Alignment-Based Recognition of Shape Outlines

Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia

Brown University, Providence RI, USA

Abstract. We present a 2D shape recognition and classification method based on matching shape outlines. The correspondence between outlines (curves) is based on a notion of an *alignment curve* and on a measure of similarity between the intrinsic properties of the curve, namely, length and curvature, and is found by an efficient dynamic-programming method. The correspondence is used to find a similarity measure which is used in a recognition system. We explore the strengths and weaknesses of the outline-based representation by examining the effectiveness of the recognition system on a variety of examples.

1 Introduction

The representation of the shape of objects can have a significant impact on the effectiveness of a recognition strategy. Shapes have been represented as curves [11, 21,2,6,22], point sets [1,15,20], feature sets [3,7], and by medial axis [23,17,18,14, 12,10,9], among others. This paper develops an approach to object recognition based on a curve-based representation of shape outline using the proposed concept of an *alignment curve*, and identifies the strengths and weaknesses of using curves to represent shapes for object recognition and for indexing into image databases by shape context.

In many object recognition and content-based image indexing applications, the object outlines are represented as curves and matched. The matching relies on either aligning feature points using an optimal similarity transformation [1, 15,20] or on a deformation-based approach to aligning the properties of the two curves [11,21,2,6,22]. *Transformation-based* methods rely on matching feature points by finding the optimal rotation, translation, and scaling parameters [1, 15,20]. *Deformation-based* methods typically involve finding a mapping from one curve to the other that minimizes an "elastic" performance functional, which penalizes "stretching" and "bending" [4,19,2,22]. The minimization problem in the discrete domain is transformed into one of matching shape signatures with curvature, bending angle, or orientation as attributes [5,13,6,11,21]. The curve-based methods in general typically suffer from one or more of the following drawbacks: asymmetric treatment of the two curves, sensitivity to sampling, lack of rotation and scaling invariance, and sensitivity to articulations and deformations of parts. We address some of these issues in the proposed method.

Another type of shape representation models the shape outline as point sets and matches the points using an assignment algorithm. Gold *et al.* [7] use graduated assignment to match image boundary features. In a recent approach, Belongie *et al.* [3] use the Hungarian method to match unordered boundary points,

C. Arcelli et al. (Eds.): IWVF4, LNCS 2059, pp. 606–618, 2001.

using a coarse histogram of the relative location of the remaining points as the feature. These methods have the advantage of not requiring ordered boundary points, but do not necessarily preserve the coherence of shapes in matching.

Shapes have also been represented by medial axis or its variants and then matched. Shock graph matching have been used in [17,18,14] for object recognition and image indexing tasks. Zhu and Yuille [23] have proposed a framework (FORMS) for matching animate shapes by comparing their skeletal graphs. These approaches do not explicitly model the instabilities of the symmetry-based representations, which can be problematic when dealing with visual transformations like occlusion, view-point variation, and articulation. Liu and Geiger [12] use the A* algorithm to match shape axis trees. Their algorithm does not preserve ordering of edges at nodes which can result in matches that do not preserve coherence of the shapes. Klein *et al.* [10,9] have recently proposed an edit-distance based approach to shape matching, which is very effective, but like other graph matching techniques can in general be computationally intensive. This gives rise to the question whether the additional effort required in skeletal matching is justified by the improvements in recognition rates for particular applications. A goal of this paper is to examine the effectiveness of outline-based matching techniques in general.

In this paper, we present an outline-based recognition method, which relies on finding the optimal correspondence between 2D outlines (curves) by comparing their *intrinsic* properties, namely, length and curvature. The basic premise of the approach is that the goodness of the optimal correspondence can be expressed as the sum of the goodness of matching subsegments. This allows us to cast the problem of finding the optimal correspondence as an energy minimization problem, which is solved by an efficient dynamic-programming algorithm. We introduce the notion of an *alignment curve* to ensure a symmetric treatment of the two curves being matched. The problem formulation and the mathematics underlying the matching process is described in Section 2. In Section 3 we discuss the proposed curve matching framework in application to shape classification and handwritten character recognition. In Section 4, we discuss some of the shortcomings and limitations of curve-based representation for recognition.

2 Curve Matching

This section first discusses the problem of matching and aligning two curve segments followed by a discussion pertaining to closed curves. Denote the curve segments to be matched by $\mathcal{C}(s) = (x(s), y(s))$, $s \in [0, L]$ and $\bar{\mathcal{C}}(\bar{s}) = (\bar{x}(\bar{s}), \bar{y}(\bar{s}))$, $\bar{s} \in [0, \bar{L}]$, where s is arc length, x and y are coordinates of each point, L is length, and where each is similarly defined for $\bar{\mathcal{C}}$. A central premise of this approach is that the "goodness" of the overall optimal match is the sum of "goodness" of the optimal matches between two corresponding subsegments. This allows an energy functional to convey the goodness of a match as a function of the correspondence or alignment of the two curves as proposed earlier in [4,22]. Let a mapping $g : [0, L] \to [0, \bar{L}]$, $g(s) = \bar{s}$, represent an alignment of the two curves.

