Determining the Similarity of Deformable Shapes

Ronen Basri — Dept. of Applied Math., The Weizmann Inst. of Science, Rehovot, 76100, Israel

Luiz Costa — Lab. of Integrated Systems, University of Sao Paulo, Sao Paulo, Brazil

Davi Geiger — Courant Institute, New York University, New York, NY, 10012

David Jacobs — NEC Research Institute, 4 Independence Way, Princeton, NJ 08540

Abstract

We study how to measure the degree of similarity between two image contours. We propose an approach for comparing contours that takes into account deformations in object shape, the articulation of parts, and variations in the shape and size of portions of objects. Our method uses dynamic programming to compute the minimum cost of bringing one shape into the other via local deformations. Using this as a starting point, we investigate the properties that such a cost function should have to model human performance and to perform usefully in a computer vision system. We suggest novel conditions on this cost function that help capture the part-based nature of objects without requiring any explicit decomposition of shapes into their parts. We then suggest several possible cost functions based on different physical models of contours, and describe experiments with these costs.

1 Introduction[1]

Detecting similarity is a key tool in interpretating images. In this paper we develop a measure of similarity between line drawings of objects. This will be used to determine the class identity ("categorization") of objects and to cluster objects into classes.

A good deal of work on categorization has emphasized the part structure of objects (eg. [5, 6, 23, 7, 2, 25, 9, 20]). We believe that part structure plays an important role in categorization, but we believe also that metric information has a significant influence on our perception of shapes. As an example, consider the two objects in Figure 1. The objects share the same set of parts and differ only in their metric properties. Yet one of the objects looks like a pair of spectacles, whereas the other does not. We argue that shape similarity is a continuous phenomenon. The similarity between two shapes deteriorates as one shape deforms with respect to the other even when they preserve the same part structure. Therefore, while there is much evidence that a measure of shape similarity should account for the salience of part structure, we feel that it is desirable to provide an integrated account of similarity that incorporates metric variation as well.

A second approach to shape similarity focuses exclusively on metric variation. This approach compares two shapes by measuring the amount of deformation required to register the shapes exactly, often using dynamic time-warping or gradient descent for computing the amount of deformation required. There has been a great deal of work on this approach. For example, it has been applied to tracking non-rigid objects (eg., [14, 19, 8, 24]) where the amount of deformation between frames may be assumed to be relatively small. This approach has also been widely applied to handwriting recognition (eg., [30, 38, 33]) using measures of similarity that tend to be particularly crafted to that domain. Deformable templates that reflect the specific shape of known objects have also been applied to contour extraction and modelling ([39, 31, 21]). Of particular relevance to our work: [32] suggest such an approach to judging the similarity of objects; [1] also propose such a method of comparing shape, with particular attention to producing fast algorithms that yield a metric; and [22, 4, 15] suggest learning a deformable model from examples.

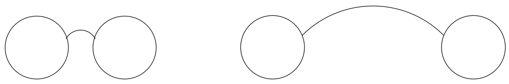

Figure 1: A pair of spectacles (left) and an object that has the same parts, but does not look like spectacles (right).

Work based on measuring deformations has not considered the role of part structure and articulation in shape similarity. Most of the above systems, for example, compare shapes by measuring the difference in tangent direction between corresponding contour points. Articulation of an object at a part boundary will introduce great differences throughout the part with this type of representation. One exception to this is [8], which uses a representation based on curvature in order to encourage matching curvature extrema between contours. While [8] suggest a cost function related to one we will suggest, they differ from us in their general motivation. They also differ from us in detail by not considering the consequences of applying such a cost measure to polygons.

Since determining shape similarity is a problem at the heart of computer vision, there is much work that addresses this problem with a wide variety of approaches. For example, work on comparing shapes with articulated parts are discussed in [7, 12, 13]. Another approach to shape comparison describes shapes

[1] This paper appered in the IEEE Workshop on Physics-Based Modeling in Computer Vision. pp. 135–143, 1995.

with a list of properties and their relations, using pattern matching techniques to determine similarity (eg. see [27], and a review in [10]). A third class of methods attempts to assign "semantic interpretation" to shapes, such as function, and make similarity judgements based on semantics (eg. [37, 16, 29, 28]). Also relevant to our work, [26] deform shapes by aligning the principal modes of their mass and stiffness matrices, [40] combine part and deformable template methods in a novel way quite different from ours, [18] compare shapes by minimizing the Hausdorff measure between them, and [35] proposes aligning shapes by applying piecewise affine transformations. See [3] for further discussion of this work.

