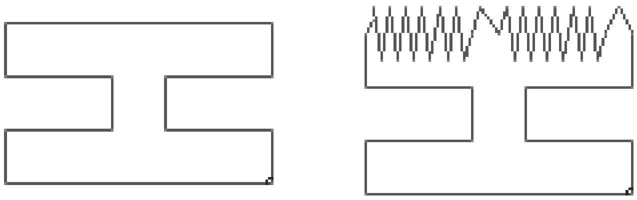


Figure 7: A shape and its locally distorted version.

In Figure 8 we see tangent functions of both shapes from Figure 7 scaled to the same length. The local distortions on one side of the second shape result in stretching of the tangent function of this side. The gray region in Figure 8 shows the side of the first shape that should correspond to the distorted side of the second shape.

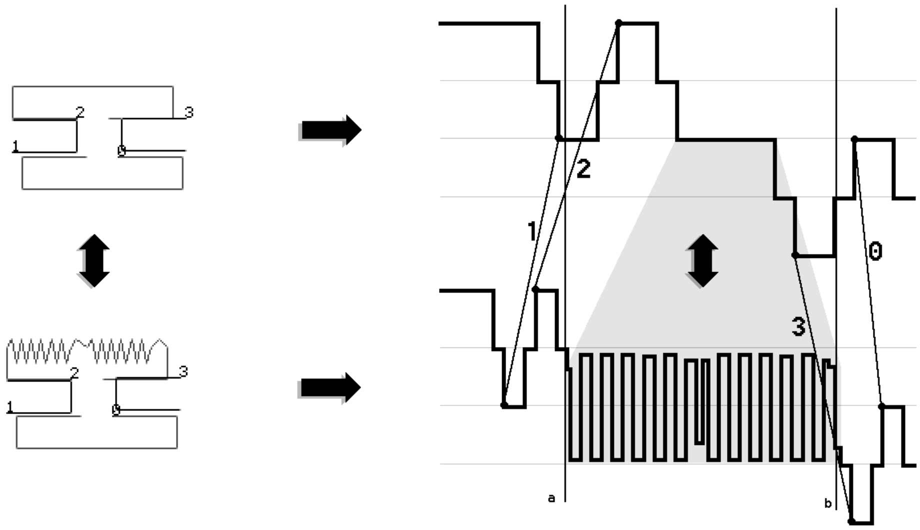


Figure 8: The gray region shows the side of the first shape that should correspond to the distorted side of the second shape.

In order to compute the distance between two shapes, Arkin et al. translate one tangent function with respect to the other so that their L_2 distance is minimal, but they do not allow the functions to be stretched. Consequently, the part of the tangent function of the second shape that represents the distorted side will correspond to the part of the tangent function of the first shape bounded with lines a and b in Figure 8. This is illustrated in Figure 9, where both functions are shown. It can be seen that there does not exist any translation of the functions that yields the correct part correspondence. In particular, the optimal translation determined by Arkin et al. does not yield the correct part correspondence. Thus, using the distance defined in [1], the distorted side is compared to about half of the boundary of the first shape, which clearly results in a very bad similarity value. The same phenomenon will take place if we do not scale the tangent functions to the same length, i.e., if we compare them using their original lengths.

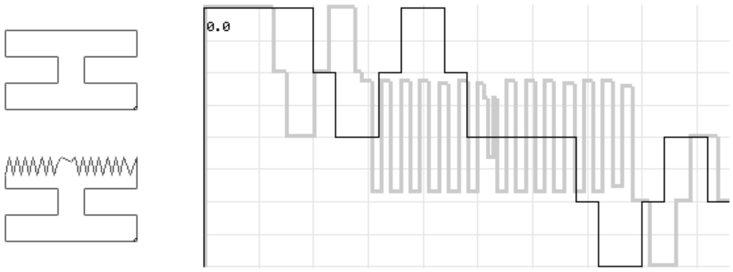


Figure 9: A comparison of tangent functions of two contours based on Arkin et al.