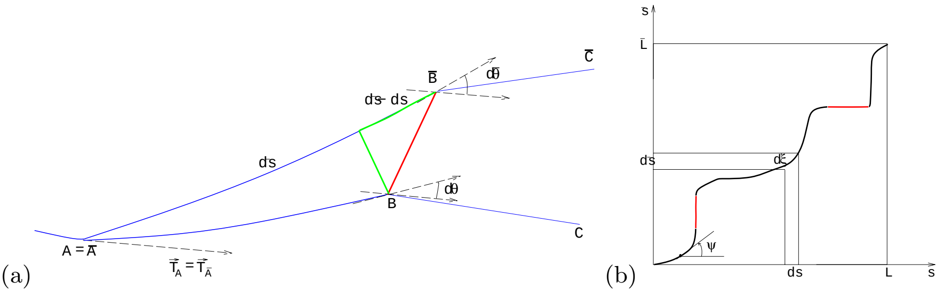

Fig. 1. (a) The cost of deforming an infinitesimal segment AB to segment $\bar{A}\bar{B}$, when the initial points and the initial tangents are aligned ($A = \bar{A}$, $\boldsymbol{T}_A = \boldsymbol{T}_{\bar{A}}$), is related to the distance $B\bar{B}$, and is defined by $|d\bar{s} - ds| + R|d\bar{\theta} - d\theta|$. (b) The alignment curve allows for a finite segment from one curve to be aligned with a single point on one curve, thus allowing for the curve segment deletion or addition.

Cohen *et al.* [4] use "bending" and "stretching" energies in a physical analogy similar to the one used in formulating active contours [8] in the form of

$$\mu[g] = \int_{\mathcal{C}} \left|\frac{\partial}{\partial s}(\bar{\mathcal{C}}(\bar{s}) - \mathcal{C}(s))\right|^2 ds + R \int_{\mathcal{C}} (\kappa_{\mathcal{C}}(s) - \kappa_{\bar{\mathcal{C}}}(\bar{s}))^2 ds,$$

where κ is the curvature, R is a parameter, and $\bar{s} = g(s)$. Younes [22] uses a similar functional. A key drawback of these approaches for recognition is that they are not invariant to the rotation of one curve with respect to the other, as the cost functional is a function of the absolute orientation of the curves. In addition, the issue of invariance to sampling has not been addressed. We now formulate the problem in an intrinsic manner which addresses both issues:
Definition: Let $\mathcal{C}\big|_{[s_1,s_2]}$ denote the portion of the curve from s_1 to s_2 and $g\big|_{([s_1,s_2],[\bar{s}_1,\bar{s}_2])}$ the restriction of the mapping g to $[s_1, s_2]$, where $\bar{s}_1 = g(s_1)$ and $\bar{s}_2 = g(s_2)$. Define a measure μ on this alignment function,

$$\mu[g]\big|_{([s_1,s_2],[\bar{s}_1,\bar{s}_2])} : g\big|_{([s_1,s_2],[\bar{s}_1,\bar{s}_2])} \rightarrow \mathbf{R}^+,$$

constructed such that it is inversely proportional to the goodness of the match, *i.e.*, it denotes the cost of deforming $\mathcal{C}\big|_{[s_1,s_2]}$ to $\bar{\mathcal{C}}\big|_{[\bar{s}_1,\bar{s}_2]}$.

We restrict this measure μ to one which satisfies an *additivity property*, *i.e.*, $\mu[g]\big|_{([s_1,s_3],[\bar{s}_1,\bar{s}_3])} = \mu[g]\big|_{([s_1,s_2],[\bar{s}_1,\bar{s}_2])} + \mu[g]\big|_{([s_2,s_3],[\bar{s}_2,\bar{s}_3])}$, where $\bar{s}_i = g(s_i)$. This property implies that the match process can be decomposed into a number of smaller matches, which in turn implies that it can be written as a functional $\mu[g]\big|_{([0,L],[0,\bar{L}])} = \int_0^L \mu[g]\big|_{([s,s+ds][g(s),g(s+ds)])} ds$. Then, the optimal match is given by $g^* = \operatorname*{argmin}_g \mu[g]\big|_{([0,L],[0,\bar{L}])}$.

Consider two infinitesimal curve segments $\mathcal{C}\big|_{[A,B]}$ and $\bar{\mathcal{C}}\big|_{[\bar{A},\bar{B}]}$ of lengths ds, $d\bar{s}$, and curvatures κ, $\bar{\kappa}$, respectively. In our approach we only compare the intrinsic aspects of the curves. Thus, we can align the curves such that the points A and $\bar{A}$, and their tangents $\boldsymbol{T}_A$ and $\boldsymbol{T}_{\bar{A}}$ coincide, Figure 1(a). The cost of matching the infinitesimal curve segments is the degree by which B and $\bar{B}$ and their respective tangents differ, namely,

$$\mu[g]\big|_{([s_1,s_1+ds],[\bar{s}_1,\bar{s}_1+d\bar{s}])} = |d\bar{s} - ds| + R|d\bar{\theta} - d\theta|, \tag{1}$$

where R is a constant. Then, the resulting functional is given by

$$\mu[g] = \int_{\mathcal{C}} \left[\left| \frac{d\bar{s}}{ds} - 1 \right| + R \left| \frac{d\bar{\theta}(\bar{s})}{d\bar{s}} \frac{d\bar{s}}{ds} - \frac{d\theta(s)}{ds} \right| \right] ds \tag{2}$$

The functional penalizes "stretching" and "bending". However, this formulation of the curve matching problem is inherently asymmetric. This is precisely the objection raised by Tagare *et al.* [19] to algorithms which are based on differentiable function of one curve to the other. They instead propose a "bimorphism", which diffeomorphically maps a pair of curves to be matched, and which corresponds to a closed curve in space of $\mathcal{C}_1 \times \mathcal{C}_2$. Specifically, they formulate a cost function that minimizes differences in local orientation change $|d\bar{\theta} - d\theta|$ along each differential segment of this curve, and seek a pair of functions ϕ_1 and ϕ_2, elements of the bimorphism, which optimize this cost functional.

