PERGAMON

Vision Research 38 (1998) 2365–2385

Vision Research

Determining the similarity of deformable shapes

Ronen Basri [a], Luiz Costa [b], Davi Geiger [c], David Jacobs [d,*]

[a] *Department of Applied Math, The Weizmann Inst. of Science, Rehovot, 76100, Israel*
[b] *Laboratory of Integrated Systems, University of Sao Paulo, Sao Paulo, Brazil*
[c] *Courant Institute, New York University, New York, NY, USA*
[d] *NEC Research Institute, 4 Independence Way, Princeton, NJ 08540, USA*

Received 9 January 1997; received in revised form 12 January 1998

Abstract

Determining the similarity of two shapes is a significant task in both machine and human vision systems that must recognize or classify objects. The exact properties of human shape similarity judgements are not well understood yet, and this task is particularly difficult in domains where the shapes are not related by rigid transformations. In this paper we identify a number of possibly desirable properties of a shape similarity method, and determine the extent to which these properties can be captured by approaches that compare local properties of the contours of the shapes, through elastic matching. Special attention is devoted to objects that possess articulations, i.e. articulated parts. Elastic matching evaluates the similarity of two shapes as the sum of local deformations needed to change one shape into another. We show that similarities of part structure can be captured by such an approach, without the explicit computation of part structure. This may be of importance, since although parts appear to play a significant role in visual recognition, it is difficult to stably determine part structure. We also show novel results about how one can evaluate smooth and polyhedral shapes with the same method. Finally, we describe shape similarity effects that cannot be handled by current approaches. © 1998 Elsevier Science Ltd. All rights reserved.

1. Introduction

The world perceived by the human eye is rich and diverse. Images observed in the course of time vary significantly and are rarely identical, yet there is a considerable resemblance between different images. Detecting this similarity is key to the interpretation of images. In this paper we consider the problem of judging similarity for the restricted, but still significant and challenging domain of comparing the boundaries of 2D shapes.

People are able to recognize objects even though their boundaries undergo a great many possible distortions due to changes in viewpoint, deformations or articulations in shape, or the variations that occur between different objects of the same class. This suggests that we have a very general and powerful ability to judge the similarity of shapes. While prior knowledge of specific objects may influence overall judgements of similarity, in this paper we focus on the problem of generic shape similarity. We consider methods for determining the similarity of two shapes without reference to prior knowledge of other specific shapes. While it is still an open question as to whether such a general shape similarity capacity exists, we feel that it does for two reasons. First, as we will attempt to show, people form clear similarity judgements for seemingly arbitrary pairs of shapes. Second, even if overall similarity judgements are based upon previously seen shapes, the most similar previous shapes must first be accessed. A general facility must tell us which previous shapes are most relevant to help us compare two new shapes.

Our paper attempts to make two contributions to the understanding of generic shape similarity processes. First, we attempt to clarify which properties may be important in human similarity judgements. We provide simple examples which suggest the role that part structure may play in similarity judgements, that indicate how human perception compares smooth shapes with discontinuous shapes, and that raises questions about the relative importance of boundary-based comparisons and comparisons of regions. This work is only suggestive; we delineate the space of significant issues that seem most relevant to building models of similarity, we

* Corresponding author. E-mail: dwj@research.nj.nec.com.

0042-6989/98/$19.00 © 1998 Elsevier Science Ltd. All rights reserved.
PII: S0042-6989(98)00043-1

do not present psychophysics that will settle these issues.

Second, we examine the extent to which these properties can be captured by measures of similarity that are based on local comparisons of the contours of shapes. To do this, we consider elastic matching, which is one of the most powerful computational approaches to measuring similarity. Elastic matching is popular because it can be computationally efficient and because it can allow for non-rigid deformations in objects under comparison. However, it is typically applied to specific applications. The goal of this paper is to better understand the potential of elastic matching as a model of human vision.

In elastic matching one searches for a correspondence between portions of two contours that minimizes some comparison cost between them. The key issue in developing the cost function is to determine how local differences between the contours should effect their perceived similarity. Other questions involve structuring the set of possible correspondences, and performing an efficient search for the best correspondence.

We particularly focus on the ability of this approach to explain the role that part structure appears to play in human similarity judgements. We show that a method which compares shapes by measuring purely local deformations can still account for some of the effects of part structure. In particular, we show how to model objects with parts that can undergo articulated motion, and to weight shape changes at part boundaries differently from changes in the middle of parts. This allows us to compare shapes in a way that respects part boundaries, without ever explicitly computing part structure, or committing to a single part decomposition of a shape.

We also consider a number of other issues in shape similarity. We show how to derive our similarity method based on a simple model of contours as elastic 1D objects. This allows us to relate the similarity computed by elastic matching to a physical energy measuring the amount of deformation needed to explain the differences in two contours. We also show how one may develop a similarity method that can compare both smooth and polygonal shapes; previous methods based on differential properties do not apply to shapes with discontinuities. We also describe a method of comparing partially occluded shapes. Finally, we discuss the implications of requiring a comparison method to provide a metric space for shapes. In order to make this very difficult problem tractable we have limited the scope of our inquiry. Clearly, there are global properties of a shape, such as symmetry, that play an important role in shape similarity and that should be considered by methods that do not make purely local comparisons. However, knowing that local comparisons of shape will not tell the whole story about similarity, our goal has been to provide better insight into the capabilities of such methods.

Although our focus is on what can be computed rather than on issues of how best to compute it, we have also developed an efficient experimental system. The comparison methods that we consider are implemented using shortest path algorithms. This provides us with an algorithm that is guaranteed to find the correspondence between the two contours that minimizes the energy function under consideration. Therefore we may evaluate the practical performance of our ideas without muddying the waters with computational heuristics that might effect system output. These experiments help to illustrate some of the points that we also make theoretically, and also demonstrate some of the potential utility of our ideas. A more abbreviated version of these results has appeared (see ref. [4]).

In sum, we feel that our work makes several potential contributions to the understanding of human shape similarity judgements. First, we provide examples that suggest some properties of generic similarity judgements; that they seem to respect the part structure of objects, and that they seem to allow comparisons between smooth and polygonal shapes. Second, we discuss how these properties can be encoded in a method that makes local comparisons of contours, without explicitly computing part structure. Finally, we provide examples to indicate some of the limitations of purely contour-based approaches to similarity.

2. Background

In this section we provide a brief overview of the extensive body of work that is relevant to the problem of shape similarity. We focus on reviewing two streams of prior work. First we look at work on matching deformable shapes using energy minimization methods that compare local portions of the contours. Next we consider work that demonstrates the role played by part structure in visual object recognition; our work will attempt to combine these two approaches. We will describe more briefly other approaches to shape similarity, including feature based methods and functional methods. See Mumford [37] for another overview of methods of judging similarity.

2.1. Methods that measure deformations

Elastic matching methods measure the difference between two shapes based on a specific correspondence between points on the shapes. A cost function weights the similarity of matched points on the two curves on the basis of local properties of the points, such as the distance between them, or the difference in the tangent or curvature of the contour at those points. The overall similarity of the contours is then measured as the sum of these local costs. We illustrate this in Fig. 1. Typically the cost function itself is used to find the correspondence between the curves; that is, some search is carried out for the matching between the curves that minimizes this cost. This matching allows for stretching of contours, so that a larger portion of one contour may match a smaller portion of the other. Such methods produce both a measure of similarity and a correspondence between contours; the goal of the algorithm may be either one or both of these. The two main components of an elastic matching method, then, are the its cost function and its search mechanism.

A number of different cost functions have been proposed for elastic matching. For example, Yoshida and Sakoe [56] propose comparing hand- printed characters by measuring the difference in angle and distance between corresponding points along the two contours. Tappert [48] compares the difference between the angle and height of corresponding points for character recognition also. In related work, Burr [11] compares contours based on the distance and angle between the contours. Related approaches are also taken by Tsai and Yu [49], and Mehrota and Grosky [36], the latter focusing on the problem of indexing into a data base of shapes.

Similar approaches have been used in work on motion tracking. In some cases this work also incorporates a cost that measures the amount of 'stretching' of a contour. Hildreth [22] proposed computing motion flow between two contours with a cost function that penalizes for the variation in the magnitude of motion. Cohen, Ayache, and Sulger [12] propose matching contours extracted from moving images of biological structures (such as beating hearts) with a cost function based on the difference in curvature between the two curves. Tagare, O'Shea and Rangarajan [47] also compare the curvature of two contours, taking care to propose a cost function that is symmetric, as we did [4] in our initial work (that is, the distance from contour A to contour B is the same as the distance from B to A). McConnell et al. [35] use a cost function based on the Euclidean distance between contour points to compare SAR images of ice floes that may deform due to melting.

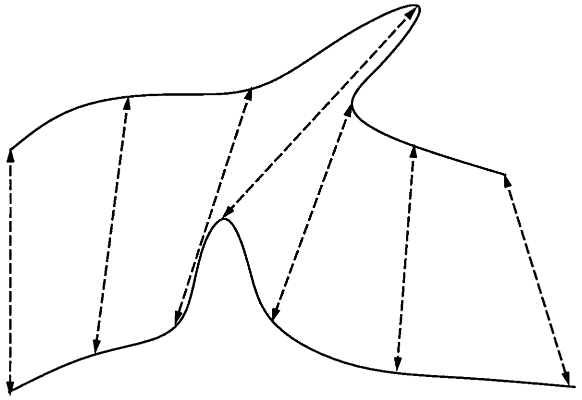


Fig. 1. In elastic matching, a correspondence is hypothesized between points on two contours. Then a cost is computed for each pair of matched points, based on the similarity of their local properties (e.g. tangent and curvature) and properties of the match (e.g. amount of stretching one contour undergoes to match the other). The total cost is the integral of this local cost over the whole correspondence.

Related work compares shapes by measuring the deformations needed to relate the regions bounded by the shapes' contours, rather than the contours themselves (eg. Amit, Grenander and Piccioni [1] and Jain, Zhong, and Lakshmanan [28]). In Section 7 we briefly discuss some of the advantages and disadvantages of comparisons based on regions rather than the contours that bound them.

Another approach is to develop a general model of shape that captures the properties of a class of objects and then to compare a new shape to this generic model. Kass, Witkin, and Terzopoulos [31] match a generic shape model to intensity images. They track contours in images by minimizing a cost function that favors smooth contours that fit the gradients of the image as well as possible. Yuille, Cohen, and Hallinan [57] recognize parts of a face using deformable models that they have hand-crafted, while Cootes et al. [14], Baumberg and Hogg [5], and Hinton, Williams and Revow [23] suggest learning deformable models of objects from examples. Finally, we should note that elastic matching has been widely used outside of vision to compare 1D objects, and especially to compare speech signals. A review of some of this work can be found in Sankoff and Kruskal [43].

Many of these methods simply propose an intuitively appealing cost function, which may be crafted for a particular application domain. However, some papers also attempt to determine general properties of shape matching methods that are based on the cost functions they propose. For example, Hildreth [22] stresses that the proposed cost function leads to unique solutions in motion tracking. Arkin et al. [2] stress that their method leads to a cost function that is invariant when contours undergo similarity transformations, and places contours in a metric space. Tagare, O'Shea and Rangarajan [47] show the symmetry of their measure and its invariance to parameterizations of the contour. Our goal will be to provide a more complete description of the properties that an elastic matching method may have in terms of the possible relevance of these properties to modeling human vision. We will show how these properties depend on the choice of the cost function used in shape comparison.

Given a choice of cost functions, many methods have been used to find the correspondence between contours that minimizes this cost. When the two shapes being compared are ordered strings, dynamic programming can be used to find an optimal correspondence (eg. [35,48,49,56]). Recently, more efficient shortest path algorithms have been used on related problems by Geiger et al. [17], which we use in our implementation. Arkin et al. [2] use a more complex, geometrical algorithm specific to their formulation of the problem. Also popular have been various methods based on gradient descent (eg. [12,31,57]). These methods have the disadvantage that they may converge to a locally optimal match that is not the best overall match. However, they may be preferable when dynamic programming is not possible because the contour points are not ordered in a 1D string, or when a good starting point for the method is available. Elastic matching of 2D images has also been considered. Tsukumo [50] and Tsukumo and Tanaka [51] describe methods of registering images of Kanji by using dynamic programming to register a 1D projection of the 2D image. The cost function is based on features of 1D slices of the 2D image. Levin and Pieraccini [33] also describe a method of warping 2D images by registering 1D slices of them, using dynamic programming. Our focus in this paper, however, is on how similarity is encoded in the cost function rather than on solution methods.