To compute our shape similarity measure, we first establish the best possible correspondence of the maximal convex/concave arcs. The corresponding maximal convex/concave arcs are labeled with the same numbers 0 to 3 in Figure 8. The maximal convex arc 3 of the first shape will correctly correspond to the part 3 of the second shape (which is the part between lines 2 and 3 on the tangent functions).

In our approach the comparison of tangent functions is done for each pair of corresponding parts separately. We scale each corresponding pair to the same length 1, and finally compute the distance of the local tangent functions obtained in this way. This is shown in Figure 10, where the tangent functions for parts 0, 1, and 3 are identical and the L_2 distance of the tangent functions for parts 2 is relatively small. This results in a relatively small value of our similarity measure for the two shapes. This example also demonstrates that our similarity measure satisfies requirements (1), (2), and (3) described in the introduction. We want to stress that the shape of both objects in this example was not simplified before the comparison.

An interesting approach to establish desirable properties of shape similarity measures is

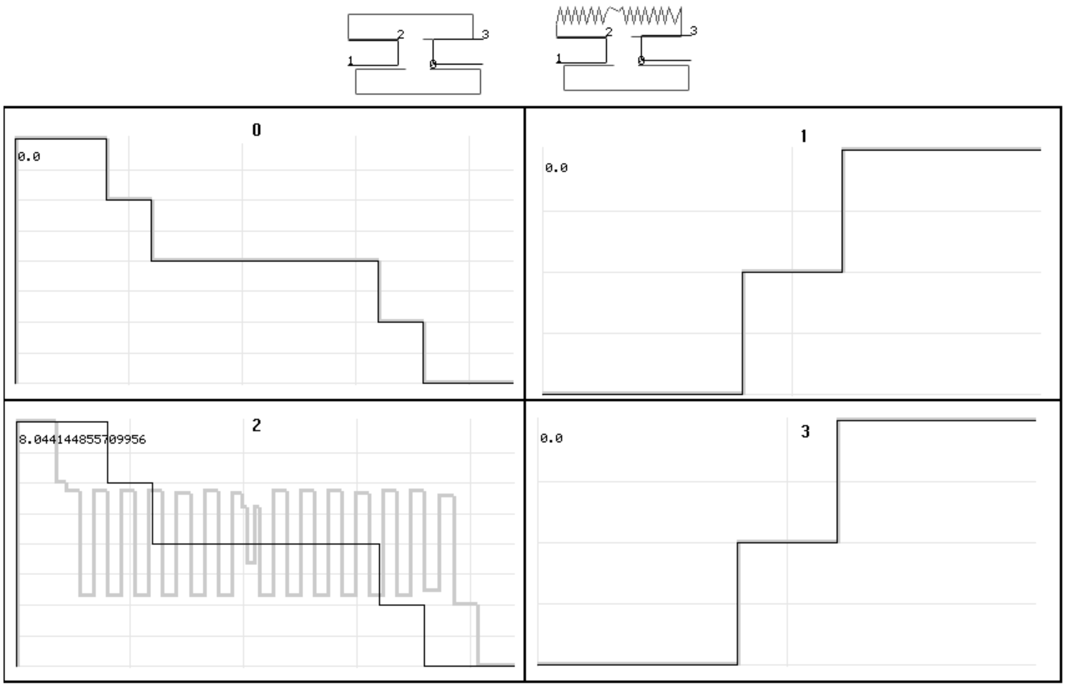


Figure 10: In our approach the comparison of tangent functions of two contours is done for each pair of corresponding parts (0, 1, 2, and 3) separately.

given in Basri et al. [2]. The desirable properties are illustrated and tested on three proposed similarity measures: spring model, linear model, and continuous deformation model. These models measure deformation energy needed to obtain one object from the other. The calculation of deformation energy is based on (best possible) correspondence of boundary points and local distortions of corresponding points as a function of local curvature differences. Thus, the calculation of the three measures requires local computation of curvature. The similarity measures in Basri et al. [2] are obtained as the integral of local distortions between corresponding contour points. The authors themselves point out a counter-intuitive performance of their measures when applied to the objects like the ones the first row in Figure 11 (Figure 17 in [2]). The H-shaped contour (a) is compared to two different distortions of it. Although the shape (b) appears more similar to (a) than the shape (c), the amount of local distortion to obtain (b) and (c) from (a) is the same. Therefore, all three measures presented in [2] imply that shapes (b) and (c) are equally similar to (a). Basri et al. argue that this counter-intuitive performance is due to the fact that their measures are based on contour representation of shapes. We do not agree with the fact that the counter-intuitive performance of measures in Basri et al. [2] is due to contour representation. The performance of our measure clearly proves that this is not the case:

Our similarity measure is based on contour representation and gives similarity values in accord with visual perception. Our measure yields $S_c((a),(b)) = 368$ and $S_c((a),(c)) = 518$, i.e., (b) is more similar than (c) to (a). The main difference is that our measure is not based on local properties, i.e., it is not based on correspondence of contour points and their local properties, but on correspondence of contour parts.

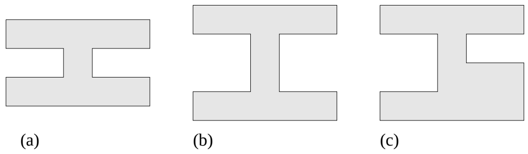


Figure 11: Our similarity measure yields results in accord with our visual perception: $S_c((a),(b)) = 368$ and $S_c((a),(c)) = 518$.

Among others, Basri et al. [2] argue that similarity measures should be sensitive to structure of visual parts of objects. To check this property, they suggest that bending an object at a part boundary should imply less changes than bending in the middle of a part. This property of our measure is illustrated in Figure 12. A more detailed comparison of our measure to the one presented in Basri et al. [2] is given in [13], where we also show that our measure satisfies constrains on similarity functions presented in [2].

The approach described in Sclaroff [19] is based on distance to object prototypes repre-

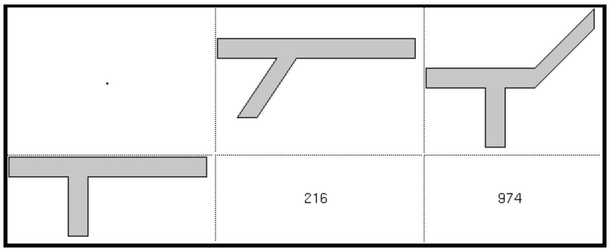


Figure 12: The results of our similarity measure on shapes similar to the ones in Table 2 in Basri et al. [2].

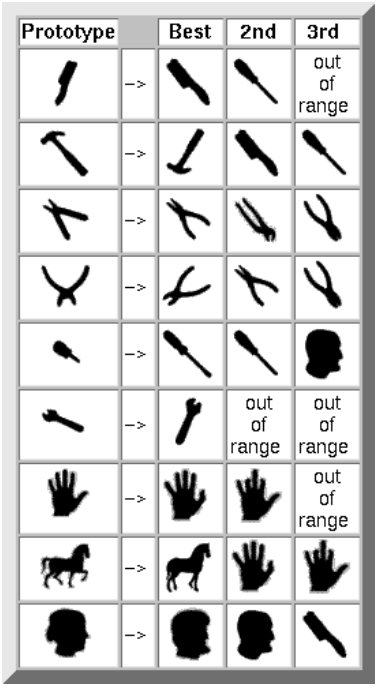


Figure 13: The most similar objects (excluding self similarity) to objects from Figure 18.

senting classes of shapes. Shape similarity is computed in terms of the amount of strain energy needed to deform one object into another. Therefore, the above discussion of approaches based on deformation energy in [2] applies also to [19]. Additionally, the computation of the shape similarity in [19] requires establishing a direct point correspondence and shape alignment, which is a highly nontrivial task. Sclaroff uses Hausdorff distance [9] to achieve this task in his experiments.

We also compared the results of our approach with an approach presented in Siddiqi et al. [21] that is based on a hierarchical structure of shocks in skeletons of 2D objects. In this approach object shape is represented as a graph of shocks. The similarity of objects is determined by a similarity measure of the graphs of shocks. Although the shape representation in [21] is not based on boundary curves, the results of our similarity measure are very similar to the results in [21]. This is demonstrated in Figure 18, whose objects are scanned from Table 1, p. 27, in [21]. To demonstrate that the performance of our measure is in accord with intuition, we show in Figure 13 the most similar objects to objects from Figure 18.

