

Curve Matching for Open 2D Curves

M. Cui, J. Femiani, J.Hu, P. Wonka, A. Razdan

June 4, 2009

Abstract

We present a curve matching framework for planar open curves under similarity transform¹ based on a new scale invariant *signature*. The signature is derived from the concept of integral of unsigned curvatures. If one input curve as a whole can be aligned with some part in the second curve then the algorithm will find the requisite starting and end positions and will estimate the similarity transform in $O(N\log(N))$ time. We extend our frame work to a more general case where some part of the first input curve can be aligned with some part of the second input curve. This is a more difficult problem that we solve in $O(N^3)$ time. The contributions of the paper are the new signature as well as faster algorithms for matching open 2D curves. We present examples from diverse application set to show that our algorithm can work across several domains.

1 Introduction

Given two curves as input, we seek to find what part of the first matches the best with a part or the whole of the second curve². This type of query is useful in many applications involving shape comparison. Example applications areas are computer vision, geospatial analysis and registration of images, computer aided geometric design, manufacturing, etc.

¹Similarity transform is defined as a 2D transform that is limited to translation, rotation and uniform scaling.

²The word *match* is intuitive rather than a precise mathematical definition at this stage and we will define it later.

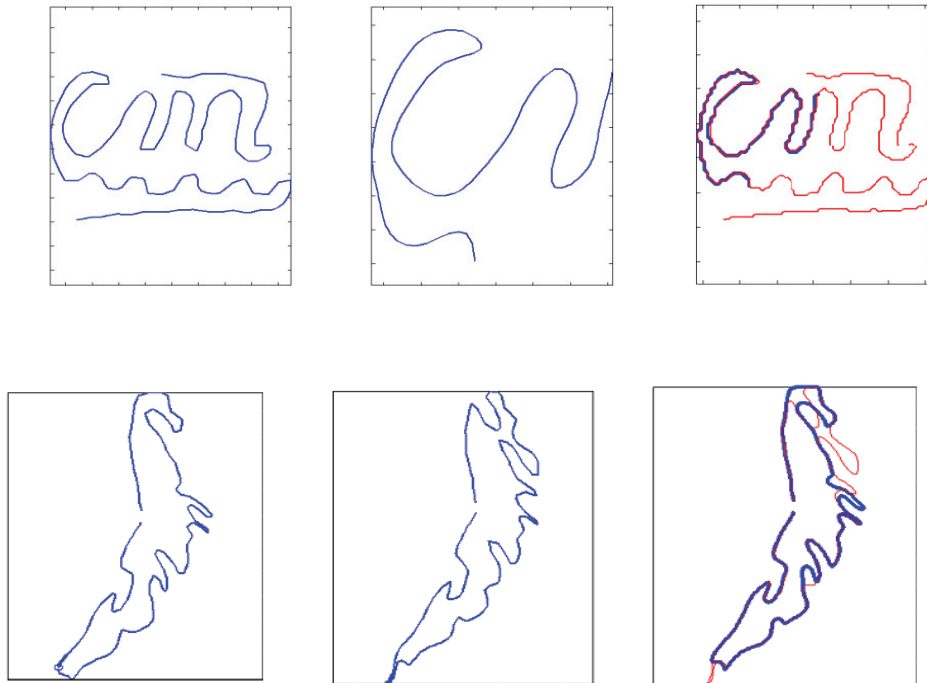


Figure 1: Examples that illustrate the applications of our algorithm. Each row is an example, the first and second column are the input curves and the last column is the aligned result of our curve matching algorithm. The first row is a sample of handwriting, the second row represents curves from aerial imagery

We want to motivate the discussion about curve matching through some examples illustrated in Figure 1, which is organized as follows.

The first and second columns represent the input curves while the last column represents the two curves aligned in the same coordinate space using the algorithms proposed by this paper.

Application 1 (first row): the first curve is a handwritten stroke and the second curve is a feature, whose presence or absence in the first curve is what we are interested in. The second curve can be completely embedded in the first. We call it a whole-to-part curve matching problem.

Application 2 (second row): the two input curves are contours extracted from aerial images. They represent the same area of interest. The goal here is to find two sub-curves that differ only by a similarity transform, which enables image registration and change detection. We call it a part-to-part curve matching problem. Note that whole-to-part is a special case of part-to-part and we will present a specialized, faster solution.

Our motivation is a well known technique that solves the part-to-part problem invariant to Euclidean transform³ [19]. It makes use of curvature with respect to arclength as a signature invariant to Euclidean transform, see Figure 3(a) for an example. However, [19] cannot be extended to curves under similarity transform: if the potentially matched parts have a scale difference, then they do not span the same length on the curvature plot, and thus cannot be compared directly. See figure 2 for an example. In contrast, our signature is scale invariant and can deal with similarity transform.

1.1 Contributions

In this paper we propose algorithms that perform whole-to-part matching in $O(N \log(N))$ and part-to-part matching in $O(N^3)$, where the input curves can differ by a similarity transform. This is a significant improvement over the state of the art methods which are $O(N^4)$. Hash-based methods [11, 20] can do the comparison in $O(N)$, but they require pre-processing steps that are of $O(N^4)$ asymptotic complexity.