2.2. Parts-based methods

A second influential computational approach to object classification stresses descriptions of shapes in terms of their part structure. Parts generally are defined to be convex or nearly convex shapes separated from the rest of the object at concavity extreme, as in Hoffman and Richards [25], or at inflections, as in Koenderink and van Doorn [30]. Many methods have been suggested for providing geometric descriptions of these parts, such as generalized cylinders [7] and superquadrics [3,39]. Some methods have been proposed for judging whether two images come from the same class of objects by describing object classes using parts that have parameterized descriptions [10,20,21]. It has also been suggested that object parts be described using a small set of qualitative or topological properties. In this approach parts with different shapes may be judged similar when they contain the same qualitative properties. Biederman [6] proposes describing parts in terms of their non-accidental properties. Marr and Nishihara [34] make qualitative comparisons between part structures described as generalized cylinders. Connell and Brady [13] also propose a qualitative, parts-based description of shape. A number of other proposals have also been made for representing objects in terms of their possible decomposition into parts and the relationships between these parts [32,44,58].

Biederman [6] has produced a good deal of evidence for the contention that human object recognition is sensitive to the part structure of objects (see also ref. [45]). However, as is shown in Fig. 2, part structure is insufficient to solely determine the category of objects. Metric properties are essential to determine the class of objects. The methods that use parts to recognize objects lack a theory specifying how the metric properties should be used and how they should interact with the parts. Such a theory should for example determine when objects that share the same part structure should belong to different classes. Other problems with methods that recognize objects by looking at their part structure are that some objects lack distinctive part structure (e.g. a shoe, as shown in Fig. 3), and that methods to extract parts from images tend to be unstable and degrade significantly in the presence of noise and occlusion. One reason for these problems is that there is no clear definition as to what constitutes the stable 'parts' of an object. It is one of the goals of our paper to show that some of these problems can be overcome by integrating parts-based and precise metric judgements into a single framework, which attends to the part structure of objects without requiring the explicit computation of parts.

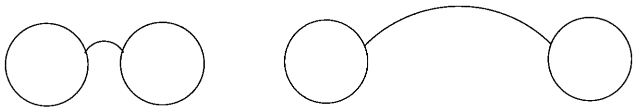


Fig. 2. A pair of spectacles (left) and an object that has the same parts, but does not look like spectacles (right).

2.3. Other work

There has been a great deal of other work on shape similarity and object recognition which we can only briefly touch on here. Some proposals for comparing shape evaluate global, rather than local, deformations. Pentland and Sclaroff [40] construct mass and stiffness matrices for the given shapes, and deform the shapes by aligning the principal modes of their mass and stiffness matrices. Huttenlocher et al. [26,27] compare two shapes by applying the rigid (or affine) transformation that minimizes the Hausdorf measure between them. To allow for occlusion only a certain fixed fraction of the contour points are used. Global approaches tend to be

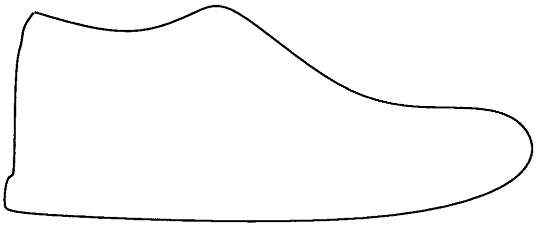


Fig. 3. A shoe. It appears to be difficult to divide this shape into meaningful parts.

sensitive to occlusion. Huttenlocher et al.'s method for handling occlusion allows only for a fixed amount of occlusion. In addition, these methods fail to account for local deformations (such as articulations) since these deformations may change the global appearance of objects considerably while the entire deformation is concentrated in specific points. Intermediates between local and global methods, Ullman [53] proposed comparing shapes after deforming them with piece-wise affine transformations, although this suggestion is not fleshed out.

Another common method of shape comparison is to describe shapes with a list of properties (features). These properties may be global (e.g. 'the object is polygonal') or local. When global properties are used (see reviews refs. [9,16]) objects are represented as points in feature space. This representation is obtained by arranging the list of properties of an object as the components of a vector associated with the object. Similarity between objects is determined by the distance in feature space between the vectors associated with the objects. When local features are used [41] an object is represented by a graph with nodes representing the feature values and edges representing the spatial arrangement of the features. Objects are considered similar if their graphs are isomorphic.

Methods that use features critically depend on the set of features extracted. A small change in the shape of an object may sometimes cause a significant change in its features. It is difficult to find a coherent set of features that can faithfully reflect all possible shapes. In addition, global features tend to be sensitive to occlusion, whereas local features tend to be sensitive to noise and to small variations in shape.

Another class of methods attempts to assign 'semantic interpretation' to shapes. The most common semantic attribute is function. For example, a chair may be defined as an object that has a sittable surface and provides a stable support [46]. Two objects are considered similar if they share the same set of semantic attributes. Methods that use function have attracted researchers from both fields of computer vision and artificial intelligence [24,42,46,55].

The replacement of geometric structure with semantics is potentially powerful. In particular, eccentrically designed objects may be recognized using this method even when they differ visually from conventional designs. Methods that use function, however, suffer from several problems. First, function is difficult to extract, especially under partial occlusion. Second, the use of function is suitable for man-made objects, but is more difficult to extend to natural objects. Third, methods that use function tend to over generalize. For example, a wide flat rock may be recognized as a chair because it is sittable. Finally, it seems useful to be able to judge the purely visual similarity of shapes, regardless of their possible functionality, and humans appear to have this ability.

3. Constraints on a similarity function

We now consider elastic matching as a potential model of human shape similarity judgements. We feel that currently, it is not well understood what properties perceptual similarity has. In fact, it is not clear whether there is a single similarity system, or whether different similarity measures may apply in different circumstances. For example, it is possible that different similarity methods operate between classes than those operating within classes. It is also possible that our prior knowledge of objects plays a significant role in our similarity judgements, a role which may vary considerably depending on the shapes we view. Therefore, our main goal is to describe the possibly desirable properties of perceptual similarity, and then to see how these might be obtained by elastic matching. Below we enumerate the desirable properties (see Section 3.1) and then show that not all these properties can be satisfied simultaneously by elastic matching.

3.1. Desired properties of a similarity function

We define a cost by finding a mapping between two contours, Γ_1 and Γ_2, and then summing the cost of local deformations that reflect the differences between the two contours. Let Γ_1 be parameterized by arclength s, so that $\Gamma_1(s)$ indicates a point on Γ_1. Then a correspondence between two curves is given by a function, $t(s)$, from arclength to arclength, so that the point $\Gamma_1(s)$ is mapped to the point $\Gamma_2[t(s)]$ (we will sometimes write $t(s)$ as just t, when s is clear). Then we consider cost functions that take this correspondence into account, by integrating over the curves a measure of the local difference between the curves, given this correspondence. This similarity must be computed based on some local property of the two contours. As we will see, a method based on local comparisons has the advantage that it can model articulations and local changes of object shape. For example, if a person raises her arm, we want the cost of this change in shape to be concentrated in a comparison of arm angle, while noticing that the rest of the person remains the same.

We will focus on costs that are based on the lowest order differential properties of the curves that can still meet our objectives. In particular, we will consider cost functions of the form:

$$C(\Gamma_1, \Gamma_2) = \min_{t(s)} \int_{\Gamma_1} F\left(k_1, k_2, \frac{dt}{ds}\right) ds.$$

Here, C is shorthand for the overall similarity between the two contours. K_1 is the curvature of Γ_1 at

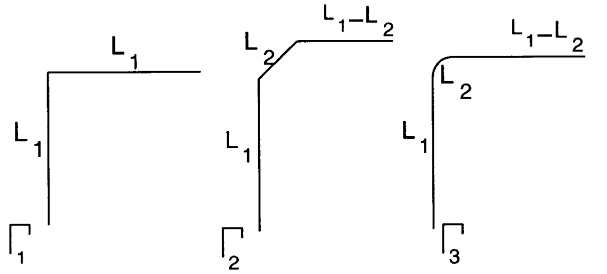


Fig. 4. Our work uses the intuition that as l_2 decreases, the two figures on the right each become more similar to the figure on the left.

point $\Gamma_1(s)$ [i.e. shorthand for $k_1(s)$], and similarly k_2 is the curvature of Γ_2 at point $\Gamma_2(t(s))$. F is some still unspecified function. The minimum is taken over all possible continuous correspondences between the two contours. This says that we judge the cost between two curves based on the local differences of curvature and relative speed of progression along the curves. More intuitively, we can think of this as judging cost based on the minimum amount of bending (change in curvature) and stretching ($\mathrm{d}t/\mathrm{d}s$, or t') required to make the curves identical. Note that curvature is the lowest order local property of curves that can meet our goals. If we based our cost on tangents, for example [48,49], then the articulation of a part would change our local description of the entire part, not just the point at which it bends. A cost function based on the stretching and bending needed to make the contours the same has considerable intuitive appeal as well. One of the potential values of a shape similarity judgement is that it may allow one to identify an object that is stretching and bending in an unknown way. While it is possible to base elastic matching on different local properties of a curve, the class of functions we consider encompass most of those that we have reviewed.

We now enumerate some possible properties of a shape similarity method, and consider the implications of each for the cost function F that will fully specify our similarity method. We will then briefly discuss the desirability of each property.

3.1.1. Continuity

C is continuous. That is, as the curves change smoothly, so does C. This implies that if a sequence of curves, $\Gamma_{2,i}$ converges to Γ_2 then $C(\Gamma_1, \Gamma_{2,i})$ converges to $C(\Gamma_1, \Gamma_2)$. This will be true if and only if F is continuous.

Perceptually, human comparisons between shapes may not be smooth. For example, a deformation that eliminates some Gestalt property such as symmetry or collinearity may have a discontinuous effect on human comparisons of shapes. While it is beyond the scope of this paper to consider the effects of global properties, such as symmetry, on shape similarity, they are clearly quite significant.

Fig. 4 shows two examples which demonstrate the requirements for continuity. We let Γ_1 be a curve consisting of two straight lines each of length L_1, connected by a corner with an exterior angle of $\pi/2$. Let Γ_2 have three straight lines of length L_1, L_2, and $L_1 - L_2$. Further, let the two corners connecting the intermediate line of Γ_2 have exterior angles of $\pi/4$. Γ_3 is like Γ_2, except that the two straight lines are connected by a circular arc of length L_2, and curvature $\pi/2L_2$, so that the angle between the two lines is also $\pi/2$. So, the three curves have the same length, and they all turn by a total angle of $\pi/2$. As $L_2 \to 0$, the two curves on the right become more and more similar to the curve on the left. In the remainder of the paper, we will rely on the intuition that as $L_2 \to 0$, the cost of comparing Γ_1 with Γ_3 should also go to zero. Some assumption such as this is necessary to define a cost for corners, whose local derivatives are not defined. In the case of Γ_2, we only rely on the intuition that the cost of comparing the two curves should be greater for some large value of L_2 than for some small value of L_2, even if we are not prepared to specify the exact form of this change in cost, or its asymptotic behavior as $L_2 \to 0$.

3.1.2. Metric

C is a metric. This implies:

1. $F(k_1, k_2, t') > 0$.
2. $F(k_1, k_1, 1) = 0$; $F(k_1, k_2, t') > 0$ for $k_1 \neq k_2$ or $t' \neq 1$.
3. $F(k_1, k_2, \mathrm{d}t/\mathrm{d}s)\mathrm{d}s = F(k_2, k_1, \mathrm{d}s/\mathrm{d}t)\mathrm{d}t$
4.

$$F(k_1, k_2, \mathrm{d}t/\mathrm{d}s)\mathrm{d}s + F(k_2, k_3, \mathrm{d}u/\mathrm{d}t)\mathrm{d}t$$

$\geq F(k_1, k_3, \mathrm{d}u/\mathrm{d}s)\mathrm{d}s, \quad \forall k_1, k_2, k_3,$where k_3 indicates the curvature of a corresponding point on Γ_3, which is parameterized by u.