Our approach: We consider the applicability of continuous-deformation methods to the more general problem of human similarity judgements. Below we propose a set of properties that will be desirable in any approach to shape similarity. In particular, we require the similarity function to be continuous, to form a metric, and to handle both polygons and smooth curves. To account for part structure we penalize deformations at the connection between parts less than we penalize deformations which modify the part structure of the object. Finally, we handle articulations by assigning cost to the local deformation at the point of articulation rather than to the entire shape. We then examine the implications of these properties.

Perhaps the most notable difference between our approach and past work is our consideration of how to prefer deformations that preserve the part structure of objects over deformations that modify the parts. By this, we hope to (i) bridge the gap between the class of methods that use part decomposition and between the class of methods that evaluate deformations and (ii) to better account for human perception of shapes.

2 Constraints on a Similarity Function

We shall define similarity through a distance function which when applied to pairs of objects returns the cost of deforming one object to the other. In this section we attempt to formulate a set of desirable properties of the similarity measure. We shall later use these constraints to find "reasonable" measures of similarity.

Let Γ_i denote a curve, and let $C(\Gamma_1, \Gamma_2)$ give the cost, or distance, between the two curves. We define a cost by finding a mapping between two contours, 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 write $t(s)$ as just t when s is clear). We consider cost functions that integrate over the curves a measure of the local difference between the curves, given this correspondence. The cost functions that we propose will depend on curvature, rather than other contour properties such as tangent direction, for reasons considered below. Therefore, we specifically propose that C have the form:

$$C(\Gamma_1, \Gamma_2) = \min \int_{\Gamma_1} F(k_1, k_2, \frac{dt}{ds}) ds. \tag{1}$$

k_1 is the curvature of Γ_1 at point $\Gamma_1(s)$, 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 ($\frac{dt}{ds}$, or t') required to make the curves identical (a similar viewpoint is taken in [8]). We now propose properties that C should possess[2]. We will then consider the implications of these properties on the function F.

2.1 Constraints

We propose the following constraints on C.

1. C is continuous.
2. C is a metric.
3. C is invariant under certain classes of transformations.
4. C should not diverge for nondifferentiable curves.
5. Bending a shape will cost less if the bending is done at a point of high curvature.
6. The cost between the deformed and original shape should grow as bending or stretching increases.
7. Many small deformations should effect shape less than one deformation of equal total magnitude.
8. Articulating an object should have a cost related to the local deformation at the point of articulation rather than a cost related to the overall shape of the articulated part.

2.2 Discussion

1. It is debatable whether human comparisons between shapes are smooth. Also, it is not clear how one should formally define continuity to be relevant to human perceptions, partly because human and machine perception do not encounter infinitesimal differences. However, for this work, we will require that C be continuous in the following sense. If the curvatures of a series of curves converge everywhere to the curvatures of some limit curve, then we require the cost between the curves and the limit curve to converge to zero. We will use this property to define our cost function on polygonal arcs, which have undefined curvatures at corners.

2. It will be useful for C to be a metric because virtually all efficient methods of finding nearest neighbors rely on the metric properties of a comparison function. However, it is unclear whether human comparisons of shape obey metric properties. For example, it is quite possible that people perceive a horse and a man to have very different appearances, while a centaur may

[2]The authors gratefully acknowledge Warren Smith's contributions in developing ideas in this section.

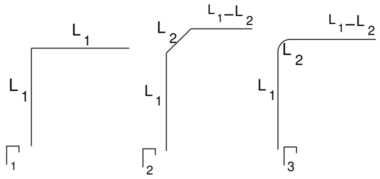

Figure 2: Our work uses the intuition that as L_2 decreases, the two figures on the right each become more similar to the figure on the left.

be quite similar to both, violating the triangle inequality. Also, [34] argues that human perception does not obey the symmetry constraint.

Given our formulation of C (Eq. 1), C is a metric if and only if its integrand, Fds, is too. This implies:

1. $F(k_1, k_2, t') \geq 0$.
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. $F(k_1, k_2, \frac{dt}{ds})ds = F(k_2, k_1, \frac{ds}{dt})dt$
4. $\forall k_1, k_2, k_3,\ F(k_1, k_2, \frac{dt}{ds})ds + F(k_2, k_3, \frac{du}{dt})dt \geq F(k_1, k_3, \frac{du}{ds})ds$, where k_3 indicates the curvature of a corresponding point on Γ_3, which is parameterized by u.