³Euclidean transform only includes rotation and translation

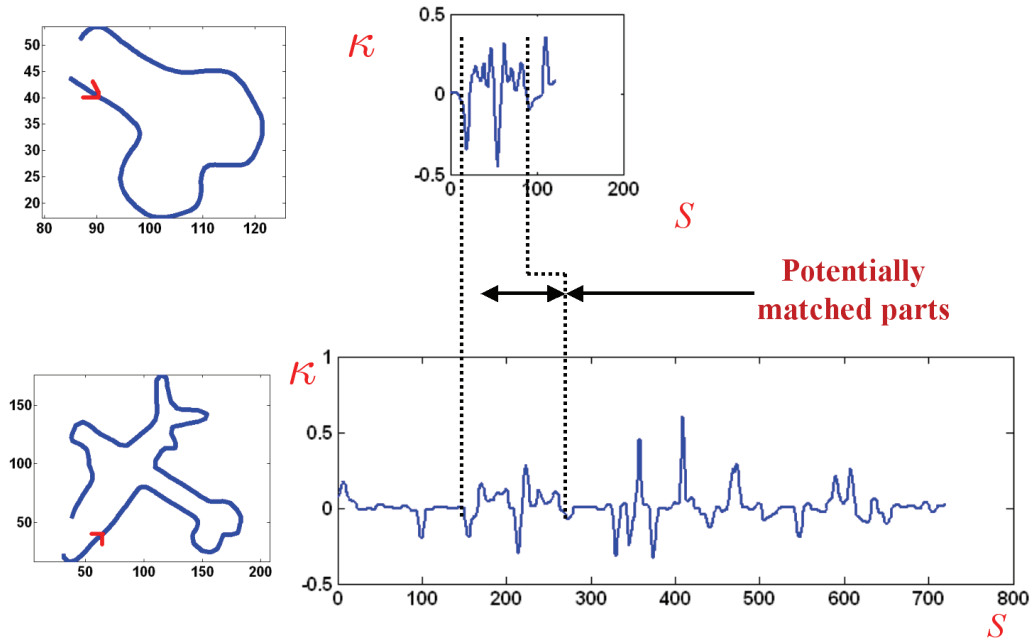


Figure 2: Two curves have potentially matched subparts, but they do not span the same length by arc length parametrization. Red arrows denote the direction of the parametrization.

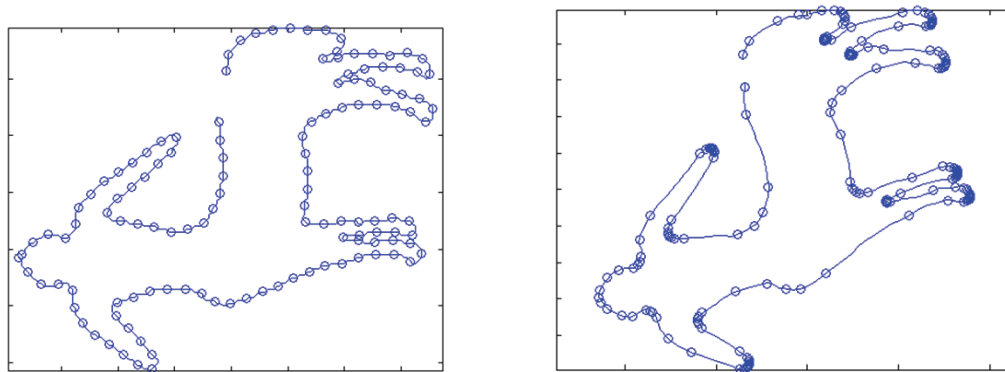


Figure 3: Left: Points sampled at equal intervals along the arc length parametrization; Right: Points sampled at equal intervals using our parametrization.

The main contributions of this paper are:

1. We outline the theoretical framework and develop a new curve *signature* based on the concept of integral of curvatures of a curve. This is also a novel way to parameterize curves that is scale invariant. We use this for solving the general problem of matching open curves.
2. We present an $O(N \log(N))$ algorithm for whole-to-part matching of a pair of open curves under similarity transform. We extend this to address part-to-part matching under similarity transform and the solution is an $O(N^3)$ algorithm.

The rest of the paper is organized as follows: Section 2 covers existing algorithms for curve matching. Section 3 describes the scale invariant signature proposed. Section 4 covers the topic of curve matching using the proposed signature and is followed by sections on experiments and discussion of results (section 5) which includes a comparison to other methods. The paper concludes with a brief summary (section 6).

2 Related Work

We focus on work that serves to place this paper relative to the state of the art. See [17] for a more general review of the area.

Methods like [1, 7, 10] belong to whole-to-whole matching, use global features and only return a distance estimation. Methods like [5, 15] belong to whole-to-whole methods, use global features and return correspondence as well as similarity. Application domains of whole-to-whole methods are quite different from ours, and therefore we do not compare our method against this category.

Methods based on Hausdorff distance [12, 21] belong to whole-to-whole matching category. These methods use point-wise features and return only a similarity estimation value. Methods such as [9, 11, 13], also belong to whole-to-whole matching using point-wise features and returns correspondence and similarity estimation value. The method in [8] deal with whole-to-part and part-to-part matching. However, in order to eliminate the influence of scale, they limit inputs to closed curves and normalize the curvature signature to a fixed interval. The method in [22] represents another class of part-to-part matching algorithm which first extracts dominant points and uses polylines to approximate the original curves.