It will be very useful for C to be a metric if we intend to use C in a machine recognition system. For example, if C is a metric, and we compare Γ_1 to Γ_2, and find them very dissimilar, and compare Γ_2 to Γ_3 and find them similar, we do not need to explicitly compare Γ_1 to Γ_3; we know these two curves will be dissimilar. For this reason, virtually all efficient methods of finding nearest neighbours rely on the metric properties of a comparison function.

On the other hand, it is not clear that human comparisons of shape obey metric properties. In particular, the triangle inequality may not hold for human comparisons of shape. For example, it is quite possible that people perceive a horse and a man to have very different appearances, while a centaur may be quite similar to both. Also, the symmetry constraint may not be obeyed [37,52]. For example, an ellipse may be perceived as more similar to a circle (it may be the image of a slanted circle) than a circle is to an ellipse (it is very uncommon for a circle to be the image of an ellipse).

3.1.3. Invariance

We may wish our cost to have some invariant properties under some classes of transformations. For example, we might wish the cost to be zero when comparing two shapes related by a Euclidean, similarity or affine transformation, or we might wish the relative nearness of curves to be fixed under some class of transformations.

By taking F as a function of curvature, we guarantee that it is invariant under Euclidean transformations. Should we wish to make it scale invariant, this would imply: $F(k_1, k_2, t') = F(k_1, k_2/\alpha, \alpha t')$. A weaker version of this condition is that the cost of scaling is independent of the curvature, i.e. that we divide F into two costs, one of which is scale-independent and the other of which only measures the cost of scaling. This would mean that: $F(k_1, k_2, t') = f(k_1, k_2, t') + g(t')$ where $f(k_1, k_2, t') = f(k_1, k_2/\alpha, \alpha t')$. This is easily achieved by making f a function only of k_1 and k_2t'. k_2t' is scale independent since k_2 and t' scale precisely inversely as Γ_2 is scaled.

In cases in which our cost function is not scale invariant, we might prefer that scaling three curves does not change their relative similarity. This implies that:

$$F(k_1, k_2, t') < F(k_1, k_3, u') \Rightarrow F(\alpha k_1, \alpha k_2, t') < F(\alpha k_1, \alpha k_3, u')$$

where α is any scale factor.

3.1.4. Handling polygons

C should not diverge for discontinuous curves. That is, any two finite length curves should be related by a finite cost. In particular, this means that C provides a meaningful comparison between polygons, or between a polygon and a smooth curve. The consequences of this property are complex, and we postpone discussion of them to the next section.

That the cost function handles both smooth and polygonal shapes, also seems quite important. In a practical system we will be comparing discretized versions of continuous or discrete contours; it therefore seems essential that a close polygonal approximation to a shape be considered quite similar to that shape, and not have a divergent cost. Humans also seem to sensibly compare polygonal and curved shapes; for example a regular octagon may appear more similar to a circle than does a regular triangle. This condition is a prerequisite to making such judgements of ideal, non-discretized contours.

3.1.5. Handling parts

We may want the cost function to reflect the part structure of objects. This has two main implications. First, objects often articulate at part boundaries. So if a part of an object rotates, the cost function should capture the fact that the relationship between the part and the rest of the object has changed while the internal shape of the part has not. Second, articulations of a shape at the likely part boundaries should cost less than bending that occurs in the middle of a part boundary.

The first condition is captured by our use of a local cost function that compares the curvature of contours. As a part articulates, the curvatures of the contour only change at the points where the part joins the body. The second condition is captured if deforming a shape by bending it will cost less if the bending is done at points of high absolute curvature, given that part boundaries typically occur at points of high curvature.

This implies:

$$F(k_1, k_2, 1) > F(k_1 + a, k_2 + a, 1),$$
$$\forall a, k_1, k_2 > 0, \forall a, k_1, k_2 < 0.$$

It is less clear what this constraint should apply when $t' \neq 1$. For example, if our function were scale invariant, this would imply that:

$$F(2, 1, 2) = 0 < F(3, 2, 2)$$

We make no assumptions for the case where one curvature is positive and one curvature is negative, since this seems less clear intuitively.

This condition reflects the perceptual phenomenon that contours appear more 'bendable' at places where there is already a high curvature than at points of low curvature. This is illustrated in Fig. 5. This reflects the part nature of objects; part boundaries tend to occur at points of high absolute curvature. Articulated or deformable objects tend to bend more easily at part boundaries.

3.1.6. Monotonicity

If one deforms a shape by bending it or stretching it, the cost between the deformed and original shape should grow as the bending or stretching increases.

This means that: $k_1 < k_2 < k_3 \Rightarrow F(k_1, k_2, 1) < F(k_1, k_3, 1)$, and $1 < t'_1 < t'_2, k_1 = k_2 \Rightarrow F(k_1, k_2, t'_1) < F(k_1, k_2, t'_2)$. Again, it is less clear how the cost function should behave if bending increases in the presence of stretching, or if stretching occurs when the bending is unequal. This condition reflects the intuition that in-

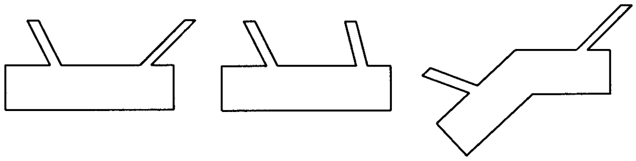


Fig. 5. The middle figure shows the figure on the left, bent at points of high curvature. On the right, the figure is bent by equal amounts at points of low curvature. By penalizing the bending on the right more than the bending in the middle, we produce comparisons sensitive to the part structure of objects.

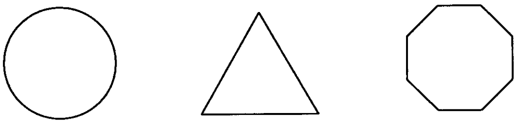


Fig. 6. A circle, regular triangle, and regular octagon, with the same perimeter. We may require a cost function to judge the octagon and circle more similar than the triangle and circle.

creasing deformations of a shape should make them less and less similar to the original shape.

3.1.7. Small versus large deformations

Two shapes that are related by many small deformations should be considered more similar than two shapes that are related by a single deformation of equal total magnitude. This constraint implies that: $nF(k_1, k_1+\epsilon, 1) < F(k_1, k_1+n\epsilon, 1)$,

$\forall 0 < k_1, 0 < \epsilon, 1 < n$.

It is generally desirable when dealing with noisy data to count a single change of a given magnitude more heavily than many small changes whose magnitude sum to the same amount. With Gaussian noise, for example, it is optimal to weight changes by the square of their magnitude. This principle may also be present psychophysically, as when comparing a circle to a regular triangle, and to a regular octagon (see Fig. 6). It is our untested hypothesis that the octagon is typically viewed as more similar to the circle. One possible reason for this is because, while the octagon has more corners to be smoothed out, these corners need be smoothed less to match them to the circle. Both the triangle and octagon, of course, have changes in angle totalling 2π. We will discuss this example in more detail later, with regard to a specific type of cost function. Overall, we feel that this and the other properties we have described might be desirable in a model of human similarity judgements, although their necessity is not yet proven.

3.2. The constraints cannot all be satisfied

We now discuss two difficulties that arise in trying to satisfy these constraints with an elastic matching method. First, we show that properly handling corners presents problems for a cost function based on purely local comparisons. Of course, curvature is not defined at the corners of a polygon. We can overcome this problem by considering a polygon as the limit of a series of smooth curves that converge to the polygon. However, other problems still remain. Consider the example comparing Γ_1 and Γ_2 (Fig. 4), as described in Section 3.1.1. For any $l_2 > 0$, a cost function based on a one-to-one correspondence between the curves will match the corner of Γ_1 to one of the corners of Γ_2, while matching Γ_2's second corner to a point on one of the straight lines in Γ_1. The remaining points are locally identical, and may be matched perfectly. Therefore, the cost between the two curves will not change as l_2 changes, contrary to our assumptions about this example.

To overcome this problem, it seems intuitively that we must match a finite portion of Γ_2 or Γ_3 to a point or small portion of Γ_1. This is most easily done by assuming that the curves are first smoothed by some fixed amount. A purely local comparison after smoothing is equivalent to a comparison with a small base of support. Notice that if we smooth the curves by some fixed amount then as $l_2 \to 0$, the smoothed corners of Γ_2 will become identical to the smoothed corner of Γ_1 and the two curves will become identical. Note that our reasoning applies to an abstract, continuous formulation of the cost function. In practice, any implementation must be discrete, which implies discrete comparisons and some implicit smoothing. However, we see that we must rely on these sorts of effects to handle corners.

We now show that problems arise even when our comparison is made after a small amount of smoothing. First, we use the comparison between Γ_1 and Γ_3 in Fig. 4 to define the cost function at discontinuities. We define a corner of angle θ as two straight lines connected at a point of infinite curvature, such that the integral of the curve's curvature is θ. As in our example, we assume that a corner of angle θ can be thought of as the limit of a sequence consisting of two lines joined by a circular arc of length l and curvature θ/l, as the length l goes to zero. Ignoring the effects of smoothing, and supposing that the cost of comparing two such sequences, which converge to corners, should be similar to the cost of comparing two corners, we find:

$$\lim_{l\to 0} lF\left(\frac{\theta_1}{l}, \frac{\theta_2}{l}, 1\right) \to c(\theta_1, \theta_2)$$

where $c(\theta_1, \theta_2)$ is a finite-valued function, that increases monotonically as θ_2 increases, for a fixed θ_1 and $\theta_1 < \theta 2$

In order for the corner cost to converge to a non-infinite value, then, it must be the case asymptotically that:

$$\lim_{k_2\to\infty} F(k_1, k_2, 1) = aF\left(k_1, \frac{k_2}{a}, 1\right)$$

for any constant a. That is, asymptotically, F must be linear in k_2. Hence, the cost of transforming a point of finite curvature into a corner will depend linearly on the angle of the corner. This creates a problem for condition 3.1.7. In particular, this implies that when comparing a polygon to a circle, the cost will depend only on the sum of the angles of the polygon, as these angles are deformed to match portions of the circle that have finite curvature. Since the exterior angles of any convex polygon sum to 2π, this means that all convex polygons will be judged equally similar to a circle. Again, we may rely on smoothing to overcome this problem. A smoothed regular octagon will have curvatures much more similar to a circle than will a smoothed regular triangle.

Although we must clearly rely on smoothing to produce desirable results in the case of corners, this discussion clarifies our choices about the cost function's behavior at corners. For when smoothing is small relative to the scale of a corner, the behavior of the cost function will approximate its behavior on unsmoothed corners. So we may choose a cost function that is asymptotically linear in curvature. This will tend to keep the cost function from growing very large in the presence of corners, but will also tend to produce a linear cost when comparing corners, which can be undesirable, as for example, in the case of comparing polygons and circles. Alternately, we can choose a cost that is asymptotically non-linear for corners, which will produce a cost for comparing corners that goes either to zero or infinity as the amount of smoothing is small. See Weiss [54] for a related discussion on an approach to handling corners in functions describing the energy of a curve.

We now demonstrate that another contradiction arises when we try to simultaneously satisfy all desired constraints on the function F. We then consider various possible tradeoffs in partially satisfying these constraints.

Let us consider what relative values F should have for $F(0, 1, 1)$, $F(0, 2, 1)$, $F(1, 2, 1)$. Constraint 3.1.6 tell us that we should have:

$$F(0, 1, 1) > F(1, 2, 1)$$

following the intuition that it becomes easier to bend a shape at higher curvature points. The metric constraint (Condition 3.1.2) tell us:

$$F(0, 2, 1) \leq F(0, 1, 1) + F(1, 2, 1)$$

Finally, Condition 3.1.7 tells us that:

$$2F(0, 1, 1) \leq F(0, 2, 1)$$

From the above equations, it follows that:

$$\begin{aligned} F(0, 2, 1) &\leq F(0, 1, 1) + F(1, 2, 1) < 2F(0, 1, 1) \\ &\leq F(0, 2, 1) \end{aligned} \qquad (1)$$

a contradiction.