Note that even if C is non-metric, some monotonic function of C might provide a metric. For example, it is possible that even if C doesn't obey the triangle inequality, $\sqrt{C}$ might.

3. Since F is a function of curvature, it is invariant under Euclidean transformations.

4. This constraint ensures that the cost function can handle both smooth and polygonal shapes. 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 appears more similar to a circle than does a regular triangle (Figure 4). Condition 4 is a prerequisite to making such judgements of ideal, non-discretized contours. Below, we will consider relaxing the requirement of a purely one-to-one mapping, but for now we will further consider its implications for comparing contours with corners.

First, a problem arises because our cost function uses curvature, which is undefined at the corners of polygons. We overcome this problem using the continuity property described above to define F at corners. 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. Such a corner is the limit of a sequence of contours consisting of two lines joined by a circular arc of length l and curvature $\frac{\theta}{l}$, as l goes to 0. These are illustrated as Γ_1 and Γ_3 in Figure 2. To compare a smooth curve to a polygonal curve, we take the limit of the comparison obtained between the smooth curve and a series of curves that converges to the polygonal curve in this way.

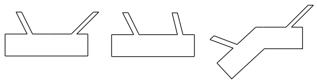

Figure 3: The middle figure shows the figure on the left, bent at points of high curvature. On the right, the figure is bent by equal amounts at points of low curvature. By penalizing the bending on the right more than the bending in the middle, we produce comparisons sensitive to the part structure of objects.

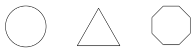

Figure 4: A circle, regular triangle, and regular octagon, with the same perimeter. We require a cost function to judge the octagon and circle more similar than the triangle and circle.

Next, we consider what properties F must have in order to provide a finite, non-zero cost when comparing corners. Suppose we match a corner point, with an angle θ_1, in Γ_1 to a smooth point that is on a small portion of Γ_2 with a constant curvature of k_2. Then, from Eq. (1) we find the cost of this comparison to be:

$$C(\Gamma_1, \Gamma_2) = \lim_{l \to 0}\ lF(\frac{\theta_1}{l}, k_2, 1) \qquad (2)$$

This implies that F must be linear in k_1, its first parameter, as k_1 approaches infinity. Similarly, F must be asymptotically linear in k_2.

One consequence of this is that the cost of transforming a point of finite curvature into a corner will depend linearly on the angle of the corner, since the cost function when applied to corners will depend only upon the asymptotic behavior of F. This implies, for example, that all convex polygons will have the same cost when compared to a circle, and leads, as we will see, to the violation of condition 7. We will discuss possible solutions to this problem in Section 2.3.

5. This condition reflects the phenomenon that contours appear more "bendable" at places where there is already a high curvature than at points of low curvature (see Figure 3). This reflects the part nature of objects; part boundaries tend to occur at points of high curvature. Articulated or deformable objects tend to bend more easily at part boundaries.

In terms of F this constraint implies:

$$F(k_1, k_2, 1) > F(k_1 + a, k_2 + a, 1), \qquad (3)$$

for all $a, k_1, k_2 > 0$ and for all $a, k_1, k_2 < 0$.

6. This condition reflects the intuition that increasing deformations of a shape should make them less and less similar to the original shape. This tells us: $k_1 < k_2 < k_3 \Rightarrow F(k_1, k_2, 1) < F(k_1, k_3, 1)$, and $1 < t'_1 < t'_2, k_1 = k_2 \Rightarrow F(k_1, k_2, t'_1) < F(k_1, k_2, t'_2)$.

7. 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. This principle is also illustrated psychophysically, by comparing a circle to a regular triangle, and to a regular octagon (see Figure 4). We believe that the octagon should be judged

to be more similar to the circle than the triangle is. 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. This constraint implies that: $nF(k_1, k_1 + \epsilon, 1) < F(k_1, k_1 + n\epsilon, 1), \forall k_1 > 0, \epsilon > 0, n > 1$. As is previously mentioned, this constraint is not met by cost functions that become linear for corners. However, we have already shown that only such cost functions will converge to finite values when comparing corners to smooth curve segments.

8. This condition seems central to the comparison of objects with movable parts; the cost of an articulation should perhaps depend on the amount of articulation, but not on the shape of the part that is articulated. This will be true of local cost measures, such as ours, that are invariant under rotation. It is not true of cost comparisons based on the difference in tangent between two contours, such as those of [1, 24, 32].

2.3 Problems with one-to-one matching

We have seen that problems arise in attempting to build a cost function that meets all the above conditions while making purely local comparisons. Such a cost must become linear for corners, which contradicts condition 7. We now point out another difficulty with a cost based on one-to-one comparisons.