Further we compared our approach to retrieval of similar objects through a similarity measure based on curvature scale space in Mokhtarian et al. [15]. The curvature scale space representation is obtained by curve evolution guided by a diffusion equation [16]. The similarity measure in [15] is applied to a database of marine animals in which every image contains one animal. We applied our similarity measure to the same database. The results of our similarity measure applied to the same objects as in [15] are presented in Figure 19. The query objects (marked with number one) are the same as objects in Figures 4(a), 4(d), 5(a), and 7(a) in [15]. The results are very similar but not identical to the results in [15].

7 Experimental Results

When comparing shapes in image databases we have to deal not only with distortions caused by noise but also with the change of object view due to change of perspective and due to motion, e.g., bending of object parts or change of a relative position among parts.

In this section, we present experimental results that illustrate that our shape similarity measure is robust with respect to all these distortions. This means that even a substantial amount of these distortions will result in small changes of the similarity values.

All figures corresponding to the experiments of shape comparison show abstracted shapes of original input images at an automatically derived stage of abstraction. Observe that although the abstracted versions are used to find the part correspondence and to compute the similarity values, it is no problem to backtrace the corresponding parts to the original shapes.

In all figures the corresponding visual parts obtained in the course of computation of our similarity measure are indexed with the same numbers in the counterclockwise direction. For visual convenience the parts are drawn slightly displaced.

7.1 Perspective Distortion

The first experiment illustrates the behavior of our similarity measure in the context of perspective distortion which we simulate by three-dimensional rotation. The left column in Figure 14 shows three versions of the same shape *'hand'*A.

The shapes in the right column are obtained by three-dimensional rotation of shape A around the x-axis followed by the orthogonal projection to the x-y-plane. This results in a scaling in the vertical direction by the cosine of each angle. The rotation-angles used are: 45° for shape B, 70° for shape C, and 80° for shape D. Observe that the correct correspondence of visual parts is computed by our algorithm. The similarity values obtained by our shape similarity measure are relatively small: $S_c(A, B) = 381, S_c(A, C) = 2309, S_c(A, D) = 4237$. For comparison, the similarity value of hand A compared to horse A in Figure 16 is S_c(hand A, horse A) = 5068.

7.2 Part bending

The second experiment shows results of the comparison of three hands. Pairs A, B and A, C differ by bending of visual parts (i.e., fingers) in space. Thus, here we additionally have partial occlusion resulting from the perspective projection. Shapes B, C differ by relative position of fingers. Observe again the correct correspondence of visual parts. The similarity values obtained by our shape similarity measure are also relatively small: $S_c(A, B) = 1998, S_c(A, C) = 2080, S_c(B, C) = 519$.

The third experiment simulates movement of limbs against the main shape by two dimensional part bending. Shown are the comparisons of a horse in different movement steps (as a horse-lover will notice, the movements are not taken from a real horse but computed with a insufficient horse-simulation). The images differ from each other by non-perspective (two dimensional) bending of head, tail and legs. Again the correspondence of parts is correct and the resulting similarity values are relatively small: $S_c(A, B) = 2188, S_c(A, C) = 3471, S_c(B, C) = 1156$.