There are several possible responses to this situation. First, if we replace the constraint:

$$F(k_1, k_2, t') > F(k_1 + a, k_2 + a, t'),$$
$$\forall a, k_1, k_2 > 0, \forall a, k_1, k_2 < 0.$$

with:

$$F(k_1, k_2, t') \geq F(k_1 + a, k_2 + a, t'),$$
$$\forall a, k_1, k_2 > 0, \forall a, k_1, k_2 < 0.$$

then all the constraints may be satisfied at equality. However, while a metric may be useful when the triangle constraint is satisfied by equality, the other two constraints are less useful at equality. That is, it is not too satisfying to allow all curve points to be equally easy to bend, or to allow many small changes to have an equal effect to one big change. A second possibility is to consider functions for F that are not metrics because they do not satisfy the triangle inequality. As we have pointed out, this may be reasonable because human perceptions of shape similarity may be non-metric, and because we may be able to design a total cost function that is a metric based on a function F that is not a metric. Third, we can rely on smoothing to satisfy constraint 3.1.7 by smoothing out many small changes more than one big change.

4. A model of cost

We have considered a few potential cost functions since, as is noted above, it is not possible to find an F with all desirable properties. These functions offer a range of possible tradeoffs. Below we introduce a cost function that is derived from a simple model of the formation of the contours (shapes) based on concatenations of springs. We show that this function satisfies all the properties except that it violates the triangle inequality. However, we are able to show that by taking the q'th root of the total cost function we obtain, for particular values of q we are able to obtain a metric. This would imply that the cost function is no longer purely the integral of a local cost, but is a function of this integral. In Appendix B we give another example of a function which relies on smoothing to satisfy the constraint that many small deformations should cost less than one large deformation. This second function is intuitively appealing because it models a deformation as the sum of a series of small deformations. We show below, however, that this function has problems with handling polygons properly.

In our model a deformation of shapes is decomposed into bending and stretching. The energy used for stretching contours is the energy of the concatenated stretched springs. The energy used for bending is also associated with a spring-like model. First we analyze the discrete case, in which the contours are made of points, and then we take the limit where the size of the springs become infinitesimal to consider continuous models.

4.1. Stretching

The energy associated with stretching a spring with stiffness α is given by Hooke's law, $E = \frac{1}{2}\alpha(\Delta x)^2$, where Δx is the displacement from equilibrium. In our model we generalize Hooke's law. We assume that the energy gets increased by a factor $(\Delta x)^p$ with displacements Δx, where $p > 1$, i.e. we assume

$$E = \frac{1}{p}\alpha(\Delta x)^p.$$

If two springs of stiffness α_1 and α_2 are connected together then the resulting spring has stiffness

$$\alpha_{\text{eq}} = \frac{\alpha_1\alpha_2}{\alpha_1^{1/(p-1)} + \alpha_2^{1/(p-1)}} \tag{2}$$

This can be shown by requiring that when a displacement Δx is applied to the resulting spring then, the corresponding displacements, Δx_1 and Δx_2 on each spring must satisfy $F_1 = F_2$, where $F = \alpha(\Delta x)^{(p-1)}$ is the force exerted by each spring. Thus, the three equations that yield Eq. (2) are

$$\frac{1}{p}\alpha_{\text{eq}}(\Delta x)^p = \frac{1}{p}\alpha_1(\Delta x)_1^p + \frac{1}{p}\alpha_2(\Delta x)_2^p$$

energy conservation,

$\Delta x = \Delta x_1 + \Delta x_2$ length conservation,

$\alpha_1(\Delta x)_1^{p-1} = \alpha_2(\Delta x)_2^{p-1}$ force equilibrium.

The main assumption we make is that the stiffness of each spring element is inversely proportional to its length to the power $p-1$, i.e. we assume that an infinitesimal element $\mathrm{d}l$ will have stiffness

$$\alpha_{\mathrm{d}l} = \frac{\alpha}{\mathrm{d}l^{p-1}},$$

which is very large. It is easy to check that two connected elements, $\mathrm{d}l_1$ and $\mathrm{d}l_2$, will yield an equivalent stiffness

$$\alpha_{\text{eq}} = \frac{\alpha_{\mathrm{d}l_1}\alpha_{\mathrm{d}l_2}}{\alpha_{\mathrm{d}l_1}^{1/p-1} + \alpha_{\mathrm{d}l_2}^{1/p-1}} = \frac{\alpha}{(\mathrm{d}l_1 + \mathrm{d}l_2)^{p-1}},$$

and the total contour will have an equivalent stiffness $\alpha_{\text{contour}} = \alpha/L^{p-1}$ where L is the length of the contour.

4.2. Matching contours

In order to compare two contours, Γ_s and Γ_t (see Fig. 7), via the stretched energy, we assume that both are stretched versions of a 'true' relaxed contour. This assumption is necessary to guarantee that the cost to deform contour Γ_s into Γ_t is the same as deforming Γ_t into Γ_s.

Let us assume that the rest (relaxed) length of the 'true' contour element is dependent upon the two contours. So we can choose for instance the infinitesimal 'true' relaxed length to be $(\mathrm{d}t + \mathrm{d}s)/2$ (when matching element $\mathrm{d}s$, from Γ_s into element $\mathrm{d}t$, from Γ_t). The stretching energy becomes the sum of stretching the relaxed element $(\mathrm{d}s + \mathrm{d}t)/2$ once into $\mathrm{d}s$ and once into $\mathrm{d}t$. More precisely,

$$\mathrm{d}E_{\text{stretching}}(\mathrm{d}s, \mathrm{d}t) = \frac{1}{p}\frac{2^{p-1}}{(\mathrm{d}s + \mathrm{d}t)^{p-1}}|\mathrm{d}t - (\mathrm{d}s + \mathrm{d}t)/2|^p + \frac{1}{p}\frac{2^{p-1}\alpha}{(\mathrm{d}s + \mathrm{d}t)^{p-1}}|\mathrm{d}s - (\mathrm{d}s + \mathrm{d}t)/2|^p = \frac{\alpha}{p}\frac{(t'-1)^p}{p(t'+1)^{p-1}}\mathrm{d}s,$$

where $t' = \mathrm{d}t/\mathrm{d}s$. Thus, the total energy of stretching Γ_s into Γ_t is

$$E_{\text{stretching}}(\Gamma_s, \Gamma_t, \mathrm{d}t, \mathrm{d}s) = \frac{\alpha}{p}\int_{\Gamma_s}\frac{|\mathrm{d}t/\mathrm{d}s - 1|^p}{(\mathrm{d}t/\mathrm{d}s + 1)^{p-1}}\mathrm{d}s.$$

4.3. Bending

For bending we assume the same spring model as above, but instead of springs of length $\mathrm{d}s$ or $\mathrm{d}t$ we have springs associated with the angles. More precisely $\mathrm{d}\theta_s = k_s\,\mathrm{d}s$ and $\mathrm{d}\theta_t = k_t\,\mathrm{d}t$ are the infinitesimal elements that are stretched (bended). Thus, to bend spring elements $\mathrm{d}\theta_s$ into $\mathrm{d}\theta_t$ it costs

$$\mathrm{d}E_{\text{bending}}(k_s, k_t, \mathrm{d}s, \mathrm{d}t) = \frac{\alpha}{p}\frac{|k_t\mathrm{d}t - k_s\mathrm{d}s|^p}{(|k_s\mathrm{d}s| + |k_t\mathrm{d}t|)^{p-1}}\mathrm{d}s = \frac{\alpha}{p}\frac{|k_t\mathrm{d}t' - k_s\mathrm{d}s|^p}{(|k_t t'| + |k_t\mathrm{d}t|)^{p-1}}$$

and the total energy is

$$E_{\text{bending}}(\Gamma_s, \Gamma_t, \mathrm{d}t/\mathrm{d}s) = \int_{\Gamma_s}\frac{\alpha}{p}\frac{|k_t t' - k_s|^p}{(|k_t t'| + |k_s|)^{p-1}}\mathrm{d}s$$

Thus, this model has total energy

$$E(\Gamma_s, \Gamma_t, \mathrm{d}t/\mathrm{d}s) = \frac{\alpha}{p}\int_{\Gamma_s}\left[\frac{|k_t t' - k_s|}{(|k_t t'| + |k_s|)^{p-1}} + \lambda\frac{|\mathrm{d}t/\mathrm{d}s - 1^p|}{(\mathrm{d}t/\mathrm{d}s + 1)^{p-1}}\right]\mathrm{d}s.$$

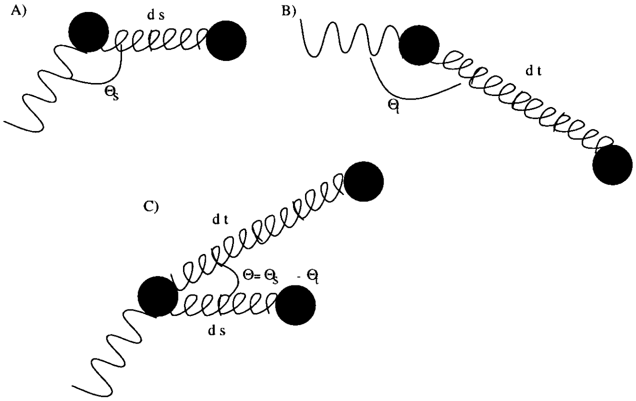


Fig. 7. (A.) and (B.) Two elements of contour 1 (2) forming an angle θ_s (θ_t). The contour is parametrized by arc length s (t). (C.) The two contours are superimposed after being aligned according to their first element. The cost of bringing them together is then defined as the 'physical' effort of bringing the element $\mathrm{d}t$ into $\mathrm{d}s$, which requires bending and stretching.

where λ weights the relative contributions of stretching and bending.

In Appendix A we demonstrate that this cost function meets the previously discussed constraints. In addition to all the other constraints, we show that by taking the appropriate qth root of the total cost over the entire contour, the bending portion of this cost preserves the triangle inequality.

Before we end this section we would like to mention several related approaches to shape description and comparison. An alternative cost function for shape comparison that explicitly allows for articulations is a function that identifies the position of extreme bending and assigns a constant cost to these bendings. A cost function that achieves that, in the spirit of Geman and Geman [18], Mumford and Shah [38], and Blake and Zisserman [8], can be written (in a discretized form) as $E_{\text{bending}} = \Sigma_{s=1}^{S}(k_t t' - k_s)^2(1 - l_s) + \gamma l_s$, where $l_s = 0, 1$. $l_s = 1$ occurs if the difference in matched curvature is high compared to γ, a parameter to be estimated. One problem with this approach is in comparing polygons to smooth curves. For example, with this cost function a triangle would be more similar to a circle than is a square (or any high order polygon), since the similarity between the polygonal straight lines and the circles is the same in both cases, but there are four corners in a square (we activate $l_s = 1$ four times) and only three corners in the triangle. That is, condition 3.1.7 would be violated. This is clearly not desirable and our approach overcomes this problem.

Finally, many methods of curve description have been proposed that are based on curve evolution. For example, recently Kimia et al. [29,44] have proposed curve evolution methods based on reaction-diffusion equations, using the results of this evolution to describe the shape of an object. It is natural to consider a diffusion process to model the similarity between two shapes as well. Kimia et al. propose simulating the reaction-diffusion equations on two shapes and comparing them based on the similarity of their curve evolutions.

In the spirit of our work, one might also imagine comparing shapes by measuring the probability that one shape, undergoing a diffusion process, could evolve into the second shape. However, diffusion processes are not reversible and information is lost. Thus, it is not always possible to obtain one shape by evolving from one into any other shape. Moreover, it is not symmetric. These limitations of the diffusion equations are overcome by our method.