Our proposed algorithm belongs to the same category as those of [3, 13, 18, 19]. Fast curve matching algorithms have been developed based on the idea of geometric hashing [11, 20]. The comparison of our method to these is highlighted in section 5.2.

3 Scale Invariant Signature based on Integral of Curvature

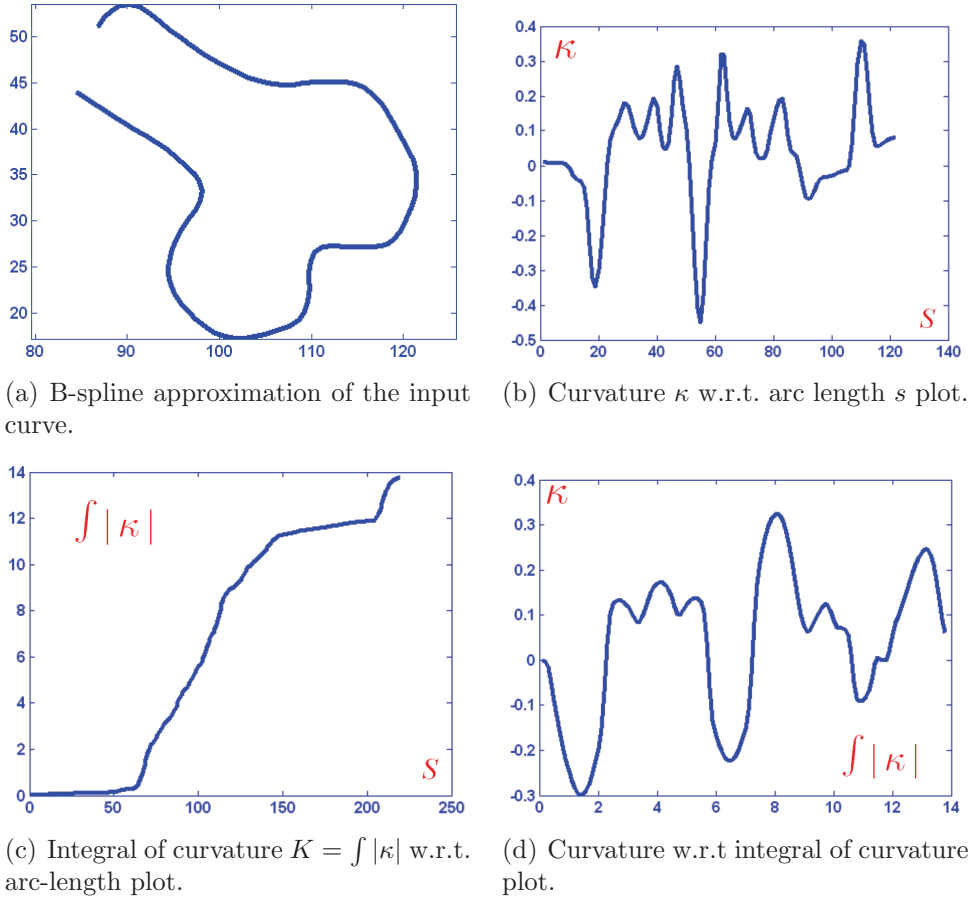


Figure 4: Overview of the signature extraction.

The input to our algorithm can be conceptually characterized as follows. First, we only deal with the problem of matching *open* 2D (planar) curves

such as in Figure 1. Second, the curves may come from any source i.e. vectorized from a digital source or a Computer Aided Design (CAD) system, as long as the curves have a parametric representation (for ease of computation), they are twice differentiable (for robust curvature estimation) and other filters or algorithms have pre-processed the input curves to remove high frequency noise. Third, the curves can come from independent coordinate systems and therefore can differ by a similarity transform (translation, rotation and uniform scaling). The output is the best match with an estimate of the similarity transform. The *best match* has different meaning for whole-to-part and part-to-part matching. The difference is explained in more detail in section 4.

Our method consists of two major parts. One is the extraction of the scale invariant *signature*, which is described below. The second part uses the signature to perform matching. This is described in section 4.1.

The steps to get the signature are listed below. These are also illustrated in Figure 4.

1. This step covers any pre-processing required to de-noise and convert the input curve to a parametric representation such that the curvature can be readily and robustly estimated for the curves. For example if the input is a digitized point set, one option is to approximate with a parametric cubic B-spline curve; see Figure 4(a)⁴.
2. Sample the curvature of the curve at equal arc length intervals. Figure 4(b) shows curvature κ w.r.t. the arc length plot.
3. Integrate the unsigned curvatures along the curve by summing the absolute values of curvatures discretely sampled along the curve. Plot the integral of the unsigned curvature $K = \int |\kappa|$ w.r.t. the arc length s ; see Figure 4(c).
4. Compute curvature (κ) of the curve at equal-interval-sampled points along the integral of unsigned curvature axis (vertical axis in previous figure 4(c)) i.e. (signed) curvature (κ) w.r.t. the integral of unsigned curvature (K) plot; see Figure 4(d). This is our signature and the core of our method. This can also be considered as a novel scale invariant parameterization of a curve.

⁴All the steps and plots outline refer to only one curve. In practice this must be computed for both the curves.

Steps 1 to 3 are straightforward, hence we will explain the development of the signature in step 4.