An example of this problem is depicted in Figure 2. Γ_1 is a curve consisting of two straight lines each of length l_1, connected by a corner with an exterior angle of $\frac{\pi}{2}$. Γ_2 consists of 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 $\frac{\pi}{4}$. So, the two curves have the same length, and they all turn by a total angle of $\frac{\pi}{2}$. As $l_2 \rightarrow 0$, Γ_2 becomes more and more similar to Γ_1. Now, 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 intuitions.

We overcome this problem by adopting the view that we should instead compare small portions of the curves. 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 in the above example then as $l_2 \rightarrow 0$, 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 will make discrete comparisons, with some implicit smoothing. However, we see that we must rely on these sorts of effects if we want to be able to notice the similarity between a curve with one big change, and a curve with two, nearby changes that together produce the same effect.

Although we must rely on smoothing to produce a desirable cost function, our previous 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 keep the cost function from growing very large in the presence of corners, but will also 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. In either approach, smoothing will somewhat mitigate an undesirable effect that must occur without smoothing. See [36] for a related discussion on handling corners in functions describing the energy of a curve.

2.4 Incompatibility between Constraints

We now demonstrate that it is impossible to simultaneously satisfy all desired constraints on the function F. We find:

$$\begin{aligned} F(0,2,1) &\leq F(0,1,1) + F(1,2,1) < 2F(0,1,1) \\ &< F(0,2,1) \end{aligned} \tag{4}$$

where the first inequality comes from the triangle inequality, the second inequality follows from Constraint 5, and, Condition 7 implies the third inequality.

There are several possible responses to this situation. First, if we allow equality in Conditions 5 and 7 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 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 7 by smoothing out many small changes more than one big change.

3 Three Models for Cost Functions

We now consider potential cost functions, and discuss the extent to which they satisfy the constraints.

3.1 Model of Continuous Deformation

First, we consider cost functions based on the intuitive idea that as we bend and stretch one contour into the other the total cost equals 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 (ie. $t' = 1$). In this case the cost of bending one curvature into another is the sum of a series of small bendings:

$$F(k_1, k_2, 1) = |\int_{k_1}^{k_2} g(k)dk| \tag{5}$$

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

$$F(k_1, k_2, 1) = |f(k_2) - f(k_1)| \tag{6}$$

for some f, which is the integral of g. F will satisfy constraint 5 if and only if g decreases monotonically (ie. bending becomes easier) as the magnitude of the curvature increases.

We note that this form of F also follows from constraint (6) and from setting the triangle inequality to equality. This results in violating constraint (7) as shown in Section 2.4, but to the smallest possible degree. See [3] for details.

To add stretching (ie. t') to the cost function, we may provide an additive stretching penalty while making our function symmetric so that:

$$F(k_1, k_2, t') = ((t'+1)|f(k_2)-f(k_1)|+\lambda|t'-1|)ds. \tag{7}$$

The right term represents a stretching cost when there is no bending, the left term gives a bending cost in the absence of stretching. λ is a positive parameter that reflects the relative weight of stretching and bending costs. It can be readily verified that this function meets the metric constraint.

In experiments we let: $g(k) = e^{-\alpha|k|} + c$ and so:

$$f(k) = \begin{cases} ck - e^{-\alpha k}, & k > 0 \\ ck + e^{\alpha k}, & k < 0 \end{cases} \tag{8}$$

This function is asymptotically linear in k, and therefore the cost function will be bounded at corners, even for arbitrarily small amounts of smoothing. 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. It is readily verified that this function meets our constraints except for condition 7.

3.2 A Physical Model Based on Springs

Next we derive a cost based on a model of contour (shapes) formation from a concatenation of springs. The deformation of these 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.

Stretching: The energy associated with stretching a spring with stiffness α is given by

$$E = \frac{1}{p}\alpha(\Delta x)^p\,, \tag{9}$$

where Δx is the displacement from equilibrium, and $p > 1$. This equation generalizes Hooke's law in which $p = 2$. If two springs of stiffness α_1 and α_2 are connected then the resulting spring has stiffness

$$\alpha_{eq} = \frac{\alpha_1 \alpha_2}{\alpha_1^{1/(p-1)} + \alpha_2^{1/(p-1)}}\,. \tag{10}$$

Our main assumption 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 dl will have stiffness

$$\alpha_{dl} = \frac{\alpha}{(dl)^{p-1}}\,. \tag{11}$$

Two connected elements, dl_1 and dl_2, will yield an equivalent stiffness

$$\alpha_{eq} = \frac{\alpha_{dl_1}\alpha_{dl_2}}{\alpha_{dl_1}^{1/(p-1)} + \alpha_{dl_2}^{1/(p-1)}} = \frac{\alpha}{(dl_1 + dl_2)^{p-1}}\,. \tag{12}$$

Thus, a contour with length L will have an equivalent stiffness $\alpha_{contour} = \frac{\alpha}{L^{p-1}}$.