4.4. Linear cost function

A special case of this model is obtained when $p = 1$. As we have noted above, it is possible to satisfy both constraints 3.1.6 and 3.1.7 at equality, while still producing a metric cost. Specifically, we may let:

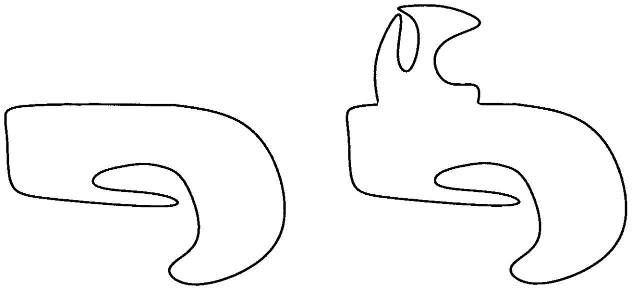


Fig. 8. We allow for the possibility of ignoring the differences between the two figures as due to an occlusion, and focusing instead on the similarities.

$$F(k_1, k_2, t') = |k_2 t' - k_1| + \lambda |t' - 1|$$

As we have already discussed, this function satisfies all other constraints. Furthermore, as with the spring model, the bending cost is scale invariant.

We may also interpret this cost function physically. For $t' = 1$ we may take this as the cost of bending a stiff material, such as wire, from one shape to another, where the cost of bending is uniform throughout the wire (does not depend on curvature). The cost also reflects a uniform, linear cost for stretching.

5. Occlusions

As Fig. 8 illustrates, when the bulk of two figures are similar, but a portion of the figures are quite different, our interpretation may be based on ignoring the dissimilar portions, possibly viewing these as due to an extraneous, occluding object. We therefore allow for the possibility that portions of the two contours remain unmatched, under the assumption that the true corresponding portions of the contour are not visible. We now consider how a proposed cost function should deal with an hypothesized occlusion. (see ref. [19] for another approach to occlusions in a dynamic programming framework).

The cost function for occlusions should reflect three factors. First, it is important to take account of the similarity between the relative positions of the beginning and ending of the hypothesized occlusion in each contour. Fig. 9, shows a rectangle with gaps that has been stretched in two different ways. The gaps signify the location of an hypothesized occlusion. We can see

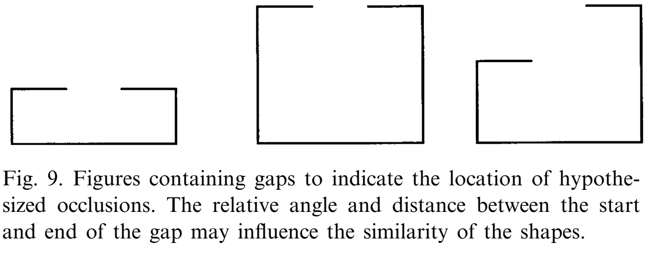


Fig. 9. Figures containing gaps to indicate the location of hypothesized occlusions. The relative angle and distance between the start and end of the gap may influence the similarity of the shapes.

that the relative position of the gap endpoints influences the cost of matching that would occur for any possible missing contour. By 'filling in' these gaps with straight lines, and matching these lines with our previously developed cost functions, we account for the stretching and bending that must occur along any contour fragment that connects the gaps.

Second, the cost function must take account of the length of contour which we do not match. One way to do this is to measure the difference between two contours less the difference that one expects for poorly matching contours, over the contour lengths that have been matched. This evaluates a match relative to one's expectations about how well similar shapes should match, and ensures that we will hypothesize an occlusion only when the contours cannot be matched well. This approach is equivalent to just adding a penalty for occlusions proportional to the length of the occluded contours.

Third, the cost function should reflect the fact that occlusions are relatively unlikely occurrences. All other factors being equal, one much prefers to match all of the contour rather than having to resort to an hypothesized occlusion. This cost should therefore reflect the expected likelihood of occlusions occurring, and may also reflect the likelihood of occlusions of various sizes occurring (i.e. given a probability distribution on the lengths of occlusions, we may determine the likelihood of any particular hypothesized occlusion).

Finally, we note that when gaps occur in contours, these may be treated as known occlusions. For example, if the beginning of a contour does not meet the ending, the gap between the two can be thought of as a contour fragment that is known to be missing. In this case, only the first of the above factors is applied. That is, we insist that the beginnings and endings of the contours match, but also consider the bending and stretching cost of matching the gaps between the ends and the beginnings of the contours. In sum, because our measure of shape similarity is based on local comparisons, it is easy to adapt it to the possible presence of occlusion.

6. Algorithm and experiments

We now describe an algorithm that implements the cost functions that we have discussed, along with experiments. The input to our cost functions has been the curvature at each point. To build a computational system, we need to be able to reliably compute the curvature at every pixel. The curvature for each point i is computed as the difference between the tangent at that point and the tangent at point $i-1$, as shown in Fig. 10.

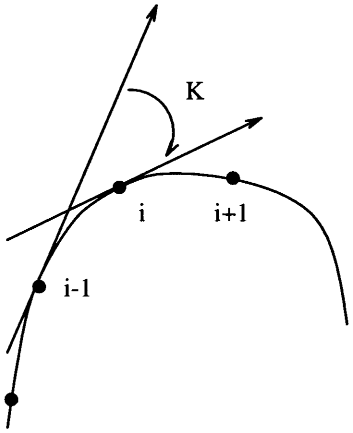


Fig. 10. We compute curvature by taking the difference of adjacent tangents.

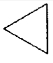
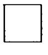
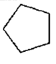
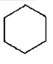
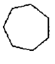
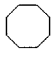
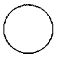
We compute the tangent at i as a weighted average of the unit vectors defined by the consecutive points surrounding i say, $[i-j, i-j+1, \ldots, i, i+1, \ldots, i+j]$. In any discretization of the curvature there will be a smoothing effect dependent on the scale j. This is a parameter in the program. The weight each vector has, w_n, inversely proportional to the distance from the closest point of the vector to the point i.

Two contours are collected and listed in two chains $S = \{x(s); \quad s = 0, 1, \ldots, |S|\}$ and $T = \{y(t); \quad t = 0, 1, \ldots, |T|\}$ where $x(s)$ are the coordinates of the first contour, parameterized by s, and $y(t)$ are the coordinates of the other contour, parametrized by t. Each one has different sizes, denoted by $|S|$ and $|T|$.

The goal is to find the match that minimizes the cost function. A standard way to achieve this involves using dynamic programming. When considering a match $[x(s), y(t)]$, we can consider the total cost to reach all possible predecessors $[x(p_s), y(p_t)]$ and the cost to go from each predecessor to the current hypothesized match $[x(s), y(t)]$. The set of possible predecessors is restricted by only considering the ones that are either one unit before $t, t-1$, or one unit before $s, s-1$, allowing the other coordinate to vary by more than one, i.e. we allow stretches (jumps) on either coordinate (contour) but not on both simultaneously.

Rather than using dynamic programming directly, however, we can instead formulate our problem as a shortest path problem. We let each match, $[x(s), y(t)]$ be a node in a graph, which is linked with a weighted edge to a predecessor state. Then, finding the lowest cost matching becomes equivalent to finding the shortest path from a start to an end state in this graph. While shortest path algorithms rely on dynamic programming, they may in some cases be more efficient than the dynamic programming method we have described, and we find this to be true on our problem (see ref. [15] for more on this difference).

Table 1
Experiments with regular polygons

								
	eq (3)	real data	10.52	9.16	7.78	7.21	6.61	5.41
		synthetic data	11.71	11.47	11.23	11.01	10.80	10.60
	eq (4)	real data	11.74	9.93	9.47	8.66	8.31	6.63
		synthetic data	12.20	12.10	11.99	11.89	11.78	11.68
	eq (5)	real data	15.35	17.30	13.55	14.37	12.73	14.30
		synthetic data	16.26	17.13	17.80	18.33	18.77	19.14

6.1. Stretch cost

To complete the algorithm we need to consider the cost to go from a predecessor to the current hypothesis match, the stretch cost. Let us consider the match $[x(s), y(t)]$ and suppose the predecessor to be $[x(p_s), y(p_t)] = [x(s - \text{Stretch}), y(t-1)]$. Then, we interpolate the curvature between $t-1$ and t by simply repeating (copying)the curvature at $t-1$, i.e. we extend k_{t-1} to the new coordinate t. Thus, the stretch cost becomes:

$$\text{Stretchcost}[(s - \text{Stretch}, t-1);\ (s, t)]$$

$$= \sum_{l=1}^{\text{Stretch}} F(k_{s-l}, k_{t-1}, 1/\text{Stretch}),$$

where $F(k_{s-1}, k_{t-1}, 1/\text{Stretch})$ is one of the cost functions discussed in the previous sections.

To simplify the experiments we have constrained the first points of each contour to match each other. This is not necessary, for closed contours we can consider all possible initial matches, but this does speed up the algorithm. For contours with a length of about 100 pixels, the algorithm's run time is on the order of a few seconds.

6.2. Occlusions

When occlusions are considered we can relax the previous restriction to consider predecessors of the form $(s - \text{Stretch}_s, t - \text{Stretch}_t)$. In this case the cost must be computed as the cost of an occlusion. The algorithm structure, however, remains the same.

We have experimented with a number of different shapes to help us understand the advantages and disadvantages of our cost function. In the experiments, curvature was computed by fitting tangent lines to seven points along the contour. The cost function had the form:

$$\frac{|k_2 t' - k_1|^2}{|k_2 t'| + |k_1|} + \lambda \frac{(t'-1)^2}{t'+1} \qquad \text{Spring model} \qquad (3)$$

We begin with two simple experiments which were designed to demonstrate basic properties of this cost function and to compare it with two other functions, the linear cost function (Section 4) and a function which is based on a continuous deformation model (see Appendix B). These two functions were defined as follows.

$$|k_2 t' - k_1| + \lambda |t' - 1| \qquad \text{Linear model} \qquad (4)$$

$$(t'+1)|e^{-\mu|k_1|} - e^{-\mu|k_2|}| + \lambda |t' - 1|$$

Continuous deformation model (5)

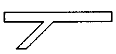
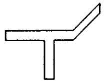
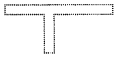
To standardize λ between the different cost functions, we considered the performance of each function on two 'L' shapes with different length sides. We set λ so that each method would produce identical least-cost correspondences for all such 'L's. In this way, although λ is a free parameter, it is set to a comparable value for each function. In function 5 we set $p = 1$.

First, Table 1 shows the cost of comparing a circle to various regular polygons. 'Synthetic' indicates that we used ideal, unsmoothed curvatures, while 'real' shows the results with curvatures computed as described above. As we expect, smoothing allows the spring and linear functions to prefer the polygons with more sides. Even without smoothing there is a slight preference in this direction due to discretization effects. Also as expected, function 5 actually prefers the polygons with fewer sides. This demonstrates the significance of constraint 3.1.7, which this function does not satisfy.

Table 2 shows a simple experiment in which we compare a shape after bending it either at a part boundary, or at points of zero curvature. This illustrates the fact that the linear cost function does not give preference to either sort of bending, while the other two costs prefer bending at part boundaries.

The next figures show the performance of our function (Eq. (3)) on several examples of hand-drawn numerals and on table-like shapes. Fig. 6 shows nine

Table 2
An experiment comparing bending at a part boundary, with bending in the middle of a part

				
	eq (3)	real data	1.68	2.31
		synthetic data	0.84	1.99
	eq (4)	real data	3.78	3.41
		synthetic data	3.14	3.14
	eq (5)	real data	6.03	6.02
		synthetic data	1.44	2.61

examples of hand-drawn numerals. While we stress that our work is meant to capture shape similarity judgements in a way that is more general than work that is specific OCR, this provides a simple, interesting example. The characters are somewhat challenging, as they deliberately show quite different writing styles. In this example we also consider the cost of the gap between the beginning and the end of each numeral, as described in Section 5.

Table 3 and Fig. 11 shows the cost of comparing all pairs of characters, using the spring model. Note that the function finds a lower cost in matching all instances of the same character than they do for any pair of different characters.