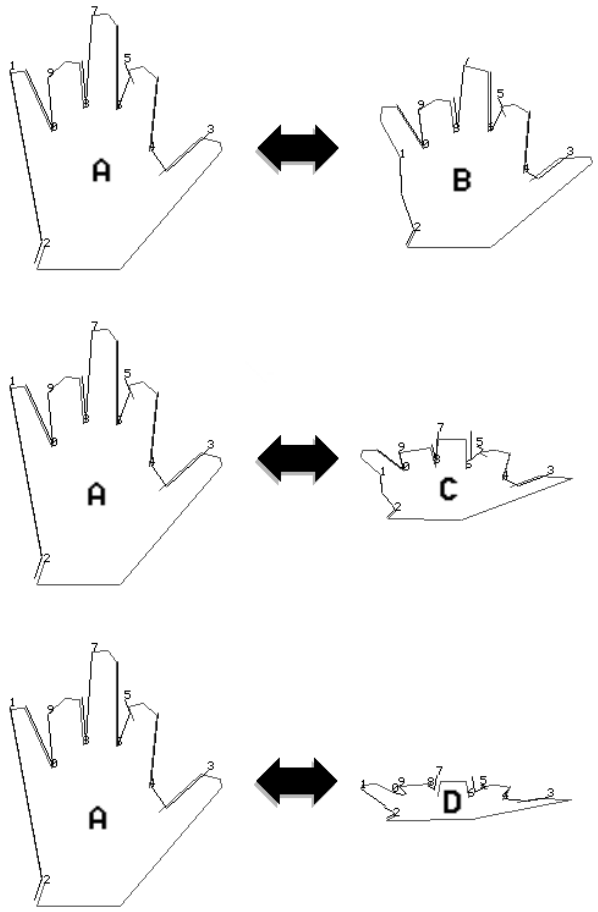


Figure 14: Shapes B, C, and D are perspective distortions of shape A. The similarity values are relatively small: $S_c(A, B) = 381, S_c(A, C) = 2309, S_c(A, D) = 4237$.

7.3 Distortions Resulting from Noise

In the puma-experiment, we compared shape A with two different distorted versions B and C. Version B has many albeit small distortions (i.e. displacements of original boundary points), version C has fewer distortions but larger ones. Finally we compared the two distorted versions B and C. Again the correspondence of parts is correct and the similarity values are relatively small: $S_c(A, B) = 1458, S_c(A, C) = 1698, S_c(B, C) = 1650$.

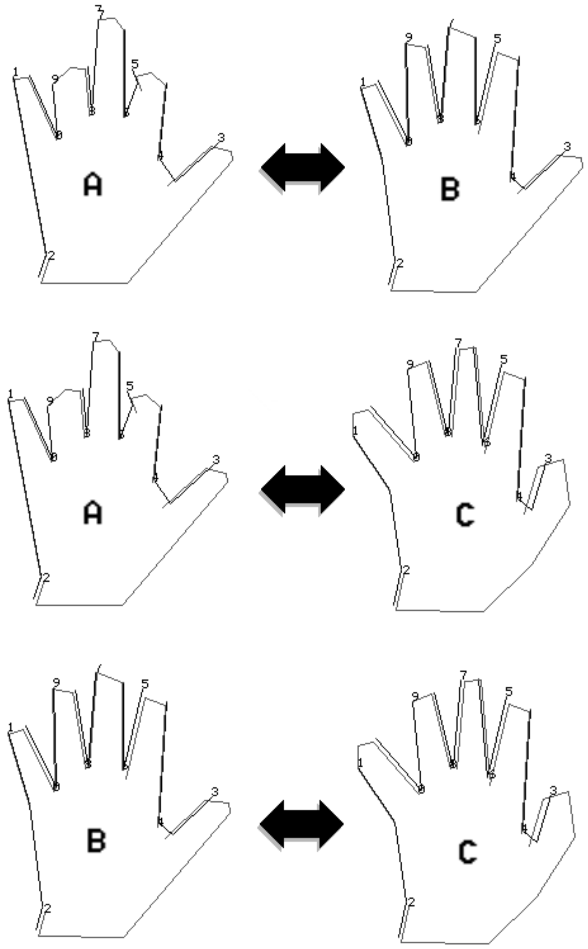


Figure 15: Similarity values by bending of visual parts: $S_c(A, B) = 1998, S_c(B, C) = 2080, S_c(B, C) = 519$.

8 Time Complexity and Processing Speed

An important criterion for databases is the time complexity and the processing speed in practice. For an analysis related to these topics, our algorithm must be divided into two stages, the shape abstraction and the comparison of the abstracted shapes.