We approach this asymmetry issue in a somewhat similar fashion. Observe that the formulation allows for mapping an entire segment of the first curve to a single point in the second curve, but it is not possible to map a single point in the first curve to a segment in the second curve. This is because the notion of an alignment is captured by a (uni-valued) function g. To alleviate this difficulty we adopt a view where an alignment between two curves is represented as a pairing of two particles, one on each curve traversing their respective paths monotonically, but with finite stops allowed. Let the alignment be specified in terms of two functions h and $\bar{h}$ relating arc length along $\mathcal{C}$ and $\bar{\mathcal{C}}$ to the newly defined curve parameter ξ, *i.e.*, $s = h(\xi)$, and $\bar{s} = \bar{h}(\xi)$. In cases where h is invertible, we have $\bar{s} = \bar{h}(h^{-1}(s)) = \bar{h} \circ h^{-1}(s)$, which allows for the use of an alignment function, $g = \bar{h} \circ h^{-1}$, as before. However, when h is not invertible, *i.e.*, when the first particle stops along the first curve for some finite time, g is not defined. While this formulation allows for a symmetric treatment of the curves, note that a superfluous degree of freedom is introduced, as in [19], because different traversals h and $\bar{h}$ may give rise to the same alignment. While Tagare *et al.* [19] treat this degree of redundancy in the optimization involving two functions, we remove this additional degree of redundancy by proposing the notion of an *alignment curve*, α, with coordinates h and $\bar{h}$

$$\alpha(\xi) \triangleq (h(\xi), \bar{h}(\xi)), \quad \xi \in [0, \tilde{L}], \quad \alpha(0) = (0,0), \quad \alpha(\tilde{L}) = (L, \bar{L}),$$

where ξ is the arc length along the alignment curve and $\tilde{L}$ is its length. The alignment curve can now be specified by a *single function*, namely, $\psi(\xi)$, $\xi \in [0, \tilde{L}]$, where ψ denotes the angle between the tangent to the curve and the x-axis, Figure 1(b). The coordinates h and $\bar{h}$ can then be obtained by integration

$$h(\xi) = \int_0^{\xi} \cos(\psi(\eta))d\eta, \qquad \bar{h}(\xi) = \int_0^{\xi} \sin(\psi(\eta))d\eta, \quad \xi \in [0, \tilde{L}].$$

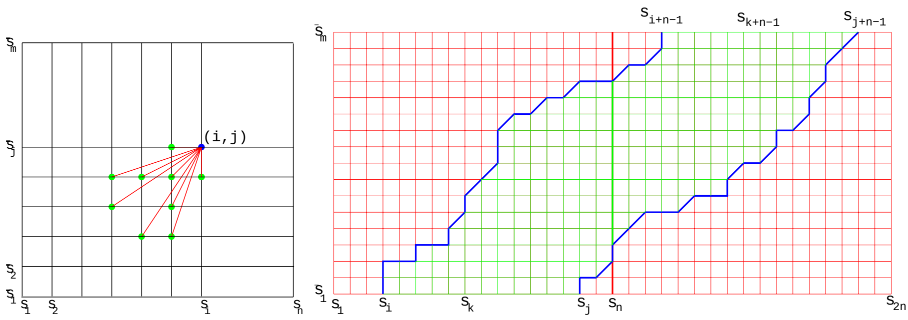

Fig. 2. (a) This figure illustrates the template that is used to in the Dynamic Programming implementation of Equation 4. The entry at (i, j) is the cost to match the curve segments $x_1, x_2, \ldots, x_i$ and $y_1, y_2, \ldots, y_j$ $d(i, j)$. To update the cost at (i, j) (blue dot) we limit the choices of the k and l, so that only costs at the a limited set of points (green dots) are considered. (b) This figure illustrates the grid used by the dynamic-programming method to compute the optimal alignment curve for closed curves. Discrete samples along the curves are the axes. The first curve $\mathcal{C}$ is repeated. If the blue curves are optimal alignment curves from $(s_i, \bar{s}_1)$ to $(s_i + n - 1, \bar{s}_m)$ and $(s_j, \bar{s}_1)$ to $(s_j + n - 1, \bar{s}_m)$, then the alignment curve from $(s_k, \bar{s}_1)$ to $(s_k + n - 1, \bar{s}_m)$ for $i < k < j$ does not cross the blue lines, so the search can be restricted to the green area. Full details are discussed in [16].

Note that ψ is constrained by monotonicity ($h' \geq 0$ and $\bar{h}' \geq 0$) to lie in $[0, \frac{\pi}{2}]$. The alignment between $\mathcal{C}$ and $\bar{\mathcal{C}}$ is then fully represented by single function ψ.