Next, Fig. 12 shows six table-like shapes, and Table 4 shows the result of comparing these shapes with our cost function. While it is difficult to say exactly how the functions should perform on this example, it is interesting to note that the function considers tables 4 and 5 to be the most similar, and tables 2 and 3 to be the most different. In general, performance seems in accord with our intuitions.

Finally, Figs. 13 and 14 show the outlines of five animals, and the relative similarity of each pair. Again, this provides a simple example in which the results accord with our intuitions.

While the focus of our paper has been on judging the similarity of contours, it is also interesting to understand how to find correspondences between points on two contours. Because elastic matching methods produce the minimum cost correspondence between two contours, they can also be used to find a correspondence in cases in which two contours are known to come from similar objects. For example, Fig. 15 shows correspondences found between the outlines of a dog and a horse. Such correspondences can provide clues to the common structure of the two shapes. Fig. 16 shows correspondences matching the images of a person, which can be used for the analysis of nonrigid motion.

In summary, the experiments demonstrate the relative advantages of our cost function in different situations. Overall, we see that this function performs well in a wide variety of situations.

7. Some remaining challenges

This paper has examined the potential effectiveness of a cost function based on local shape comparison. Obviously, we have not considered the role of global properties of shapes such as whether two shapes are both symmetric, or related by a single affine transformation. Nor have we considered other contour properties, such as contour texture. It is clear that our work addresses only some of the issues in shape similarity. In this section, we raise three, somewhat more subtle issues in shape similarity that we feel have not been well addressed by our work, or by any other work that has come to our attention.

First we consider the question of whether we should compare shapes by comparing their contours or their internal regions. Consider Fig. 17. The local distortion one must apply to the figure on the left to produce the figures in the middle and on the right is identical. In each case, two vertical lines are stretched by a factor of two. And yet, we feel that the middle figure clearly resembles the figure on the left more than does the figure on the right. While this might be due to effects of symmetry and collinearity, one other possible explanation for this phenomena can be found by interpreting each countour as the boundary of a 2D material. Although the contours are equally distorted in the two figures, the internal 2D regions are distorted in very different ways. This suggests that our general approach might be improved by considering a cost based on a mapping between 2D regions bounded by the contours that we compare. A related example is also discussed by Mumford [37].

Table 3
Results using the spring model (Eq. (3))

	3	3	3	3	5	5	5	6	6
3	0	3.97	3.90	5.06	11.91	11.15	12.32	18.47	18.59
3	3.97	0	2.43	2.82	11.20	10.25	11.58	18.01	18.22
3	3.90	2.43	0	3.90	10.31	11.07	10.97	18.19	18.34
3	5.06	2.82	3.90	0	11.14	9.92	11.73	15.55	16.45
5	11.91	11.20	10.31	11.14	0	2.17	2.35	9.82	9.73
5	11.15	10.25	11.07	9.92	2.17	0	3.10	9.38	10.10
5	12.32	11.58	10.97	11.73	2.35	3.10	0	9.51	10.26
6	18.47	18.01	18.19	15.55	9.82	9.38	9.51	0	2.96
6	18.59	18.22	18.34	16.45	9.73	10.10	10.26	2.96	0

One can imagine an approach, analogous to ours, in which one seeks the continuous one-to-one mapping from one 2D region to another that minimizes the stretching and bending in the 2D mapping. Such an approach faces several obstacles, including determining the appropriate cost function for such a mapping, finding efficient methods of optimizing such a cost function (dynamic programming does not easily extend to 2D mappings), and understanding the implications of such an approach in modeling human perceptions. However, it is possible that such an approach could be fruitful, since the example in Fig. 17 suggests that human perception is sensitive to the amount of distortion of the 2D regions enclosed by a contour, not just to the amount of distortion in the contour when viewed as a 1D object.

However, for a possible counterexample to this argument, consider Fig. 18. When comparing the shapes on the extreme left and the extreme right, the regions enclosed by the contours appear to be hardly distorted at all. There is only a bending of the thin middle region connecting the regions on either side of the object. On the other hand, the middle shape seems to have a highly distorted interior. None-the-less, the authors' intuitions are that the middle figure seems more similar to the figure on the left than does the right-most figure. At least it does not appear that the middle figure is much more distorted than the right-most one. This suggests that for this figure, at least, comparison based on the amount of distortion of the region bounded by the contour would not lead to good results.

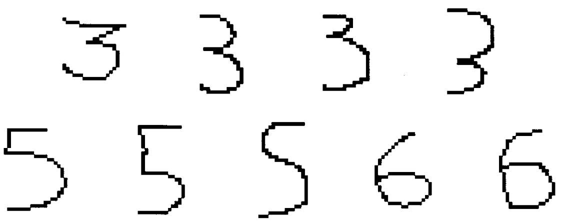


Fig. 11. The numerals used in experiments.

Both of these figures are meant merely to be suggestive; the desired behavior of any system may not be obvious. These figures merely raise the issue of whether any method based either on deforming contours or internal regions can handle all cases. We suggest also that these examples may prove challenging to many other approaches to judging shape similarity, as well as to our own.

Second, Fig. 19 shows another example that will be challenging to our approach, and perhaps to others. Despite the obvious similarity between the two shapes, there does not appear to be any good way of comparing them in terms of a correspondence between their contours, regions, or parts. Perhaps this example is better thought of as a problem in texture, rather than shape. However, it points out the difficulty of applying a single, general approach to judging shape similarity.

Third, in Fig. 20 we show a complex shape, with one part shifted to a new position. While our approach could ignore the shifting limb as occluded, and match the remainder of the shape, this seems to ignore useful information. It seems more desirable if one can note that these two limbs are identical, but shifted to different positions. Providing a specific mechanism for doing this, however, appears difficult.

These three examples are meant to illustrate the challenging nature of the general shape similarity problem. In our work, we have attempted to develop one approach, based on local deformations, as completely as possible. However, it is clear that this approach can only provide one piece of the solution to the general shape similarity problem.

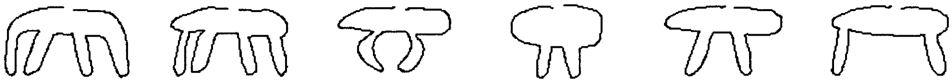


Fig. 12. Six table-like shapes used in experiments.

8. Conclusions

This paper has considered the problem of capturing human intuitions about shape similarity in a cost functions based on local deformations. In doing so we have identified novel constraints on the form of such cost functions, in particular considering how such costs can model the part-based nature of objects, and studying the behavior of cost functions that can be applied to both smooth and polygonal shapes. We have derived three novel cost functions from physical models of contours, and described experiments that demonstrate some of the strengths and weaknesses of each cost function. Hopefully, we also lay the groundwork for alternate approaches to judging shape similarity based on local deformations. For example, our work may be extended by considering the deformation of 2D regions rather than their 1D bounding contours, or by considering the significance of other local contour properties, such as curvature extreme.

Primarily, we have attempted to demonstrate that computational models of shape similarity should pay close attention to the quantitative, metric properties of shapes. These quantitative properties should be considered in ways that relate to the more qualitative, part-based nature of objects. However, we feel that a promising approach to part-based analysis is one that considers parts as a continuous property, rather than committing to an all-or-nothing decomposition of a shape into parts, since this latter approach does not degrade gracefully for noisy, occluded or mildly distorted shapes. In sum, if one feels that as a contour is distorted it becomes less and less similar to its original shape in a way that depends on the type of distortion, then one of the key problems of shape similarity is to understand the relationship between the type of distortion and the change in similarity. Our work systematically addresses this problem.

Acknowledgements

The authors would like to thank Warren Smith for several helpful suggestions. The vision group at the Weizmann Institute is supported in part by the Israeli Ministry of Science, Grant No. 8504. R. Basri is an incumbent of Arye Dissentshik Career Development Chair at the Weizmann Institute. D. Geiger was supported by AFOSR under F 49620-96-1-0159 and F 49620-96-1-0028 and a CAREER award from the NSF.

Appendix A. Properties of the spring model

In this appendix we show how the spring model satisfies the desired properties described above. In particular it is interesting to note that for this model to become a metric we need to consider a q root where $q > p(p-1)$ (as we will show). To satisfy all the properties we obtain that $p \geq 1$.

Let us be more precise. We will concentrate on the bending cost and assume that $k_2 > k_1 > 0$ and $t' = 1$. To obtain the formulae for the other cases we perform the same type of manipulations. Thus, we start with the cost function $F(k_2, k_1) = (k_2 - k_1)^p(k_2t' + k_1)^{1-\mathrm{p}}$.

(1.) F is continuous. That is by construction.

(2.) Metric properties:

(a) $F(k_1, k_2, t') \geq 0$. This is clearly satisfied.

(b) $F(k_1, k_1, 1) = 0$; $F(k_1, k_2, t') > 0$ for $k_1 \neq k_2$ or $t' \neq 1$. Both are clearly satisfied.

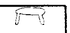
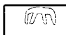
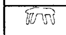
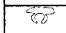
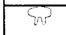
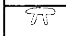
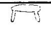
(c) The triangle inequality is not satisfied for this function. However, in the end of this appendix we show that by taking the total bending cost as $E^{1/q}_{\text{bending}}$, with $q > p(p-1)$ we can satisfy the triangle inequality.

(3.) To guarantee that our cost is scale invariant we need $F(k_1, k_2, t') = F(k_1, k_2/\alpha, \alpha t')$. This is satisfied since

$$\begin{aligned}F(k_1, k_2, t') &= (k_2t' - k_1)^p(k_2t' + k_1)^{1-p}\\ &= \left(\frac{k_2}{\alpha}\alpha t' - k_1\right)^p\left(\frac{k_1}{\alpha}\alpha t' + k_1\right)^{1-p}\\ &= F\left(k_1, \frac{k_2}{\alpha}, \alpha t'\right).\end{aligned}$$

Note that this is the case when the stretching term is not considered. This is to say that our measure is not scale invariant, but is instead divided into a scale-invariant bending cost and a stretching cost that penalizes scaling.

Table 4
Results for table-like shapes in Fig. 12 using the spring model.

						
	0	11.61	13.23	12.13	10.11	15.48
	11.61	0	18.21	16.90	15.84	18.54
	13.23	18.21	0	9.33	8.23	11.99
	12.13	16.90	9.33	0	6.16	9.31
	10.11	15.84	8.23	6.16	0	9.50
	15.48	18.54	11.99	9.31	9.50	0

(4.) Finite corner cost.

$$\lim_{k_2 \to \infty} F(k_2, k_1) = k_2$$

(5.) The constraint that it is easier to change the curvature of a contour at points where curvature is already high implies that:

$$\frac{\partial F}{\partial k_1} + \frac{\partial F}{\partial k_2} \leq 0 \qquad \textit{for } k_2 > k_1$$

We consider this in conjunction with the next constraint.

(6.) The constraint that increased bending also increases dissimilarity is equivalent to:

$$\frac{\partial F}{\partial k_1} \leq 0 \quad \text{and} \quad \frac{\partial F}{\partial k_2} \geq 0 \qquad t' = 1, k_2 > k_1,$$

A similar constraint holds for stretching. We will not explicitly show that the constraint on stretching holds, since the cost function and constraint that we get by allowing t' to vary with $k_1 = k_2 = 1$ are identical to the case where $t' = k_1 = 1$ and k_2 is allowed to vary. We can show this constraint holds for $p \geq 1$, as follows:

$$\frac{\partial F}{\partial k_1} = (k_2 - k_1)^{p-1}(k_2 + k_1)^{-p}$$
$$\times [-p(k_2 + k_1) + (1-p)(k_2 - k_1)]$$

$$\frac{\partial F}{\partial k_1} = (k_2 - k_1)^{p-1}(k_2 + k_1)^{-p}[(1-2p)k_2 - k_1] \leq 0 \quad \Rightarrow$$

$$p \geq \frac{1}{2} - \frac{k_1}{2k_2},$$

$$\frac{\partial F}{\partial k_2} = (k_2 - k_1)^{p-1}(k_2 + k_1)^{-p}$$
$$\times [p(k_2 + k_1) + (1-p)(k_2 - k_1)]$$

$$\frac{\partial F}{\partial k_2} = (k_2 - k_1)^{p-1}(k_2 + k_1)^{-p}[k_2 + (2p-1)k_1] \leq 0 \quad \Rightarrow$$

$$p \geq 0,$$

$$\frac{\partial F}{\partial k_1} + \frac{\partial F}{\partial k_2} = 2(1-p)(k_2 - k_1)^p(k_2 + k_1)^{-p} \leq 0 \quad \Rightarrow$$

$$p \geq 1$$

(7.) We need to show:

$$nF(k_1, k_1 + \varepsilon, 1) < F(k_1, k_1 + n\varepsilon, 1)$$

assuming WLOG that $k_1 > 0$ and assuming also that $\epsilon < 0, n < 1$. For the function under consideration, this means:

$$\frac{n\varepsilon}{2k_1 + \varepsilon} < \frac{n^2\varepsilon^2}{2k_1 + n\varepsilon}$$

This follows from straight-forward algebraic manipulation.