Shape abstraction requires first sorting all vertices according to their relevance measure, which results in $O(n \log(n))$ time complexity with n denoting the number of vertices. Every single abstraction step is done by deleting one point (linear complexity), computing the new values of its neighbors and sorting them into the list (complexity: $log(n - s)$, where s denotes the number of previous abstraction steps). Hence the complexity of the

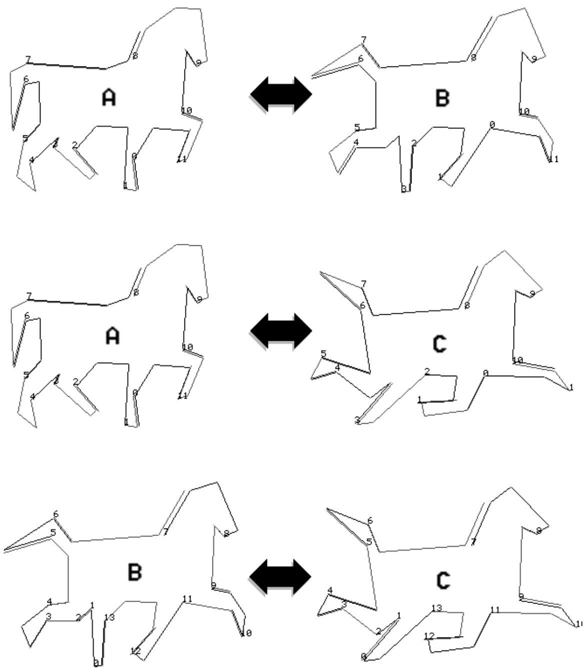


Figure 16: Similarity values of part movement: $S_c(A, B) = 2188, S_c(A, C) = 3471, S_c(B, C) = 1156$.

abstraction algorithm results in $O(n\log(n) + O(n) + O(log(n - s)) = O(n\log(n))$.

Processing speed: the algorithm was implemented in C on a 233MHz Pentium PC-computer. For example, it takes 5ms to compute all abstraction stages down to three vertices, using a shape boundary containing 290 vertices.

The computation of shape similarity measure is more time-expensive. First the optimal correspondence of maximal convex/concave arcs is computed using dynamical programming, the complexity for two shapes is $O(mn^2)$, where m, n denote the number of maximal convex/concave arcs of each boundary and $n \leq m$. To be invariant to rotation, one of the shapes must be rotated modulo the starting points of the convex/concave arcs, which is of order $O(n)$. This leads to a total complexity of $O(mn^3)$.

Processing speed: the algorithm was implemented in C++ on a 233MHz Pentium computer, the average comparison time (including rotation)of two objects consisting of about 15-20 maximal convex/concave arcs is 50ms.

The processing time can be drastically reduced if some pre-information about rotation

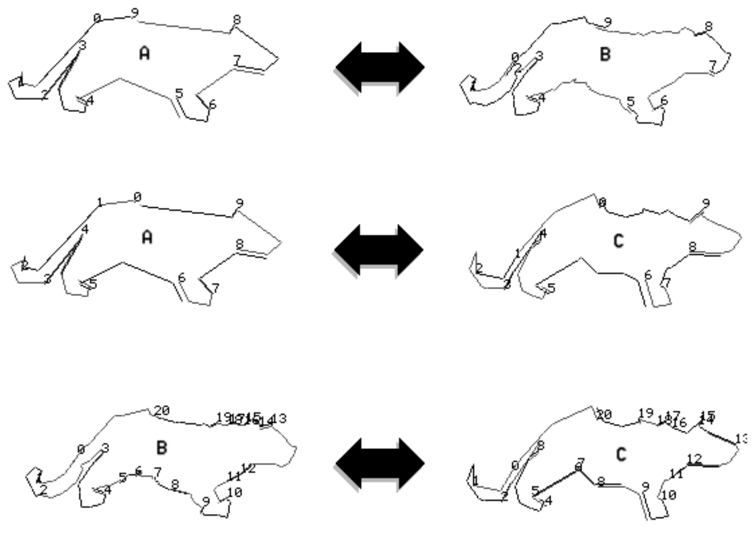


Figure 17: Similarity values in the presence of noise: $S_c(A, B) = 1458, S_c(A, C) = 1698, S_c(B, C) = 1650$.

is taken into account. The database system was designed to detect similarity between arbitrary objects, hence no pre-information was used for speed optimization.