The goodness of the match corresponding to the alignment curve can now be rewritten in terms of ψ. First, if $h' \neq 0$ and $\bar{h}' \neq 0$ for $\xi \in [\xi_1, \xi_2]$, then $g = \bar{h} \circ h^{-1}$ is well defined and we rewrite $\mu[\psi]$ in terms of g using Equation 1, which after some simplification results in

$$\mu(\psi)\Big|_{[\xi_1,\xi_2]} = \int_{\xi_1}^{\xi_2} \Big[|\cos(\psi) - \sin(\psi)| + R|\kappa(h)\cos(\psi) - \bar{\kappa}(\bar{h})\sin(\psi)| \Big] d\xi \qquad (3)$$

Second, consider that one of h' or $\bar{h}'$ is zero at a point, say $h'(\xi) = 0$, implying that this point maps to a corresponding interval $[\bar{h}(\xi), \bar{h}(\xi + d\xi)]$. The cost of mapping the point $h(\xi)$ to the interval $[\bar{h}(\xi), \bar{h}(\xi + d\xi)]$ is defined by enforcing continuity of the cost with deformations: consider the cost of aligning the interval $[h(\xi), h(\xi + d\xi)]$ to the interval $[\bar{h}(\xi), \bar{h}(\xi + d\xi)]$ as the first interval shrinks to a point, *i.e.*, as $\psi \to \frac{\pi}{2}$, $\cos(\psi) \to 0$, Similarly, the case where an interval in the first curve is mapped to a point in the second curve, should be the limiting case of $\psi \to 0$ or $\sin(\psi) \to 0$. Thus, the overall cost of the alignment ψ is well defined

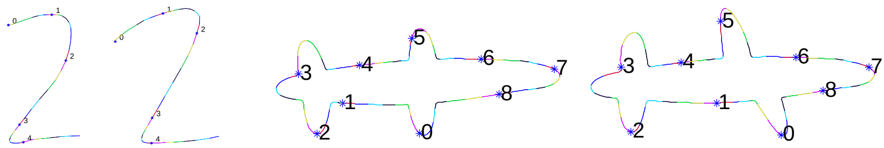

Fig. 3. Examples of the optimal alignment between curves obtained using the curve matching algorithm. The alignment is indicated by arbitrarily coloring portions of the aligned curves by identical colors with a number indicating the each portion's end point. Observe that the alignment is intuitive for both open and closed curves.

in all cases of Equation 3, and is found by minimizing

$$\begin{cases} \mu[\psi] = \int_0^{\tilde{L}} \left[|\cos(\psi) - \sin(\psi)| + R|\kappa(h)\cos(\psi) - \bar{\kappa}(\bar{h})\sin(\psi)| \right] d\xi, \\ 0 \leq \psi \leq \frac{\pi}{2}, \int_0^{\tilde{L}} \cos(\psi) d\xi = L, \text{and} \int_0^{\tilde{L}} \sin(\psi) d\xi = \bar{L}. \end{cases} \tag{4}$$

Then, the optimal alignment is given by $\psi^* = \underset{\psi}{\operatorname{argmin}}\, \mu(\psi)\big|_{[0,\tilde{L}]}$.

Definition: Let the *edit distance* between two curve segments $\mathcal{C}$ and $\bar{\mathcal{C}}$ be defined as the cost of the optimal alignment of the two curves, $d(\mathcal{C}, \bar{\mathcal{C}}) = \mu(\psi^*)$. It is straightforward to show the following [16].

Lemma 1. *If h^* and $\bar{h}^*$ specify the optimal alignment given by ψ^*, the distance function satisfies the following* suboptimal property *for $\xi_1 < \xi_2 < \xi_3, s_i = h^*(\xi_i), \bar{s}_i = \bar{h}^*(\xi_i), i = 1, 2, 3,$*

$$d(\mathcal{C}\big|_{[s_1,s_3]}, \bar{\mathcal{C}}\big|_{[\bar{s}_1,\bar{s}_3]}) = d(\mathcal{C}\big|_{[s_1,s_2]}, \bar{\mathcal{C}}\big|_{[\bar{s}_1,\bar{s}_2]}) + d(\mathcal{C}\big|_{[s_2,s_3]}, \bar{\mathcal{C}}\big|_{[\bar{s}_2,\bar{s}_3]}). \tag{5}$$

Matching Closed Curves: The edit distance between two closed curves is the minimum cost of the matching the open curve segments starting at s_1 and $\bar{s}_1$, and terminating at s_1^* and $\bar{s}_1^*$ having traversed the entire curve.

$$d(\mathcal{C}_{closed}, \bar{\mathcal{C}}_{closed}) = \min_{[s_1,\bar{s}_1]} d(\mathcal{C}\big|_{[s_1,s_1^*]}, \bar{\mathcal{C}}\big|_{[\bar{s}_1,\bar{s}_1^*]}).$$

When matching closed curves, we do not have to find the alignment for all pairs of start point correspondences. It is sufficient to choose a start point s_1 on curve $\mathcal{C}$, and the find the optimal alignments for all possible start points on the curve $\bar{\mathcal{C}}$. If we choose another point s_2, instead of s_1, we will get the same optimal alignment using Lemma 1.

The curve matching is implemented using a fast[1] dynamic-programming method, as outlined in Figure 2 and described in detail in [16]. Figure 3 illustrates the alignment for two simple cases. In all the examples, we set $R = 10$.

[1] The complexity of the algorithm to match curve segments and closed curves is $O(n^2)$ and $O(n^2 log(n))$, respectively, where n is the number of samples along the curves. It takes 0.04 secs and 1.6 secs to match curve segments and closed curves with 50 samples respectively on an SGI INDIGO^2 (195MHz).