Matching contours: In order to compare two contours, Γ_s and Γ_t 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. We choose the infinitesimal "true" relaxed length to be $(dt+ds)/2$ (when matching element ds, from Γ_s, to element dt, from Γ_t). The stretching energy becomes

$$\begin{aligned} dE_{stretch}(ds, dt) &= \tfrac{1}{p}\tfrac{2^{p-1}\alpha}{(ds+dt)^{p-1}}|dt - (ds+dt)/2|^p + \\ &\quad \tfrac{1}{p}\tfrac{2^{p-1}\alpha}{(ds+dt)^{p-1}}|ds - (ds+dt)/2|^p \\ &= \tfrac{\alpha}{p}\tfrac{(t'-1)^p}{(t'+1)^{p-1}}ds\,, \end{aligned} \tag{13}$$

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

$$dE_{bending}(k_s, k_t, ds, dt) = \frac{\alpha|k_t\,dt - k_s\,ds|^p}{p(|k_s\,ds| + |k_t\,dt|)^{p-1}}ds \tag{14}$$

Thus, this model has total energy

$$E(\Gamma_s, \Gamma_t, t') = \frac{\alpha}{p}\int_{\Gamma_s}\Big[\frac{|k_t\,t' - k_s|^p}{(|k_t\,t'| + |k_s|)^{p-1}} + \lambda\frac{|t'-1|^p}{(t'+1)^{p-1}}\Big]ds\,. \tag{15}$$

where λ weights the relative contributions of stretching and bending.

In [3] we demonstrate that this cost function meets the previously discussed constraints, except that it is not a metric.

3.3 Linear Cost Function

Finally, we include a third possible cost function. As we have noted before, it is possible to satisfy both

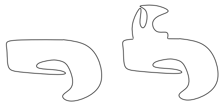

Figure 5: We allow for the possibility of ignoring the differences between the two figures as due to an occlusion, and focusing instead on the similarities.

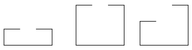

Figure 6: These figures contain gaps to indicate the location of hypothesized occlusions. The relative angle and distance between the start and end of the gap may influence the similarity of the shapes.

constraints 6 and 7 at equality, while still producing a metric cost. Specifically, we may let:

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

This function also arises from the spring model, with $p = 1$. It can be interpreted as representing the deformation of a stiff material. As we have already discussed, this function satisfies all other constraints.

4 Occlusions

As figure 5 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 [11] for another approach to this problem).

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. Figure 6, 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 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 account for the length of contour that we do not match. We suggest adding a penalty for occlusions proportional to the length of the occluded contours. This discourages occlusions that jump long, fairly well-matched contour segments.

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 (ie. 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.

			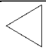		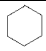	
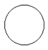	eq (16)	real data	10.52	9.15	7.21	5.41
		synthetic	11.71	11.47	11.01	10.60
	eq (17)	real data	11.74	9.93	8.67	6.63
		synthetic	12.20	12.10	11.89	11.68
	eq (18)	real data	15.35	17.30	14.37	14.30
		synthetic	16.26	17.13	18.33	19.13

Table 1: Experiments with regular polygons.

5 Algorithm and Experiments

We now describe an algorithm that implements the cost functions that we have discussed, along with experiments. The curvature at each point is computed as the difference between its tangent and the tangent at the preceding point. We compute the tangent at a point as a weighted average of the (in our experiments 7) unit vectors in the neighborhood of the point, where the weight decreases with the distance from the point.

Given two contours, we use dynamic programming to find the match that minimizes the cost function. Dynamic programming has been frequently applied to this problem, and its use is described in detail in [3], or in [24], for example. To speed up the experiments we match the first points of each contour to each other. Alternately, for closed contours we could consider all possible initial matches.