We next show that the cost based on a spring model can satisfy the triangle inequality if the qth root of the cost function taken. We do this only for the case of bending, with no stretching.

A.1. Triangle Inequality

Let us consider an infinitesimal contour element ds. The cost to match ds to dt in the other contour is given by:

$$\mathrm{d}E_{\text{bend}} = (k_2 t' - k_1)^p (k_2 t' + k_1)^{1-p} \mathrm{d}s$$
$$= (k_2 \mathrm{d}t - k_1 \mathrm{d}s)^p (k_2 \mathrm{d}t - k_1 \mathrm{d}s)^{1-p}.$$

Thus, we can rename the variables as $\theta_2 = k_2 \mathrm{d}t$ $\theta_1 = k_1 \mathrm{d}s$, and $C(\theta_i, \theta_j) = (\mathrm{d}E_{\text{bend}})^{1/q}$, and

$$C(1, 2) = (\mathrm{d}E_{\text{bend}})^{1/q} = (\theta_2 - \theta_1)^{p/q}(\theta_2 + \theta_1)^{(1-p)/q}.$$

We would like to show that $(\mathrm{d}E_{\text{bend}})^{1/q}$ satisfies the triangle inequality. Note that if $(\mathrm{d}E_{\text{bend}})^{1/q}$ satisfies the triangle inequality, so does $(E_{\text{bend}})^{1/q}$ (where E_{bend} is the cost of bending for the entire contour) since

$$A_i^{1/q} + B_i^{1/q} \geq C_i^{1/q} \quad \forall i$$
$$\Rightarrow \left(\sum_i A_i\right)^{1/q} + \left(\sum_i B_i\right)^{1/q} \geq \left(\sum_i C_i\right)^{1/q}.$$

Thus, it is enough to show that

$$C(1, 2) + C(2, 3) \geq C(1, 3), \tag{6}$$

For any choice of $\theta_1, \theta_2, \theta_3$. Without any loss of generality let us assume $0 \leq \theta_1 \leq \theta_2 \leq \theta_3$. We first note

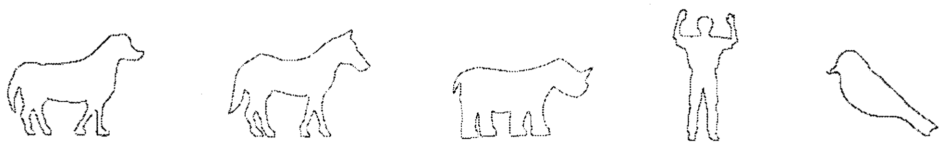


Fig. 13. Line drawings of five animals. All the contours used have almost the same size (up to 5% difference).

that for $\theta_1=\theta_2$ or $\theta_2=\theta_3$ the inequality becomes the equality since, $C(A,A)=0$. We then investigate the conditions such that by decreasing θ_1, starting from $\theta_1=\theta_2$, the inequality is always preserved (for any value of θ_2 and θ_3). More precisely,

$$-\frac{\partial C(1,2)}{\partial\theta_1}\ge -\frac{\partial C(1,3)}{\partial\theta_1} \quad \text{and}$$

$$C(1,2)+C(2,3)\ge C(1,2) \quad\Rightarrow$$

$$C(1-\varepsilon,2)+C(2,3)\ge C(1-\varepsilon,3),$$

$\varepsilon < < 1$, and if this is true for any θ_1, θ_2, θ_3 we can start at $\theta_1=\theta_2$ (where the equality is satisfied) and repeat the process of decreasing θ_1 until $\theta_1=0$ (the bottom). Thus, we need to determine when

$$\frac{\partial C(1,2)}{\partial\theta_1}\ge\frac{\partial C(1,3)}{\partial\theta_1}.$$

Again, we can guarantee this to be true for any θ_2 and $\theta_3\ge\theta_2$, if $\frac{\partial^2 C(1,2)}{\partial\theta_1\,\partial\theta_2}\ge 0$, since

$$\frac{\partial C(1,3)}{\partial\theta_1}|\theta_3=\theta_2+\varepsilon-\frac{\partial C(1,2)}{\partial\theta_1}|\theta_2\ge 0 \quad \text{for any } \theta_2$$

$$\Rightarrow\frac{\partial^2 C(1,2)}{\partial\theta_1\,\partial\theta_2}\ge 0.$$

Cost					
	0	4.82	7.22	17.88	20.59
	4.82	0	7.94	18.61	20.19
	7.22	7.94	0	19.34	21.50
	17.88	18.61	19.34	0	18.07
	20.59	20.19	21.50	18.07	0

Fig. 14. Applying our similarity measure to the five animals. The four-legged animals have similar matching scores (costs) and much less than when matched to the person or to the bird. Within the fours-legged group, the most distinct is the rhinoceros, whose thick legs and horn increased the cost significantly.

Then, by starting at any θ_2 and incrementing it, one can reach any θ_3 and the property $-\frac{\partial C(1,2)}{\partial\theta_1}\ge-\frac{\partial C(1,3)}{\partial\theta_1}$ will be satisfied. We have reduced the problem to one of showing under what conditions on q

$$\frac{\partial^2 C(1,2)}{\partial\theta_1\,\partial\theta_2}=-1q(\theta_2-\theta_1)^{(p/q)-2}(\theta_2+\theta_1)^{[(1-p)/q]-2} \times\{(1-q)\theta_2^2+(2p-1)(1-q)\theta_1^2 +2[2p(p-1)+(1-q)]\theta_1\theta_2\}\ge 0.$$

Thus, for what values of q is

$$(1-q)\theta_2^2+(2p-1)(1-q)\theta_1^2 +2[2p(p-1)+(1-q)]\theta_1\theta_2\le 0 \quad ?$$

It is clear that $(1-q)+2p(p-1)\le 0\to q\ge 1+2p(p-1)$ does satisfy this inequality, since the first two terms are negative (for $q\ge 1$). This implies, for example $p=1\to q\ge 1$ (which is saying that the linear cost is a metric, obviously) and $p=2\to q\ge 5$ which is quite a loose bound. One can examine this equation more closely, and ask the condition for the function $E(\theta_1,\theta_2)=(q-1)\theta_2^2+(2p-1)\theta_1^2+2[(q-1)-2p(p-1)]\theta_1\theta_2$ to be positive. We can show that for the two eigenvalues of the Hessian matrix to be positive (convex function) we need $q-1\ge p(\sqrt{2p-1}-1)$ and they are the same as to guarantee that the minimum of the function is positive. Thus, we obtain a tighter bound on $q, q>1+p(\sqrt{2p-1}-1)$ which gives approximately for $p=2\to q\ge 2.4$.

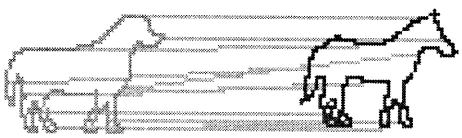


Fig. 15. The matching helps to visualize how the local computations tend to preserve object parts (e.g. legs go with legs, head with head).

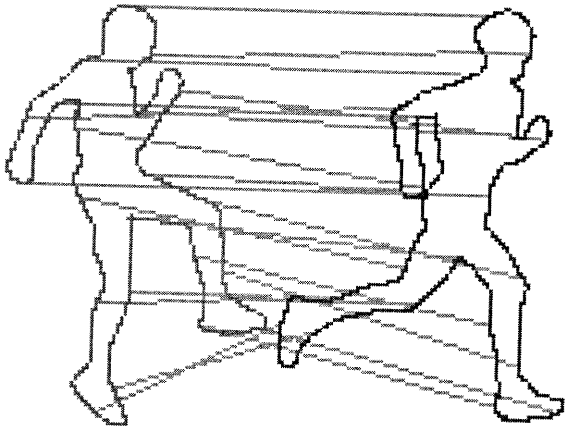


Fig. 16. This figure shows the correspondence obtained between two contours produced by a person with their limbs in different positions.

Appendix B. Cost based on continuous deformation

Below we consider a cost function based on the intuitive idea that we slowly bend and stretch one contour into the other, with the total cost equal to the sum of an infinite number of infinitesimal deformations. That is, we assume that Γ_1 and Γ_2, are connected by a continuous series of contours, and that the cost of deforming Γ_1 to Γ_2 is the sum of the costs of deforming each contour into the next one in the series.

To begin, we consider the form such a cost function takes in the absence of stretching (i.e. $t' = 1$). Here we assume that the cost of deforming one curvature into another is the sum of a series of small deformations, that is:

$$F(k_1, k_2, 1) = \left| \int_{k_1}^{k_2} g(k)\mathrm{d}k \right|$$

for some function g. Intuitively, g is a function that describes how difficult it is to deform a contour point infinitesimally, as a function of the curvature at that point. This implies that we may write:

$$F(k_1,k_2, 1) = |f(k_2) - f(k_1)|$$

for some f, which is the integral of g.

F will satisfy constraint 3.1.6 if and only if g decreases monotonically as the curvature increases, for positive curvature, or decreases for negative curvature. This is equivalent to assuming that an object is easier to bend as the magnitude of curvature increases.

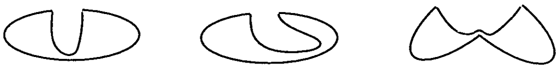


Fig. 18. A figure, with two possible distortions.

One reason for choosing a cost function of this type is the underlying, intuitive physical model of one contour gradually changing into the other. A second reason follows from our constraints. If constraint 3.1.6 and the triangle inequality hold, we have, for $0 < k_1 < \mathrm{k}_2$:

$$F\left(k_1, \frac{k_1 + k_2}{2}, 1\right) + F\left(\frac{k_1 + k_2}{2}, k_2, 1\right) \geq F(k_1, k_2, 1)$$

$$F\left(k_1, \frac{k_1 + k_2}{2}, 1\right) > F\left(\frac{k_1 + k_2}{2}, k_2, 1\right)$$

This implies that:

$$2F\left(k_1, \frac{k_1 + k_2}{2}, 1\right) \geq F(k_1, k_2, 1)$$

violating constraint 3.1.7. However, if we wish to violate this constraint to the smallest possible degree, we should minimize:

$$2F\left(k_1, \frac{k_1 + k_2}{2}, 1\right) - F(k_1, k_2, 1)$$

This can be done in two ways, by minimizing:

$$F\left(k_1, \frac{k_1 + k_2}{2}, 1\right) - F\left(\frac{k_1 + k_2}{2}, k_2, 1\right)$$

or by setting:

$$F\left(k_1, \frac{k_1 + k_2}{2}, 1\right) + F\left(\frac{k_1 + k_2}{2}, k_2, 1\right) = F(k_1, k_2, 1)$$

The first approach simply weakens the effect of constraint 3.1.6, but the second approach has no apparent disadvantage. Therefore, we desire a cost function for which the triangle inequality is at equality when changes in curvature are all in the same direction, that is:

$$F(k_1, k_2,1) + F(k_2, k_3, 1) = F(k_1, k_3, 1)$$

for $k_1 \leq k_2 \leq k_3$

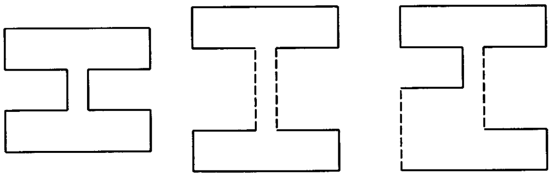


Fig. 17. The figure on the left is distorted into the other two figures by stretching each of the dashed lines by a factor of 2.