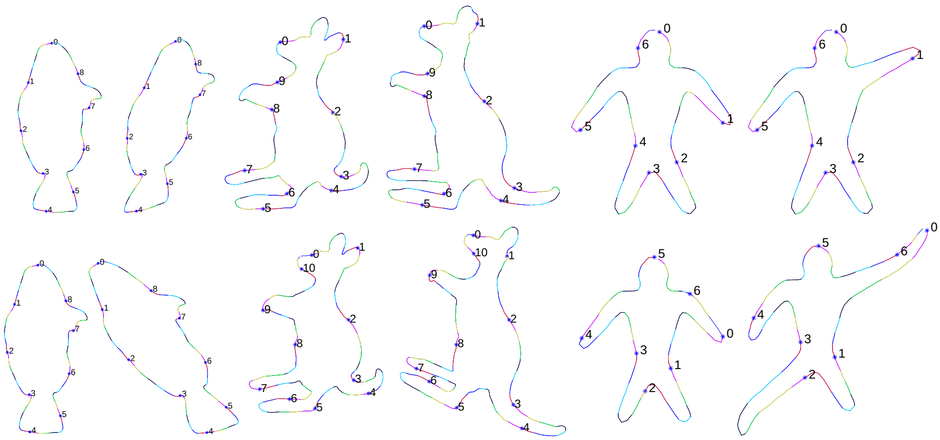

Fig. 4. This figure illustrate the performance of the curve matching algorithm in presence of an affine transformation (a), view-point variation (b) and articulation and deformation of parts (c). The alignment is indicated by arbitrarily coloring portions of the aligned curves by identical colors with a number indicating the each portion's end point. Observe that the matches are intuitive, *e.g.*, hands, legs and head of the dolls correspond in the presence of articulation, stretching and bending. Note that the different views of the kangaroo were obtained by taking snapshots of a 3D model.

We have also seen experimentally that the alignment is relatively insensitive to the choice of R.

3 Recognition Using Shape Outline Alignment

In this section, we examine the effectiveness of curve matching for recognizing shape outlines and characters. The curve alignment framework gives a correspondence between two curves, which is then used to measure the similarity between two curves. One can either use edit distance or normalized Euclidean distance between corresponding points [11,6]. For curve matching to be effective in object recognition, it has to perform well under a variety of visual transformations such as occlusion, articulation and deformation of parts, and view-point variation, which we examine now. Figure 4 shows that the curve matching algorithm works well in the presence of commonly occurring visual transformations, affine transformations, modest amounts of view-point variation, and under some articulation and deformations like stretching and bending of parts.

Object Recognition: We illustrate the use of curve matching for shape classification on a database of 36 shapes. The database consists of shapes from six different categories (fishes, tools, planes, rabbits, "greebles", and hands) with six different shapes in each category. Comparisons are made between every pair of shapes. The five nearest neighbors for each shape are highlighted. Observe that

Table 1. Costs of matching pairs of shape outlines in a database of 36 shapes, 6 samples of each of 6 categories. The five nearest neighbor are from the "correct" category 36/36, 35/36, 35/36, 33/36, 27/36 cases.