We used the cost functions:

$$\frac{|k_2 t' - k_1|^2}{|k_2 t'| + |k_1|} + \lambda \frac{(t'-1)^2}{t'+1} \quad \text{Spring Model} \qquad (16)$$

$$|k_2 t' - k_1| + \lambda\, |t' - 1| \quad \text{Linear Model} \qquad (17)$$

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

$$\text{Continuous Deformation Model} \qquad (18)$$

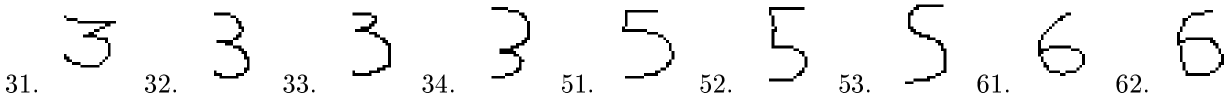

Figure 7: The numerals used in experiments.

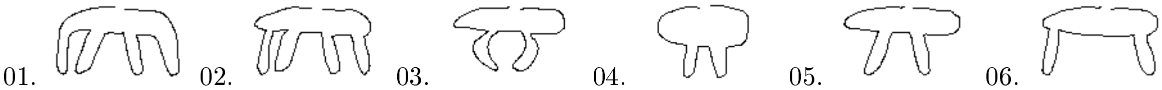

Figure 8: Six table-like shapes used in experiments.

			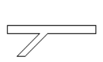	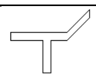
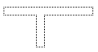	eq (16)	real data	1.68	2.31
		synthetic	0.84	1.99
	eq (17)	real data	3.78	3.41
		synthetic	3.14	3.14
	eq (18)	real data	6.03	6.02
		synthetic	1.44	2.61

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

	3	3	3	3	5	5	5	6	6
3	0	4.0	3.9	5.1	11.9	11.2	12.3	18.5	18.6
3	-	0	2.4	2.8	11.2	10.3	11.6	18.0	18.2
3	-	-	0	3.9	10.3	11.1	11.0	18.2	18.3
3	-	-	-	0	11.1	9.9	11.7	15.6	16.5
5	-	-	-	-	0	2.2	2.4	9.8	9.7
5	-	-	-	-	-	0	3.1	9.4	10.1
5	-	-	-	-	-	-	0	9.5	10.3
6	-	-	-	-	-	-	-	0	3.0
6	-	-	-	-	-	-	-	-	0

Table 3: Results using equation (16).

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. So, although λ is a free parameter, it is set to a comparable value for each function. In function 18 we set $\mu = 1$.

Table 1 shows the cost of comparing a circle to regular polygons with the same perimeter length. "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 18 prefers the polygons with fewer sides. This demon-

	3	3	3	3	5	5	5	6	6
3	0	5.4	5.1	7.0	12.1	12.9	11.9	18.4	18.9
3	-	0	4.5	5.0	12.1	12.4	11.7	18.2	18.7
3	-	-	0	6.1	11.7	12.7	11.6	17.9	18.6
3	-	-	-	0	11.5	11.5	11.6	16.4	17.0
5	-	-	-	-	0	3.3	4.1	10.8	11.0
5	-	-	-	-	-	0	4.3	11.0	11.2
5	-	-	-	-	-	-	0	11.3	11.4
6	-	-	-	-	-	-	-	0	5.6
6	-	-	-	-	-	-	-	-	0

Table 4: Results using equation (17).

	3	3	3	3	5	5	5	6	6
3	0	9.8	10.1	11.1	11.2	11.1	13.2	13.2	12.3
3	-	0	7.0	7.0	10.1	10.3	10.1	12.3	12.7
3	-	-	0	8.5	8.2	9.3	9.5	12.7	12.0
3	-	-	-	0	9.7	8.3	8.5	11.5	11.6
5	-	-	-	-	0	5.3	8.1	11.9	10.0
5	-	-	-	-	-	0	7.4	10.9	9.3
5	-	-	-	-	-	-	0	10.2	10.7
6	-	-	-	-	-	-	-	0	9.2
6	-	-	-	-	-	-	-	-	0

Table 5: Results using equation (18).

	01	02	03	04	05	06
01	0	11.61	13.23	12.13	10.11	15.48
02	11.61	0	18.21	16.90	15.84	18.54
03	13.23	18.21	0	9.33	8.23	11.99
04	12.13	16.90	9.33	0	6.16	9.31
05	10.11	15.84	8.23	6.16	0	9.50
06	15.48	18.54	11.99	9.31	9.50	0

Table 6: Results for table-like shapes in figure 8 using the spring model.