3.1 The Proposed Signature

The integral of the absolute curvature, is defined in the following way. The absolute curvature κ at any arc length s of an arc length parameterized curve $\mathbf{x}(s) : x(s), y(s)$ is given by:

$$|\kappa(s)| = \|\ddot{\mathbf{x}}(s)\|, \quad (1)$$

where $\ddot{\cdot}$ represents the second derivative.

The integral of the absolute curvature from point s_1 on the curve with respect to another fixed point s_2 can be defined as:

$$K(s_1 : s_2) = \int_{s_1}^{s_2} |\kappa(s)| ds \quad (2)$$

with $\kappa(s)$ defined as (1). It should be noted that the integral of curvature, taking the signed value, is also an integral invariant. We prefer to use the integral of unsigned curvature rather than the signed curvature because in curves that have many inflection points, the signed integral can be extremely small and can create numerical problems, while the unsigned integral presents a reasonable range of values to work with. Also, the integral of the signed curvature is not invertible, a property we use in our algorithm.

Now we claim $K(s_1 : s_2)$ is invariant under similarity transform, where s_1 is an arbitrary start point on the curve. We present a proof in support of the claim.

Proposition 1. *The integral of absolute curvature is invariant under similarity transform.*

Proof: Curvature and arclength of a curve are invariant to translation and rotation. For a given curve $\mathbf{x}(s)$ of arclength l , the sum of absolute curvature K is:

$$K(0 : l) = \int_0^l |\kappa(s)| ds \quad (3)$$

where $k(s)$ is curvature at a point on the curve. Scaling the curve by a factor of α , we get a new curve $\bar{\mathbf{x}}(\bar{s})$ with arclength $= \alpha l$. For the scaled curve, we

have:

$$\bar{K}(0 : \alpha l) = \int_0^{\alpha l} |\bar{\kappa}(\bar{s})| d\bar{s} \quad (4)$$

where ' - ' represents the scaled curve. since curvature changes inversely to the scale of the curve, therefore:

$$|\bar{\kappa}(\bar{s})| = \frac{1}{\alpha} |\kappa(s)| \quad (5)$$

and $d\bar{s} = \alpha ds$. Substituting in 4 we have:

$$\bar{K}(0 : \alpha l) = \int_0^{\alpha l} \frac{1}{\alpha} |\kappa(s)| d\bar{s} \quad (6)$$

$$\bar{K}(0 : \alpha l) = \int_0^l \frac{1}{\alpha} |\kappa(s)| \alpha ds = \int_0^l |\kappa(s)| ds \quad (7)$$

We next illustrate the geometric interpretation of this argument with help of Figure 5. Assume we have two curves $A(s)$ and $\bar{A}(\bar{s})$ (the curves are not shown but their arc length *vs.* integral of curvature plots are). Suppose the segments between s_1 and s_2 on A and \bar{s}_1 and \bar{s}_2 on \bar{A} potentially match each other modulo scale factor m . Then we have the following claim (the proof is an extension of Proposition 1).

Proposition 2. *For any s between (s_1, s_2) and \bar{s} between (\bar{s}_1, \bar{s}_2) , if $K(s_1 : s) = \bar{K}(\bar{s}_1 : \bar{s})$, then $(s - s_1)/(\bar{s} - \bar{s}_1) = m$*

If some parts of the two input curves are potentially identical, up to a similarity transform, and the scaling factor between them is m ; then, their span on the integral of curvature axis is the same and the ratio of their span on the arc length axis is m (see Figure 5).

It is still not straight-forward to directly find matched parts on the arc length *vs.* integral of curvature plot, since the integral of curvature value of start position s_1 and \bar{s}_1 can be any value. We eliminate the influence of the start offsets by taking the derivative of arc length s w.r.t. K . At this point we assume $K(s_0 : s)$ and $\bar{K}(\bar{s}_0 : \bar{s})$ both have an inverse, called $\Gamma(k)$ and $\bar{\Gamma}(\bar{k})$, respectively. It should be noted that if the curvature is truly zero (flat sections of the curve) then one does not have the inverse function. The correct methodology is to use pre-image rather than inverse.

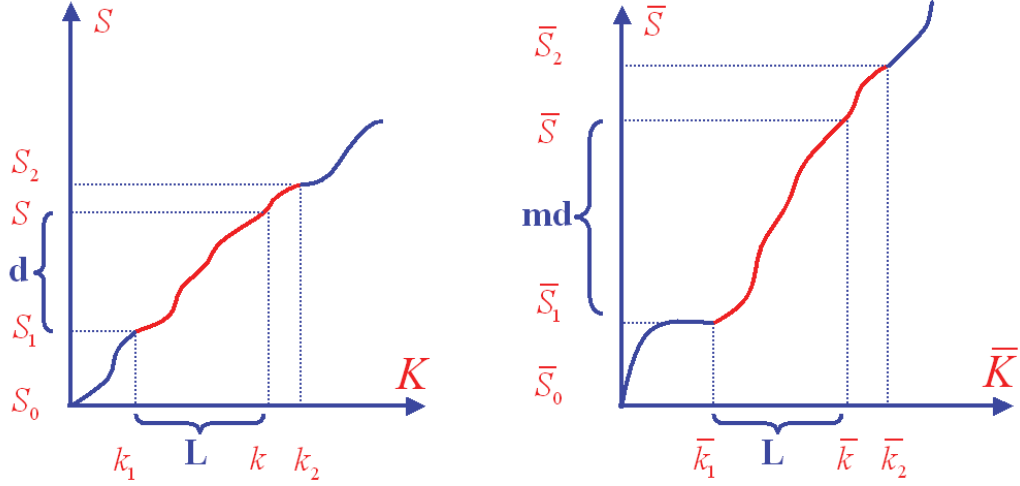


Figure 5: K^{-1} and \bar{K}^{-1} plots of two hypothetical curves that match each other modulo similarity transform. The segment between $[\bar{s}_1, \bar{s}_2]$ is a scaled version of the segment between $[s_1, s_2]$ and the scale factor is m .