0	5	9	14	7	10	15	33	12	14	27	35	23	16	14	25	21	22	21	22	19	19	21	20	20	21	18	20	15	27	19	28	19	20	21	23
5	0	14	10	6	9	19	21	15	14	14	14	14	17	13	26	25	22	20	21	19	17	18	18	19	19	21	22	18	22	20	19	20	20	20	19
9	14	0	6	6	7	30	31	30	33	29	33	21	18	10	25	21	16	22	22	16	16	20	16	14	24	15	14	14	16	18	19	18	23	21	19
14	10	6	0	7	6	21	26	13	14	12	20	23	17	14	17	24	18	24	23	21	18	19	17	26	25	20	16	16	14	19	19	19	25	20	20
7	6	6	7	0	7	34	35	20	19	20	34	28	22	15	23	21	23	18	20	19	17	21	16	22	22	19	18	13	17	18	20	18	21	22	18
10	9	7	6	7	0	40	40	32	32	17	14	25	20	18	28	24	18	25	26	24	20	21	21	26	27	24	23	16	25	23	22	20	24	22	27
15	19	30	21	34	40	0	2	12	10	6	3	28	23	20	24	28	29	32	31	32	30	34	22	27	21	26	30	31	20	28	25	29	30	29	26
33	21	31	26	35	40	2	0	12	5	4	3	30	25	21	24	24	30	31	29	32	32	30	23	26	20	26	25	31	26	28	32	28	34	35	25
12	15	30	13	20	32	12	12	0	3	4	4	17	19	22	23	18	22	25	24	23	27	26	27	25	23	17	17	21	23	22	27	27	25	31	29
14	14	33	14	19	32	10	5	3	0	5	7	20	23	23	23	23	25	22	22	23	26	26	26	26	24	25	19	18	24	26	22	26	28	31	27
27	14	29	12	20	17	6	4	4	5	0	9	19	21	15	28	17	25	27	22	29	22	22	25	28	24	16	16	15	22	30	28	29	24	24	28
35	14	33	20	34	14	3	3	4	7	9	0	21	22	20	23	22	29	26	28	22	28	25	32	25	25	24	26	25	20	31	27	30	29	28	27
23	14	21	23	28	25	28	30	17	20	19	21	0	14	13	9	9	9	13	14	14	13	17	15	14	17	17	19	10	17	18	23	22	23	20	20
16	17	18	17	22	20	23	25	19	23	21	22	14	0	7	10	11	21	13	18	17	17	20	15	17	17	17	14	17	17	20	23	22	22	23	21
14	13	10	14	15	18	20	21	22	23	15	20	13	7	0	8	12	18	15	17	15	16	16	16	16	13	16	14	15	16	16	14	16	18	16	20
25	26	25	17	23	28	24	24	23	23	28	23	9	10	8	0	9	21	22	25	24	19	17	19	18	18	17	20	18	19	23	23	23	24	24	23
21	25	21	24	21	24	28	24	18	23	17	22	9	11	12	9	0	13	14	14	18	23	22	21	18	19	18	19	14	16	22	22	22	23	23	23
22	22	16	18	23	18	29	30	22	25	25	29	9	21	18	21	13	0	20	19	20	18	23	20	13	15	13	13	15	19	13	25	13	19	14	14
21	20	22	24	18	25	32	31	25	22	27	26	13	13	15	22	14	20	0	3	4	3	5	5	16	11	12	17	15	14	18	17	17	19	19	16
22	21	22	23	20	26	31	29	24	22	22	28	14	18	17	25	14	19	3	0	5	8	5	8	11	15	14	16	20	16	19	18	18	18	20	17
19	19	16	21	19	24	32	32	23	23	29	22	14	17	15	24	18	20	4	5	0	4	5	7	10	16	13	15	13	12	19	19	18	21	20	15
19	17	16	18	17	20	30	32	27	26	22	28	13	17	16	19	23	18	3	8	4	0	11	3	14	13	14	14	12	15	18	18	18	20	19	19
21	18	20	19	21	21	34	30	26	26	22	25	17	20	16	17	22	23	5	5	5	11	0	4	14	12	13	13	11	14	21	21	21	23	23	21
20	18	16	17	16	21	22	23	27	26	25	32	15	15	16	19	21	20	5	8	7	3	4	0	15	15	13	11	9	8	19	18	20	20	21	17
20	19	14	26	22	26	27	26	25	26	28	25	14	17	16	18	18	13	16	11	10	14	14	15	0	3	3	4	3	2	19	21	19	20	21	20
21	19	24	25	22	27	21	20	23	24	24	25	17	17	13	18	19	15	11	15	16	13	12	15	3	0	4	2	2	2	19	23	23	21	20	15
18	21	15	20	19	24	26	26	17	25	16	24	17	17	16	17	18	13	12	14	13	14	13	13	3	4	0	4	3	3	18	16	17	20	19	20
20	22	14	16	18	23	30	25	17	19	16	26	19	14	14	20	19	13	17	16	15	14	13	11	4	2	4	0	4	3	17	17	17	18	19	13
15	18	14	16	13	16	31	31	21	18	15	25	10	17	15	18	14	15	15	20	13	12	11	9	3	2	3	4	0	3	17	11	17	15	25	17
27	22	16	14	17	25	20	26	23	24	22	20	17	17	16	19	16	19	14	16	12	15	14	8	2	2	3	3	3	0	18	17	18	21	20	17
19	20	18	19	18	23	28	28	22	26	30	31	18	20	16	23	22	13	18	19	19	18	21	19	19	19	18	17	17	18	0	1	0	6	5	6
28	19	19	19	20	22	25	32	27	22	28	27	23	23	14	23	22	25	17	18	19	18	21	18	21	23	16	17	11	17	1	0	1	6	5	7
19	20	18	19	18	20	29	28	27	26	29	30	22	22	16	23	22	13	17	18	18	18	21	20	19	23	17	17	17	18	0	1	0	6	5	6
20	20	23	25	21	24	30	34	25	28	24	29	23	22	18	24	23	19	19	18	21	20	23	20	20	21	20	18	15	21	6	6	6	0	5	10
21	20	21	20	22	22	29	35	31	31	24	28	20	23	16	24	23	14	19	20	20	19	23	21	21	20	19	19	25	20	5	5	5	5	0	11
23	19	19	20	18	27	26	25	29	27	28	27	20	21	20	23	23	14	16	17	15	19	21	17	20	15	20	13	17	17	6	7	6	10	11	0

the shapes are categorized intuitively, *i.e.*, the nearest neighbors of the "tool" shapes are in the "tool" category and similarly for others.

Handwritten character recognition: As another example we have selected handwritten character recognition which due to its inherently one-dimensional nature, is well suited to this approach. As in the case of recognizing shape outlines, the optimal alignment between two characters is found and then used to compute a distance measure between the two. We have used a database of 88 digits consisting of 6 different characters, to perform recognition experiments. Matching is done between every pair of characters in the database, and the top 25 matches of a few sample characters are shown in Table 2. Observe that in

Table 2. The top 20 matches for a few handwritten digits. The database used in this experiment consists of 88 handwritten digits. The number below the matching character is the computed distance. Observe that most of the top matches of a character are samples of the same character, *i.e.*, the top matches of "2" are samples of "2".

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7	7	7	3
	4	5	8	8	8	9	9	10	11	11	11	13	14	14	14	15	15	15	15	18
3	3	3	3	3	3	3	3	3	3	3	3	3	3	7	7	7	7	3	7	7
	5	6	6	6	7	7	7	7	7	8	8	10	11	15	16	16	16	16	17	17
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	9	9	9	9	9	8
	4	4	4	4	4	5	6	6	7	7	7	8	8	9	22	23	24	25	26	26
8	8	8	8	8	8	8	8	8	8	8	8	8	6	8	6	6	6	6	6	6
	7	7	7	7	8	9	9	10	10	11	11	11	12	14	25	26	26	27	28	28

Table 3. The top five matches for a few sample characters are shown. The number below each match is the computed distance measure between the two characters.

0	0	0	U	6	G
	4	7	10	13	14
1	1	1	)	(	)
	2	4	5	6	10
2	2	7	2	3	)
	4	11	11	13	15
3	3	2	2	)	)
	5	11	16	16	19
6	6	5	0	G	0
	2	10	13	14	15

L	L	1	(	1	)
	4	11	15	16	18
m	m	N	G	3	)
	5	15	17	21	22
8	8	0	0	6	U
	12	17	18	20	21
P	P	V	0	0	G
	4	17	18	18	19
S	S	)	9	5	1
	4	10	11	11	12

9	9	1	S	5	)
	6	8	9	11	15
G	G	6	0	0	U
	5	12	19	19	19
V	V	U	0	0	P
	5	8	11	12	13
W	W	V	N	m	Z
	6	16	17	18	21
Z	Z	2	7	)	)
	3	7	10	13	14

most cases, the top matches are other samples of the same character, indicating the potential of this approach in handwritten character recognition.