	01	02	03	04	05	06
01	0	18.78	19.37	17.81	15.89	21.48
02	18.78	0	24.99	23.79	22.12	25.44
03	19.37	24.99	0	14.62	13.39	16.75
04	17.81	23.79	14.62	0	10.09	12.74
05	15.89	22.12	13.39	10.09	0	13.85
06	21.48	25.44	16.75	12.74	13.85	0

Table 7: Results for table-like shapes in figure 8 using the linear model.

	01	02	03	04	05	06
01	0	27.08	25.66	21.38	20.64	23.83
02	27.08	0	29.85	26.85	25.51	27.74
03	25.66	29.85	0	20.66	17.79	24.33
04	21.38	26.85	20.66	0	17.04	18.27
05	20.64	25.51	17.79	17.04	0	20.21
06	23.83	27.74	24.33	18.27	20.21	0

Table 8: Results for table-like shapes in figure 8 using the continuous deformation model.

strates the significance of constraint 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.

Figure 5 shows several 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 to 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 4.

Tables 3, 4, and 5 show the cost of comparing all pairs of characters, for each function. Note that the spring and linear models find a lower cost in matching all instances of the same character than they do for any pair of different characters. The continuous deformation cost function finds some "3"s more similar to some "5"s than to some of the other "3"s, again apparently due to insufficiently weighting a single, big difference in curvatures. None-the-less, even this cost function would perfectly classify each character, if it were compared to all others, and classified according to the nearest neighbor.

Finally, Figure 8 shows six table-like shapes, and Tables 6, 7, and 8 show the results of each function when comparing these shapes. It is interesting to note that each 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. Overall, we see that the functions, especially the one based on the spring model, perform well in a wide variety of situations.

6 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 symmetry. Nor have we considered other contour properties, such as contour texture. In this section, we raise three other 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 Figure 9. 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, 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, another explanation for this phenomena can be found by interpreting each contour as the boundary of a 2-D material. Although the contours are equally distorted in the two figures, the internal 2-D 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 2-D regions bounded by the contours that we compare (however, see [3] for a counterexample).

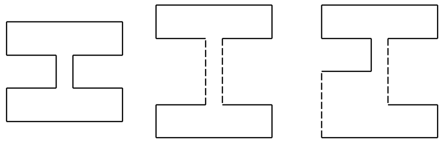

Figure 9: The figure on the left is distorted into the other two figures by stretching each of the dashed lines by a factor of 2.

Figure 10: Two figures that appear similar, although their contours, regions and part structure seem quite different.

Second, Figure 10 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 Figure 11 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 displaced.

7 Conclusions

We have considered the problem of capturing human intuitions about shape similarity in 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

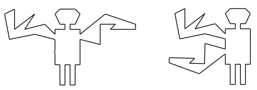

Figure 11: A figure, with one limb moved to a new position.

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 function.

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.

References

[1] Arkin. E., P. Chew, D. Huttenlocher, K. Kedem, and J. Mitchell, 1991, "An Efficiently Computable Metric for Comparing Polygonal Shapes," *IEEE Trans. on PAMI*, **13** (3):209–216.

[2] Bajcsy, R., and Solina, F, 1987. Three dimensional object representation revisited. *ICCV*:231–240.

[3] Basri, R., Costa, L., Geiger, D., and Jacobs, D.W., 1995. "Determining the Similarity of Deformable Shapes," *Technical Report*, to appear.

[4] Baumberg, A. and D. Hogg, 1994 "Learning Flexible Models from Image Sequences," *ECCV*:299–308.

[5] Biederman, I., 1985, "Human Image Understanding: Recent Research and a Theory," *CVGIP*, **32**:29–73.

[6] Binford, T.O., 1971, Visual perception by computer. *IEEE Conf. on Systems and Control.*

[7] Brooks, R., 1981, "Symbolic Reasoning Among 3-D Models and 2-D Images," *AI*, **17**:285–348.

[8] Cohen, I., N. Ayache, and P. Sulger, 1992, "Tracking Points on Deformable Objects Using Curvature Information," *Eur. Conf. on Comp. Vis.*:458–466.

[9] Connell, J.H. and M. Brady. "Generating and generalizing models of visual objects," *AI*, **31**:159–183.

[10] Duda, R.O. and Hart, P.E., 1973. *Pattern classification and scene analysis*, John Wiley and Sons, Inc.

[11] Gorman, J., R. Mitchell, and F. Kuhl, 1988, "Partial Shape Recognition Using Dynamic Programming," *IEEE Trans. on PAMI*, **10**(2):257–266.

[12] Grimson, W., 1989, "On the Recognition of Parameterized 2D Objects," *IJCV* **2**(4):353-372.