9 Conclusions

We developed a shape similarity measure for contours of 2D objects that satisfies necessary requirements for cognitively motivated shape similarity measures. The main strength of our shape similarity measure is that it establishes the best possible correspondence of visually significant parts. A discrete evolution method that is used as a pre-filter for shape comparison yields significant visual parts as maximal convex arcs. We are presently working on an extension of our approach to object surfaces.

Acknowledgments

The work of Longin Jan Latecki was supported by a research grant Ec170/1-1 from the German Research Foundation (DFG) entitled *"Shape Representation in Discrete Struc-*

tures". The help of Prof. Ulrich Eckhardt (University of Hamburg) and Prof. Hans-Joachim Kroll (Technical University of Munich) in realization of this project is gratefully acknowledged. The helpful comments from Prof. Robert Melter are appreciated.

References

[1] M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. PAMI*, 13:209–206, 1991.

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

[3] J. Beusmans, D.D. Hoffman, and B.M. Bennett. Description of solid shape and its inference from occluding contours. *Journal of the Optical Society of America*, A. 4:1155–1167, 1987.

[4] D. Forsyth, J. Malik, and R. Wilensky. Searching for digital pictures. *Scientific American*, pages 88–93, June 1997.

[5] H. Freeman. Shape description via the use of critical points. *Pattern Recognition*, 10:159–166, 1978.

[6] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.

[7] D. D. Hoffman and M. Singh. Salience of visual parts. *Cognition*, 63:29–78, 1997.

[8] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory*, 8:179–187, 1962.

[9] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. PAMI*, 15:850–863, 1993.

[10] J. J. Koenderink and A. J. Doorn. The shape of smooth objects and the way contours end. *Perception*, 11:129–137, 1981.

[11] L. J. Latecki and R. Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73:441–454, 1999.

[12] L. J. Latecki and R. Lakämper. Polygon evolution by vertex deletion. In M. Nielsen, P. Johansen, O.F. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision. Proc. of Int. Conf. on Scale-Space'99*, volume LNCS 1682, Corfu, Greece, September 1999.

[13] L. J. Latecki and R. Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, submitted.

[14] L. J. Latecki, R. Lakämper, and U. Eckhardt. http://www.math.uni-hamburg.de/home/lakaemper/shape.

[15] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In A. W. M. Smeulders and R. Jain, editors, *Image Databases and Multi-Media Search*, pages 51–58. World Scientific Publishing, Singapore, 1997.

[16] F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. PAMI*, 14:789–805, 1992.

[17] H. Nishida. Matching and recognition of deformed closed contours based on structual transformation models. *Pattern Recognition*, 31:1557–1571, 1998.

[18] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. *Int. J. Computer Vision*, 18:233–254, 1996.

[19] S. Sclaroff. Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition*, 30:627–641, 1997.

[20] K. Siddiqi and B. B. Kimia. Parts of visual form: Computational aspects. *IEEE Trans. PAMI*, 17:239–251, 1995.

[21] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *Int. J. of Computer Vision*, to appear; http://www.cim.mcgill.ca/ siddiqi/journal.html.

[22] K. Siddiqi, K. J. Tresness, and B. B. Kimia. Parts of visual form: Psychophysical aspects. *Perception*, 25:399–424, 1996.

[23] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. on Computers*, 21:269–281, 1972.

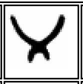
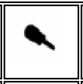
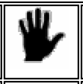
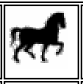
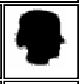
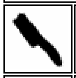
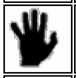
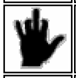
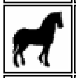
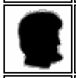
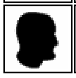
		Class Prototype								
Instance										
	->	245	2772	3436	3078	1274	1950	4840	8189	1631
	->	2515	898	4564	4347	2348	1864	5933	7638	3408
	->	3205	5721	3607	3982	2702	3921	3812	7005	3407
	->	3348	5346	1480	806	4191	5242	3896	6424	3161
	->	4868	5376	1537	1906	3890	5097	5402	7403	4137
	->	5279	5433	1789	526	3902	4790	3945	5921	3389
	->	5627	5429	1777	1545	3783	5801	4844	6095	4043
	->	774	2998	2724	4198	626	1465	4979	8862	2569
	->	1587	2814	2861	4749	652	1685	5633	8951	2336
	->	2779	5075	3737	4893	1567	258	4631	8057	1995
	->	6231	9451	4418	4332	4627	5759	519	4611	4145
	->	5007	8250	4860	4202	4218	4815	1998	5938	3522
	->	8668	12057	6201	5854	6295	7175	3186	2914	5720
	->	2406	4935	5385	3868	1772	2251	5320	6124	798
	->	1855	4296	3205	3222	1045	2528	4761	6054	877