Prototype formation: Prototypes have been used to improve the efficiency of object and character recognition [5] and indexing into image databases. Typically, a representative sample is used as the prototype. Instead, an "average" curve can be used. The curve alignment framework allows us to generate the average of set of curves [16]. Figure 5 shows the average curve for a set of fish outlines, and handwritten digits. The average outline of handwritten characters are used in the handwritten character recognition experiments with excellent results. For 327 digits and alphabets (34 categories) written by one subject, 323 characters (98.8%) were correctly recognized. The top five matches for a few sample characters are shown in Table 3.

Morphing: Morphing a shape to another has a variety of applications in computer graphics and animation. The proposed curve matching framework can be used to generate a sequence of images morphing a shape to another when the shapes are not very dissimilar. Figure 5 shows the morphing of the outline of a cat to that of a kangaroo. Curve matching has also been used in a variety of other applications including tracking objects in a video sequence, comparing

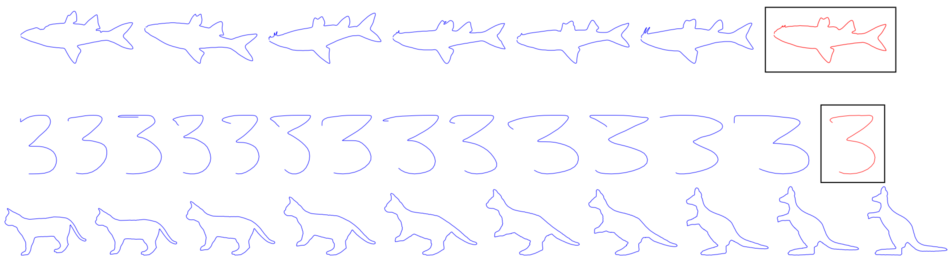

Fig. 5. Top and middle rows: A collection of curves (blue) and their average (red). Bottom row: Sequence of deforming the outline of a cat to that of a kangaroo.

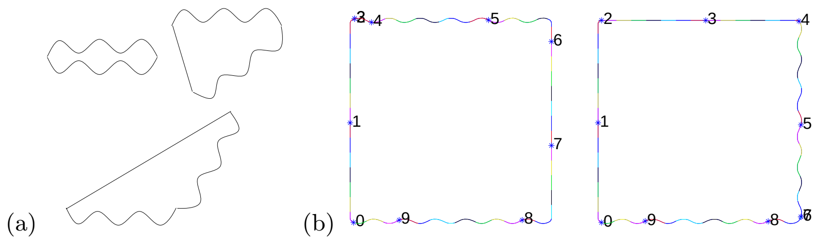

Fig. 6. (a) Curves do not represent the interior of the shape, and hence cannot adequately distinguish between perceptually distinct shapes whose outlines have similar features. (b) This also implies that in relating two shapes by curve matching, outline features take precedence over matching the shape interior! Curve matching aligns the wavy sides of the two squares, ignoring the spatial configuration as a square.

medical structures, registering 3D volume datasets by aligning characteristic 3D space curves like ridges. Thus, our proposed scheme can be useful in numerous applications.

4 Discussion and Conclusion

We have presented a computational framework to find the optimal correspondence between two 2D curves. The main contribution of this paper is to propose a new scheme for curve matching that is symmetric in its treatment of the two curves, is highly efficient, and works well in a variety of computer vision applications including shape classification, hand-written character recognition, prototype formation, and morphing. The optimal correspondence is computed by using the concept of an alignment curve and due to the use of intrinsic properties is invariant to rotations and translations and gives the intuitive matches in the presence of visual transformations like viewpoint variation, articulation and occlusions of limited extent.