[13] Hel-Or, Y. and M. Werman, 1994, "Constraint-Fusion for Interpretation of Articulated Objects," *CVPR*:39–45.

[14] Hildreth, E., 1983, *The Measurement of Visual Motion*, MIT Press, Cambridge.

[15] Hinton, G., C. Williams, and M. Revow, "Adaptive Elastic Models for Hand-Printed Character Recognition," *NIPS-4*:512–519.

[16] Ho, S., 1987. "Representing and using functional definitions for visual recognition," *Ph.D. Thesis, Univ. of Wisconsin, Madison.*

[17] Hoffman, D. and Richards, W., 1984, "Parts of Recognition," in S. Pinker (Ed.) *Visual Cognition*, MIT Press, Cambridge.

[18] Huttenlocher, D., J. Noh, and W. Rucklidge, 1993, "Tracking Non-Rigid Objects in Complex Scenes," *ICCV*:93–101.

[19] Kass, M., A. Witkin, and D. Terzopoulos, 1988, "Snakes: Active Contour Models," *IJCV* **1**(4):321-331.

[20] Kupeev, K. and H. Wolfson, 1994, "On Shape Similarity," *ICPR*:227–237.

[21] Lai, K. and R. Chin, 1994, "Deformable contours: Modeling and Extraction," *CVPR*:601–608.

[22] Lanitits, A., C. Taylor, and T. Cootes, 1993, "A Generic System for Classifying Variable Objects Using Flexible Template Matching," *Brit. Mach. Vis. Conf.*:329–338.

[23] Marr, D. and Nishihara, H., 1978, "Representation and Recognition of the Spatial Organization of Three Dimensional Structure," *Proc. of the Royal Society of London B*, **200**:269-294.

[24] McConnell, R. R. Kwok, J. Curlander, W. Kober, S. Pang, "Ψ-S Correlation and Dynamic Time Warping: Two Methods for Tracking Ice Floes in SAR Images," *IEEE Trans. on Geo. and Rem. Sensing*, **29** (6):1004–1012.

[25] Pentland, A., 1987, "Recognition by Parts." *ICCV*:612-620.

[26] Pentland, A. and S. Sclaroff, 1991, "Closed-Form Solutions for Physically Based Shape Modeling and Recognition," *IEEE Trans. on PAMI*, **13**(7):715-729.

[27] Pope, A. and D. Lowe, 1993, "Learning Object Recognition Models from Images," *ICCV*:296–301.

[28] Rivlin, E., S. Dickenson, and A. Rosenfeld, 1994, "Recognition by Functional Parts", *CVPR*:267–275.

[29] Stark, L., and K. Bowyer, 1991, "Achieving generalized object recognition through reasoning about association of function to structure," *IEEE Trans. on PAMI* **13**(10):992–1006.

[30] Tappert, C., 1982, "Cursive Script Recognition by Elastic Matching," *IBM J. of Res. Develop.* **26**(6):765–771.

[31] Terzopoulos, D. and D. Metaxas, 1991, "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics," *IEEE Trans. on PAMI* **13**(7):703–714.

[32] Tsai, W. and S. Yu, 1985, "Attributed String Matching with Merging for Shape Recognition," *IEEE Trans. on PAMI*, **7**:453–462.

[33] Tsukumo, J., 1992, "Handprinted Kanji Character Recognition based on Flexible Template Matching," *ICPR*:483–486.

[34] Tversky, A, 1977, "Features of similarity," *Psychological Review*, **84**(4): 327–352.

[35] Ullman, S., 1989, "Aligning Pictorial Descriptions: An Approach to Object Recognition," *Cognition* **32**(3):193-254.

[36] Weiss, I., 1988, "3D Shape Representation by Contours," *CVGIP* **41**(1):80-100.

[37] Winston, P.H., T.O. Binford, B. Katz, and M. Lowry, 1984, "Learning physical description from functional definitions, examples and precedents," *MIT, AI Memo 679.*

[38] Yoshida, K. and H. Sakoe, 1982, "Online Handwritten Character Recognition for a Personal Computer System," *IEEE Trans. on Consumer Electronics* **CE-28**(3):202–209.

[39] Yuille, A., D. Cohen, and P. Hallinan, 1989, "Feature Extraction from Faces using Deformable Templates," *CVPR*:104–109.

[40] Zhu, S. and A. Yuille, 1994, "FORMS: A Flexible Object Recognition and Modelling System," *Harvard, TR 94-1*.