Proposition 3. Assume a segment of the first curve spanned by $[k_1, k_2]$ along Γ is a match to the segment of the second curve bounded by the span $[\bar{k}_1, \bar{k}_2]$ along $\bar{\Gamma}$ modulo scale factor m , then:

- (1) $k_2 - k_1 = \bar{k}_2 - \bar{k}_1$.
- (2) For any k between $[k_1, k_2]$ and \bar{k} between $[\bar{k}_1, \bar{k}_2]$ if we have $k - k_1 = \bar{k} - \bar{k}_1$ then $\dot{\Gamma}(k) = m\dot{\bar{\Gamma}}(\bar{k})$, where $\dot{\Gamma}$ means taking the derivative with respect to $\int |\kappa|$.

Proof.

1. It follows from analysis above that if two parts match then integral of curvature from the start points should be the same.

2. Further:

$$\Gamma(k + \Delta k) - \Gamma(k_1) = m(\bar{\Gamma}(\bar{k} + \Delta k) - \bar{\Gamma}(\bar{k}_1)) \quad (8)$$

$$\Gamma(k - \Delta k) - \Gamma(k_1) = m(\bar{\Gamma}(\bar{k} - \Delta k) - \bar{\Gamma}(\bar{k}_1)) \quad (9)$$

Subtracting eq 9 from eq 8, dividing both sides of the resulting equation by $2\Delta k$ and taking the limit $\Delta k \rightarrow 0$, we get the following equation:

$$\lim_{\Delta k \rightarrow 0} \frac{\Gamma(k + \Delta k) - \Gamma(k - \Delta k)}{2\Delta k} = m \lim_{\Delta k \rightarrow 0} \frac{\bar{\Gamma}(\bar{k} + \Delta k) - \bar{\Gamma}(\bar{k} - \Delta k)}{2\Delta k} \quad (10)$$

which means $\dot{\Gamma}(k) = m\dot{\Gamma}(\bar{k})$ □

The derivative of arc length with respect to integral of curvature removes the impact of the start point for match. This is only one step away from achieving the plot in Figure.4(f). Since $\dot{\Gamma} = \frac{dS}{df|\kappa|}$, by the inverse function theorem we know $\dot{\Gamma} = \frac{1}{\frac{df|\kappa|}{dS}} = \frac{1}{|\kappa|}$ Also $\frac{1}{|\kappa|} = \rho$, where ρ is the radius of curvature. Therefore, we are in essence comparing the inverse of curvature or radius of curvature. For matched parts, the radius of curvature w.r.t. integral of curvature plots are scaled versions of each other, and the curvature w.r.t. integral of curvature plots themselves are also scaled versions of each other. Therefore, we can directly sample curvature instead of taking the derivative of arc length.

If we sample the curve based on equally spaced intervals of absolute integral of curvature, as shown in Figure. 3(b), then the potentially matched parts will span the same length even if they have a scale difference, as shown in Figure. 6. We also know that curvature is invariant to translation and rotation, and so is its integral. Now we have a parametrization domain that is truly invariant to similarity transform and it greatly facilitates our next step for matching.

4 Curve Matching

4.1 Whole-to-Part Matching

The domain of our signature is scale invariant but the range still has a scaling factor. We adopt normalized cross-correlation [23], which is not influenced by the scale, to evaluate similarity between two matched curves.

For whole-to-part match, we know one curve is potentially embedded in the other one, therefore, we only need to find the position where the first curve aligns to in the second curve. Since we are using the curvature signature parameterized by the integral of curvature interval, the only difference between the first curve and the matched part in the second curve is the scale difference; as curvature is inversely proportional to scale. This is equal to a template matching problem and normalized cross correlation can handle it regardless of the scale factor. The formulation to calculate the normalized cross correlation [6] between the two curves is:

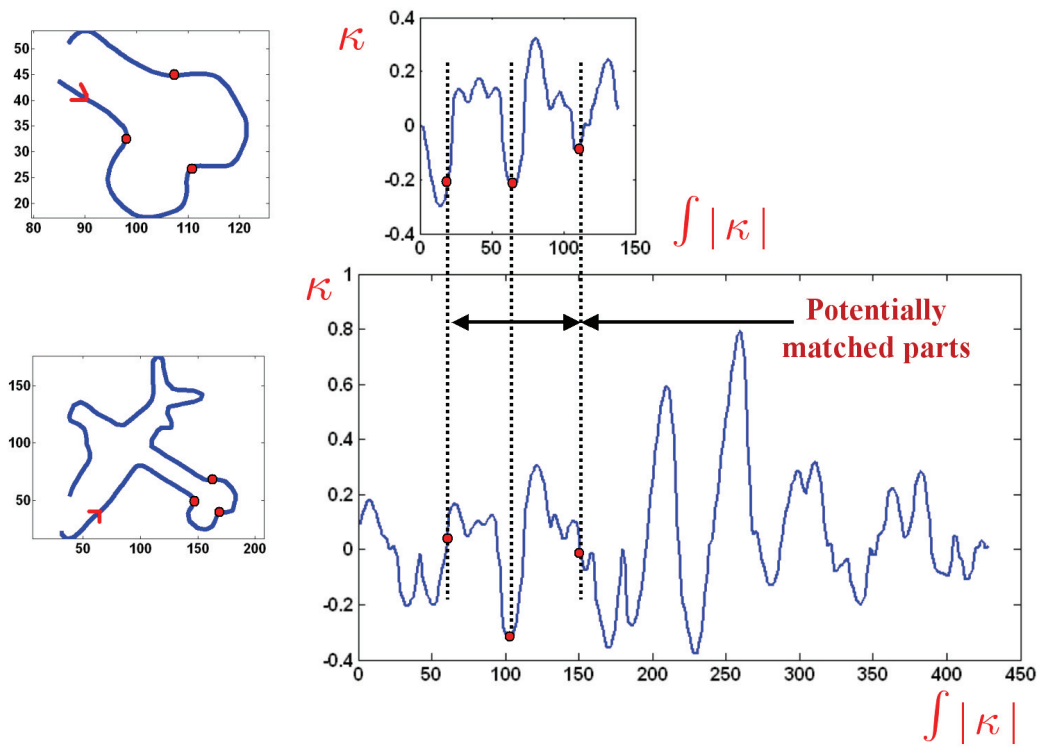


Figure 6: Two curves have potentially matched subparts, and they span the same length by our new parametrization.

$$\nu(u) = \frac{\sum_{i \in \Omega} [f(i) - \bar{f}][t(i-u) - \bar{t}]}{\sqrt{\sum_{i \in \Omega} [f(i) - \bar{f}]^2 \sum_{i \in \Omega} [t(i-u) - \bar{t}]^2}}, \quad (11)$$

where t is the first curve working as a template window sliding along the second curve f and u is the offset of one curve to the other. \bar{t} is the mean value over the whole template and \bar{f} is the mean value of the sliding window in the second curve. The range of the ν value lies in $[-1, 1]$. In our implementation, we only consider positive correlation. The larger the value $\nu(u)$, the better the match. Once we find the maximal position of u , we can decide the relative alignment of the two curves easily, as illustrated in Figure. 7. We can also set a threshold below which we claim the pattern does not exist in the target.

Since the problem can be solved with a naive algorithm of $O(N^3)$ time complexity, it is essential to analyze the complexity of our algorithm. Using the sum table techniques first proposed by Lewis in [6], it is possible to pre-calculate the denominator in equation(11) in $O(N)$ and the storage requirement is also $O(N)$. The numerator of (eq 11) can be calculated in $O(N \log(N))$. Therefore, the overall time complexity is $O(N \log(N))$. The storage requirement is $O(N)$.

It should be mentioned that [2, 14] made some further improvements to Lewis' original method. However, we find in [2], the result is only an approximation of the normalized cross correlation ν . While in [14], the improvement is specialized to a situation where two inputs have same length in each dimension (they dealt with 2D images). Therefore we use the original implementation.

4.2 Part-to-Part Matching

One may also want to ask if there is a part of one curve that matches a part of another curve. This is a more complex problem and is referred to as part-to-part curve matching problem. We can directly extend (a more optimal way is presented later) our whole-to-part algorithm to work on the part-to-part matching problem as follows:

For each start point and end point on input C_1 , run the whole-to-part algorithm on this segment on curve C_1 and whole curve C_2 . Find the best alignment position and also record the best ν score. Denote this ν score as best local score for this particular segment. Finally return the segment that

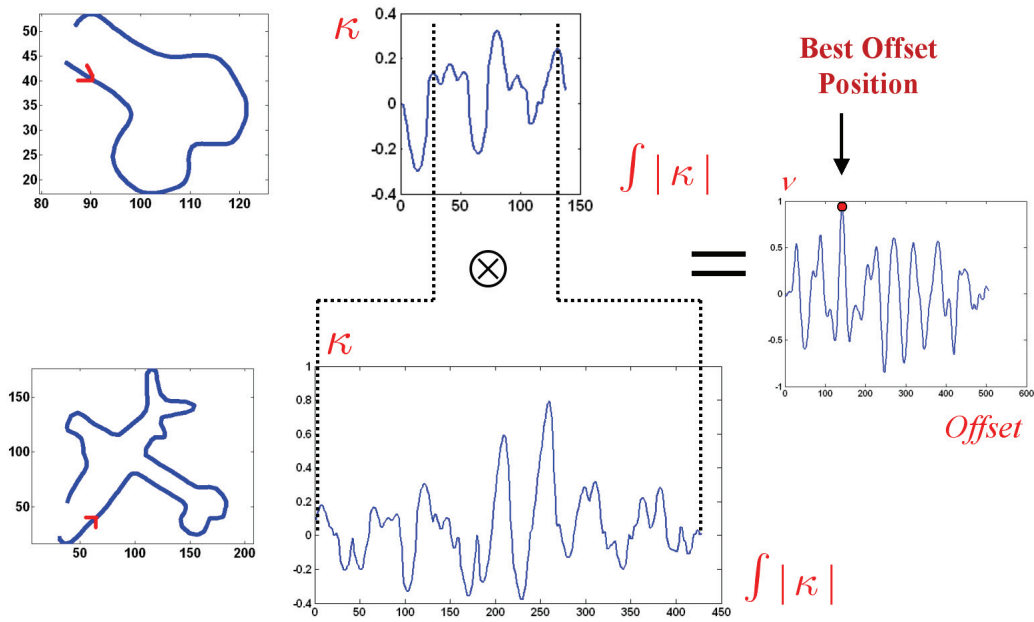


Figure 7: Two curves and their normalized cross correlation. The red spot represents alignment position.

has the best local ν score among all segments. However, two issues make the performance poor ($O(N^3 \log(N))$) in practice.