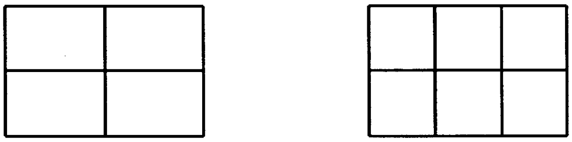


Fig. 19. Two figures that appear similar, although their contours, regions and part structure seem quite different.

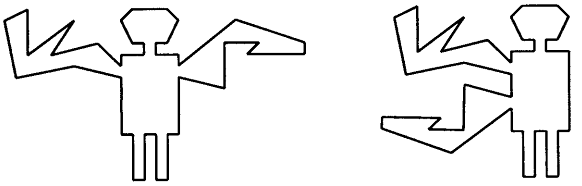


Fig. 20. A figure, with one limb moved to a new position.

This implies in fact that $F(k_1, k_3, 1)$ is exactly the sum of a series of intermediate changes in curvature. So our model, which measures the cost of gradually deforming one contour into another, produces cost functions that will least violate constraint 3.1.7, so that smoothing will have the best chance of overcoming this effect to satisfy constraint 3.1.7.

We now wish to add the effect of stretching (i.e. t') to our cost function. We can make our function symmetric, and provide an additive stretching penalty by altering it so that:

$$F(k_1, k_2, t') = [(t' + 1)|f(k_2) - f(k_1)| + \lambda|t'' - 1|]\mathrm{d}s$$

The right side represents a stretching cost when there is no bending, the left side gives a bending cost in the absence of stretching. Notice that this is symmetric because:

$$F(k_1, k_2, t') = \left[\left(\frac{\mathrm{d}t}{\mathrm{d}s} + 1\right)|f(k_2) - f(k_1)| + \lambda\left|\frac{\mathrm{d}t}{\mathrm{d}s} - 1\right|\right]\mathrm{d}s$$

$$F(k_1, k_2, t') = [(\mathrm{d}t + \mathrm{d}s)|f(k_2)\mathrm{d}t - f(k_1)\mathrm{d}s| + \lambda|\mathrm{d}t - \mathrm{d}s|]$$

$$F(k_1, k_2, t') = F\left(k_2, k_1, \frac{\mathrm{d}s}{\mathrm{d}t}\right)$$

λ is a positive parameter that reflects the relative weight of stretching and bending costs. Note that this stretching meets the metric constraint.

We now propose a particular form for the function, with which we will experiment. We let:

$$g(k) = e^{-\alpha|k|} + c$$

and therefore:

$$f(k) = ck - e^{-\alpha k}, \quad 0 < k$$

$$f(k) = ck + e^{\alpha k}, \quad k < 0$$

This is a simple, intuitive function that fulfils our needs. It is asymptotically linear in k, and therefore the cost function will be bounded at corners, even for arbitrarily small amounts of smoothing, as discussed in Section 3. Added to this is a term that causes the difficulty of bending a contour to decrease exponentially with curvature. The parameter c controls the relative weight of these terms, while α determines the rate at which the exponential term falls off. Fig. 21 illustrates the function.

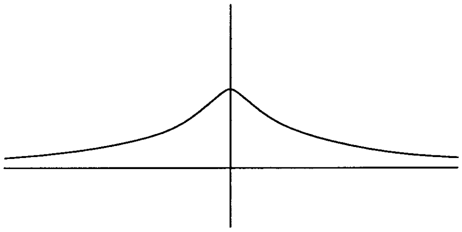


Fig. 21. The function $e^{-\alpha|k|} + c$, for $\alpha = 1, c = 0.1$.

It is readily verified that this function meets our constraints with the following exceptions. First, its behavior under uniform scaling is not considered, except in that we can readily see that when matching two lines of different length, the cost will be the same for all correspondences. Second, as previously noted, this function does not satisfy condition 3.1.7.

References

[1] Amit Y, Grenander U, Piccioni M. Structural image restoration through deformable templates. J Am Statis Assoc 1991;86(414):376–87.

[2] Arkin E, Chew P, Huttenlocher D, Kedem K, Mitchell J. An efficiently computable metric for comparing polygonal shapes. IEEE Trans on PA MI 1991;13(3):209–16.

[3] Bajcsy R, Solina F. Three dimensional object representation revisited. Proc First ICCV Conf Lond 1987;231–240.

[4] Basri R, Costa L, Geiger D, Jacobs D. Determining the similarity of deformable objects. IEEE Workshop on Physics-based Modeling in Computer Vision 1995;135–143.

[5] Baumberg A, Hogg D. Learning flexible models from image sequences. ECCV 1995;299–308.

[6] Biederman I. Human image understanding: recent research and a theory. Comp Graph Vis Image Process 1985;32:29–73.

[7] Binford TO. Visual perception by computer. IEEE Conf Sys Control 1971.

[8] Blake A, Zisserman A. Visual Reconstruction. Cambridge, MA: MIT Press, 1987.

[9] Bolles R, Cain R. Recognizing and locating partially visible objects: the local feature-focus method. Int J Robot Res 1982;1(3):57–82.

[10] Brooks R. Symbolic reasoning among 3D models and 2D images. Artif Intell 1981;17:285–348.

[11] Burr D. Elastic matching of line drawings. IEEE Trans Pattern Anal Mach Intell 1981;3(6):708–13.

[12] Cohen I, Ayache N, Sulger P. Tracking points on deformable objects using curvature information. Eur Conf on Comp Vis 1992;458–466.

[13] Connell JH, Brady M. Generating and generalizing models of visual objects. Artif Intell 1987;31:159–83.

[14] Cootes TF, Taylor CJ, Cooper DH, Graham J. Training models of shape from sets of examples. Proc Br Mach Vis Conf Springer-Verlag 1992;9–18.

[15] Cormen T, Leiserson C, Rivest R. Introduction to Algorithms. Cambridge, MA: MIT Press, 1990.

[16] Duda RO, Hart PE. Pattern Classification and Scene Analysis. New York: Wiley-Interscience, 1973.

[17] Geiger D, Gupta A, Costa L, Vlontzos J. Dynamic programming for detecting, tracking and matching deformable contours. IEEE Trans Pattern Anal Mach Intell 1995;17(3):294–302.

[18] Geman S, Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Mach Intell 1984;6(7):721–41.

[19] Gorman J, Mitchell R, Kuhl F. Partial shape recognition using dynamic programming. IEEE Trans Pattern Anal Mach Intell 1988;10(2):257–66.

[20] Grimson W. On the Recognition of Parameterized Objects. AI Memo 1108. Cambridge, MA: MIT 1987.

[21] Hel-Or Y, Werman M. Constraint-fusion for interpretation of articulated objects. IEEE Conf on Comp Vis Pattern Recog 1994:39–45.

[22] Hildreth E. The Measurement Of Visual Motion. Cambridge, MA: MIT, 1983.

[23] Hinton G, Williams C, Revow M. Adaptive elastic models for hand-printed character recognition. NIPS 4:512–519.

[24] Ho S. Representing and using functional definitions for visual recognition, PhD dissertation. University of Wisconsin, Madison 1987.

[25] Hoffman D, Richards W. Parts of Recognition. In: Pinker S, editor. Visual Cognition. Cambridge, MA: MIT Press, 1984.

[26] Huttenlocher D, Klanderman G, Rucklidge W. Comparing images using the Hausdorff distance. IEEE Trans Pattern Anal Mach Intell 1993;15(9):850–63.

[27] Huttenlocher D, Noh J, Rucklidge W. Tracking non-rigid objects in complex scenes. Fourth Int Conf Comp Vis 1993;93–101.

[28] Jain A, Zhong Y, Lakshmanan S. Object matching using deformable templates. IEEE Trans Pattern Anal Mach Intell 1996;18(3):267–78.

[29] Kimia B, Tannenbaum A, Zucker S. Shapes, shocks and deformations. Int J Comp Vis 1995;15:189–224.

[30] Koenderink JJ, van Doom AJ. The shape of smooth objects and the way contours end. Perception 1981;11:129–37.

[31] Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. Int J Comp Vis 1988;1(4):321–31.

[32] Kupeev K, Wolfson H. On shape similarity. Proc Int Conf Pattern Recog 1994;227–237.

[33] Levin E, Pieraccini R. Dynamic planar warping for optical character recognition. ICASSP III 1992;149–152.

[34] Marr D, Nishihara H. Representation and recognition of the spatial organization of three dimensional structure. Proc R Soc Lond B 1978;200:269–94.

[35] McConnell R, Kwok R, Curlander J, Kober W, Pang S. S correlation and dynamic time warping: two methods for tracking ice floes in SAR images. IEEE Trans Geo Rem Sensing 1991;29(6):1004–12.

[36] Mehrotra R, Grosky W. Shape matching utilizing indexed hypotheses generation and testing. IEEE Trans Robot Autom 1989;5(1):70–7.

[37] Mumford D. Mathematical theories of shape: do they model perception? SPIE Vol 1570. Geom Meth Comp Vis 1991;2–10.

[38] Mumford D, Shah J. Boundary detection by minimizing functionals, I. Proc IEEE Conf Comp Vis Pattern Recog 1985.

[39] Pentland A. Recognition by parts. Proc First Int Conf Comp Vis 1987;612–620.

[40] Pentland A, Sclaroff S. Closed-form solutions for physically based shape modeling and recognition. IEEE Trans Pattern Anal Mach Intell 1991;13(7):715–29.

[41] Pope A, Lowe D. Learning object recognition models from images. Fourth Int Conf Comp Vis 1993;296–301.

[42] Rivlin E, Dickenson S, Rosenfeld A. Recognition by functional parts. IEEE Conf Comp Vis Pattern Recog 1994;267–275.

[43] Sankoff D, Kruskal J,(editors). Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison. Reading Ma: Addison-Wesley, 1983.

[44] Siddiqi K, Kimia B.A shock grammar for recognition. Proc IEEE Conf Comp Vis Pattern Recog 1996;507–513.

[45] Siddiqi K, Tresness K, Kimia B. Parts of visual form: ecological and psychophysical aspects. Perception 1996;25:399–424.

[46] Stark L, Bowyer K. Achieving generalized object recognition through reasoning about association of function to structure. IEEE Trans Pattern Anal Mach Intell 1991;13(10):992–1006.

[47] Tagare H, O'Shea D, Rangarajan A. A geometric criterion for shape-based non-rigid correspondence Fifth Int Conf Comp Vis 1995;434–439.

[48] Tappert C. Cursive script recognition by elastic matching. IBM J Res Develop 1982;26(6):765–71.

[49] Tsai W, Yu S. IEEE Trans Pattern Anal Mach Intell 1985;7:453–462.

[50] Tsukumo J. Handprinted Kanji character recognition based on flexible template matching. Eleventh ICPR 1992;483–486.

[51] Tsukumo J, Tanaka H. Classification of handprinted chinese characters using non-linear normalization and correlation methods. Ninth ICPR 1988;168–171.

[52] Tversky A. Features of similarity. Psycholog Rev 1977;84(4):327–52.

[53] Ullman S. Aligning pictorial descriptions: an approach to object recognition. Cognition 1989;32(3):193–254.

[54] Weiss I. 3D shape representation by contours. CVGIP 1988;41(1):80–100.

[55] Winston PH, Binford TO, Katz B, Lowry M. Learning Physical Description from Functional Definitions, Examples and Precedents. AI Memo 679. Cambridge,MA: MIT, 1984.

[56] Yoshida K, Sakoe H. Online handwritten character recognition for a personal computer system. IEEE Trans Consum Electron CE 1982;28(3):202–9.

[57] Yuille A, Cohen D, Hallinan P. Feature extraction from faces using deformable templates. CVPR 1989;104–109.

[58] Zhu S, Yuille A. FORMS: A flexible object recognition and modelling system. Harvard, TR 1994;94–1.