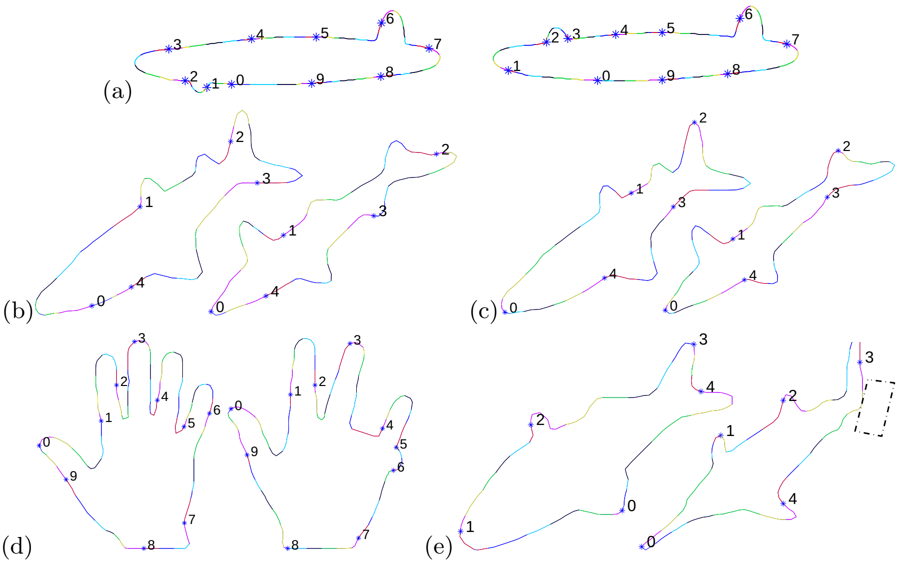

Fig. 7. This figure illustrates the sensitivity of the curve matching to spatial arrangement of parts (a, b, c, d) and to occlusion (e). Top row: We perceive two ellipses with protrusions. The larger protrusion is matched correctly, as it lies on the same side of the ellipse. However, the smaller protrusion is matched incorrectly as it lies on opposite sides of the ellipse. Middle row: The correspondence of the fishes in (b) is incorrect, as a fin on the fish on the left is matched to the head of the fish on the right. This incorrect match is because there is an extra fin in the fish on the left. (c) illustrates that the correspondence is intuitive when the fin on the "correct" side is removed. Bottom row: The missing finger and the small bump of the hand on the right causes the curve matching to give the un-intuitive match (d). (e) shows a case where curve matching gives the un-intuitive correspondence for similar shapes in presence of occlusion. Part of the tail of fish on the right (shown by the box) is occluded in this case.

We have studied the effectiveness of curve matching for shape matching and classification, especially in comparison to our group's work on shock graph-based methods [10,9], and evaluated its strengths and weaknesses. While the full comparison is beyond the scope of this paper, we summarize the main points of differences below [16]. The major advantage of curve matching is its computational efficiency. We have shown that with our proposed curve matching method we can achieve acceptable recognition rates for shape matching even under a range of visual transformations while maintaining computational efficiency. However, we have identified a number of areas where curve matching fails for 2D shape recognition. The first shortcoming of curve-based representation is that they do not represent interior of the shape. Hence, curve matching cannot easily distinguish between some perceptually distinct shapes when the local curve-based features are in conflict with the global shape percept, Figure 6. Another drawback of curve representation and hence curve matching is the sensitivity to the presence and spatial arrangement of parts. Figure 7 shows examples where curve matching gives the un-intuitive correspondence when the parts are arranged differently. Curve matching works well in the presence of occlusion, if it does not affect the overall part structure of the object. When the occlusion adds or deletes a part, curve matching can fail, as shown in Figure 7(e).

In conclusion, curve-based representation is the natural choice in handwritten character recognition and in other applications where the data is inherently one-dimensional. Also, for shape recognition, prototype formation and morphing where the variation in shape does not alter the part structure, curve matching works well. However, in the presence of large scale variations of the outline resulting in changes in the part structure, curve matching can fail, and more comprehensive representations which explicitly represent the shape interior, such as skeletal graphs are necessary despite their relatively higher computational cost.

Acknowledgements. Philip Klein acknowledges the support of NSF Grant CCR-9700146 and Benjamin Kimia acknowledges the support of NSF grants IRI-9700497 and BCS-9980091. We are grateful to Farzin Mokhtarian for providing the fish outlines, Jayashree Subrohmania for the handwritten character data, and Michael Tarr for the "greebles".

References

1. N. Ayache and O. Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *PAMI*, 8(1):44–54, January 1986.
2. R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.
3. S. Belongie and J. Malik. Matching with shape contexts. *CBAIVL*, 2000.
4. I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. *ECCV*, pages 458–466, 1992.
5. S. Connell and A. Jain. Learning prototypes for on-line handwritten digits. *ICPR*, page PRP1, August 1998.
6. Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *PAMI*, 21(12):1312–1328, December 1999.
7. S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *PAMI*, 18(4):377–388, 1996.
8. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1:321–331, 1988.
9. P. Klein, T. Sebastian, and B. Kimia. Shape matching using edit-distance: an implementation. *SODA*, pages 781–790, 2001.
10. P. Klein, S. Tirthapura, D. Sharvit, and B. Kimia. A tree-edit distance algorithm for comparing simple, closed shapes. *SODA*, pages 696–704, 2000.
11. H. Liu and M. Srinath. Partial shape classification using contour matching in distance transformation. *PAMI*, 12(11):1072–1079, November 1990.
12. T. Liu and D. Geiger. Approximate tree matching and shape similarity. *ICCV*, pages 456–462, September 1999.

13. E. Milios and E. Petrakis. Shape retrieval based on dynamic programming. *Trans. Image Proc.*, 9(1):141–146, 2000.
14. M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *PAMI*, 21(11):1105–1120, November 1999.
15. J. Schwartz and M. Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *Intl. J. Rob. Res.*, 6(2):29–44, 1987.
16. T. Sebastian, P. Klein, and B. Kimia. Curve matching using alignment curve. Technical Report LEMS 184, LEMS, Brown University, June 2000.
17. D. Sharvit, J. Chan, H. Tek, and B. B. Kimia. Symmetry-based indexing of image databases. *JVCIR*, 9(4):366–380, December 1998.
18. K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *IJCV*, 35(1):13–32, November 1999.
19. H. Tagare, D. O'Shea, and A. Rangarajan. A geometric correspondence for shape-based non-rigid correspondence. *ICCV*, pages 434–439, 1995.
20. S. Umeyama. Parameterized point pattern matching and its application to recognition of object families. *PAMI*, 15(2):136–144, February 1993.
21. H. Wolfson. On curve matching. *PAMI*, 12(5):483–489, May 1990.
22. L. Younes. Computable elastic distance between shapes. *SIAM J. Appl. Math.*, 58:565–586, 1998.
23. S. C. Zhu and A. L. Yuille. FORMS: A flexible object recognition and modeling system. *IJCV*, 20(3), 1996.