First, the similarity metric needs some modification to get useful results. In whole-to-part matching, the length of the match is fixed and all the ν scores can be fairly compared. In part-to-part matching there is a fundamental issue and that is that an infinitesimally small part of one curve will match many parts of another curve. We consider this as a degenerate case. In other words, it is very likely that a shorter match will have a higher ν score. If only ν score are used, we may end up with some perfect but trivial matches, such as a small curve segment. To favor longer matches, we use two simple strategies:

1. We use thresholding on the length of match.
2. The final score S used to compare the match is a weighted multiplication of ν score and length of the match: $S = \nu * L$, where L is the length of the match.

Second, the time complexity of the above algorithm is $O(N^3 \log N)$. By careful observation we find that in this particular case using DFT for cross correlation is not the best choice. If we perform cross correlation between whole curves, the complexity will be $O(N^2)$. During the process of calculating whole-to-whole cross correlation, the part-to-whole cross correlation can be found and stored for constant retrieval time for later use. To be specific, we store an $N \times N$ matrix P , where $P_{i,j} = a_1 * b_i + a_2 * b_{i+1} + \dots + a_j * b_{i+j-1}$. Notice that $P_{i,j} = P_{i,j-1} + a_j * b_{i+j-1}$. This confirms that the matrix P can be computed in $O(N^2)$.

In comparison to the direction extension mentioned before ($O(N^3 \log(N))$), for our part-to-part algorithm the time complexity is reduced to $O(N^3)$. The space complexity is now $O(N^2)$ in order to store the matrix P .

4.3 Transform Estimation

After start points and end points on both curves have been decided, we can estimate the similarity transform between them using a well known method due to Horn [4] Horn's method can give a closed form solution and is fast.

5 Experiments and Discussion

To test our algorithm, we have drawn some test cases from different applications domains. These are handwriting strokes, aerial images and Kimia image database [16].

We implemented a naive arclength based search algorithm to compare with ours. This algorithm is of complexity $O(N^4)$. It works directly in arclength domain instead of the integral of curvature domain. It has four loops over all possible start and end points of matched parts on two curves. Within the most inner loop it first estimates the scale based on the ratio of the lengths between start and end points on both curves. Then one of the curves is rescaled to make them the same scale. The two curves are then sampled at equal arclength intervals and at each sample point the curvature is computed. The angle between the two vectors are calculated to estimate the similarity.

We also implemented an extended version of Wolfson’s method [19]. The original method can achieve part-to-part matching under Euclidean transform in $O(N^2)$. We extended it to similarity transform by exhaustively searching over a certain range of scale with small steps. Note that for each possible scale we can find the longest match. The best scale is set to the scale that yields the longest match.

The results and comparisons are organized as follows.

5.1 Visual Quality Analysis

First, we give detailed results for two pairs of curves to highlight the matching process. We follow these with some examples where only the matched results are shown. Also, except for one set of examples in Figure 10, the rest of the results focus on part-to-part matching. Since that is a harder challenge, we have emphasized those results.

In Figures 8 and 9, we show two examples of matching results in detail.

In the two figures mentioned above, (a) shows input curves in the same coordinate system, thereby highlighting the scale difference. The matched parts found by our methods are shown in bold in the same figure. We show the correspondence of the matched parts using our method in (b). This is done by back-projecting the integral of curvature match on the original curve. Figures 8(c) and 9(c) show matched parts using the naive search algorithm briefly described above. Figures 8(d) and 9(d) show the matches obtained by

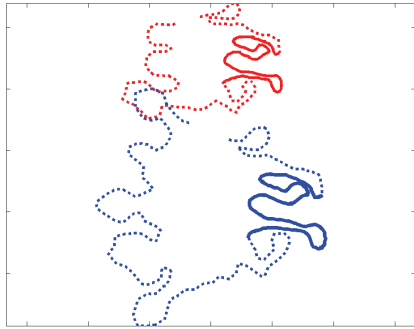
extended Wolfson’s method. We can see that among three methods the extended Wolfson’s method performs the worst. It is unable to find the ground truth scaling factor. Our inference is that this is because the method uses a pointwise difference comparison of curvature instead of cross correlation based comparison. From experiments we observed that even when we set a very low tolerance threshold, the pointwise comparison yields a much shorter match than correlation. The naive method can find the ground truth scale factor, however, its computation time is much longer.

Note that for some cases all three methods cannot find a match that is obvious to human eyes. Two reasons are: 1) the input comes from the digital binary images from Kimia database. Rescaling images causes the extracted curves to accentuate noise; 2) Since we do not know the scale beforehand, we use same sampling rate when we do numerical integration of unsigned sum curvature. Therefore, the one that is a smaller in scale has larger step size and less accuracy.