Figure 18: The similarity values computed by our similarity measure on test objects corresponding to Table 1, p. 27, in [21].

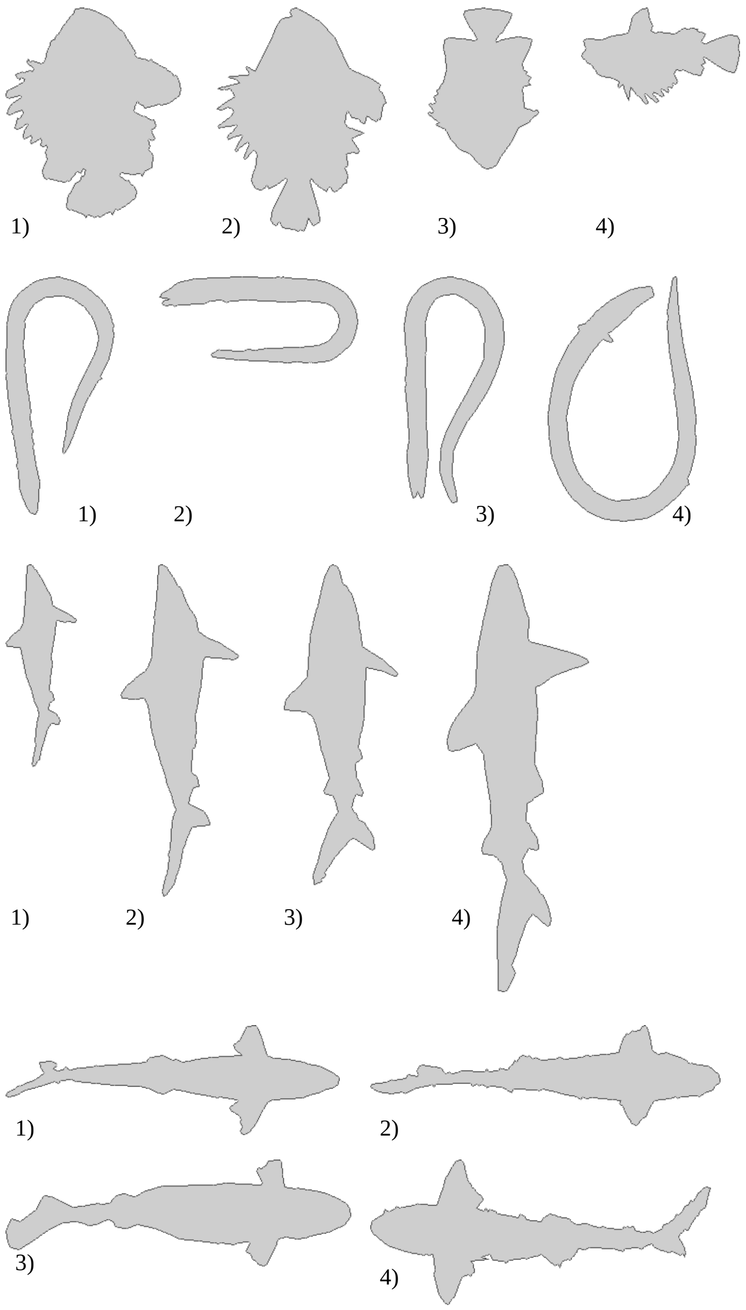


Figure 19: For each query object marked with (1), the most similar objects retrieved by our similarity measure are presented. The retrieval was performed in a database of 1100 marine animals obtained from http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html.