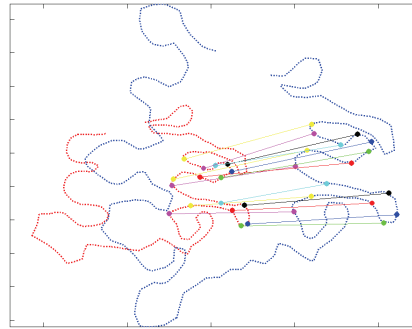
Figures 10 and 11 show results on whole-to-part and part-to-part matching respectively. The last row in figure 11 is one pair of the curves from the the Kimia image database. However, we modified the original curves to highlight our algorithm’s capabilities. We created more complex open curves by overlaying two binary images from the database and creating open curves as test cases.

5.2 Complexity Comparison

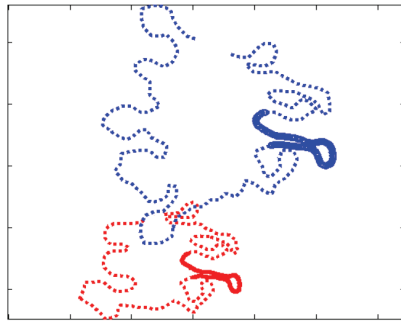
First we compare our method to others based on the asymptotic complexity analysis rather than running times. In [3] the complexity is $O(N^4)$ and the algorithm only deals with whole-to-part match. In [18] the goal is essentially the same as our algorithm in the sense that we both desire a part-to-part match. However, in [18] the algorithm does not allow a uniform scaling between the two curves, and the complexity is still $O(N^4)$. In our case we allow uniform scaling and the complexity is reduced to $O(N^3)$. Geometric hashing and its variants [11, 20] can do a whole-to-part query in $O(N)$. However, these methods requires a pre-processing step for all curves in the database. This part has complexity of $O(MN^4)$, where M denotes number of target curves in database. When geometric hashing method is extended to part-to-part matching, the part-to-part query needs $O(N^4)$ computations to find a best match. Therefore, our method has better overall performance than existing methods known to us.



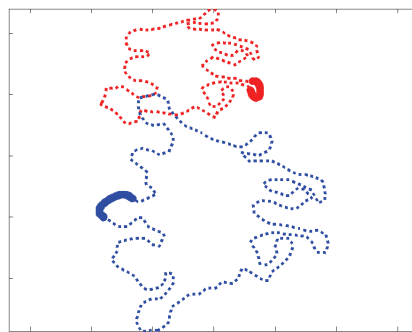
(a) Input curves in the same coordinate system. Note difference in the scale. Matched parts found by our algorithm are highlighted.



(b) Correspondence between the matched parts.

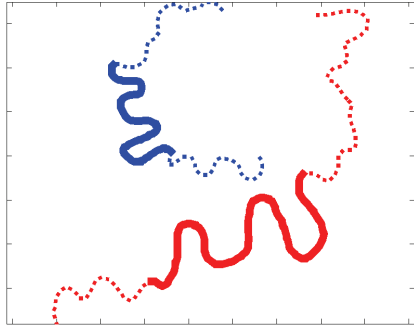


(c) Matched parts found by a naive $O(n^4)$ arclength based search.

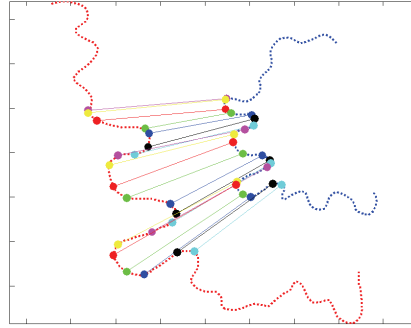


(d) Matched parts found by extended Wolfson's method

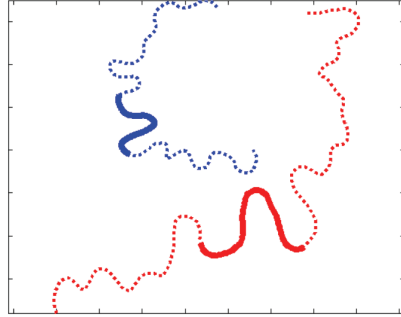
Figure 8: Example 1 shows part-to-part matching for a pair of curves. (a) is the match found using our algorithm compared to (c)(d). The proposed method outperforms arclength based methods i.e. it finds a longer match than the exhaustive method.



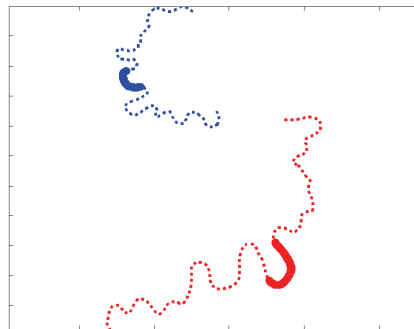
(a) Input curves in the same coordinate system. Note difference in the scale. Matched parts found by our algorithm are highlighted.



(b) Correspondence between the matched parts.



(c) Matched parts found by a naive $O(n^4)$ arclength based search.



(d) Matched parts found by extended Wolfson's method

Figure 9: Example 2 highlighting details of part-to-part matching. The proposed method finds a longer match compared to arclength based methods.

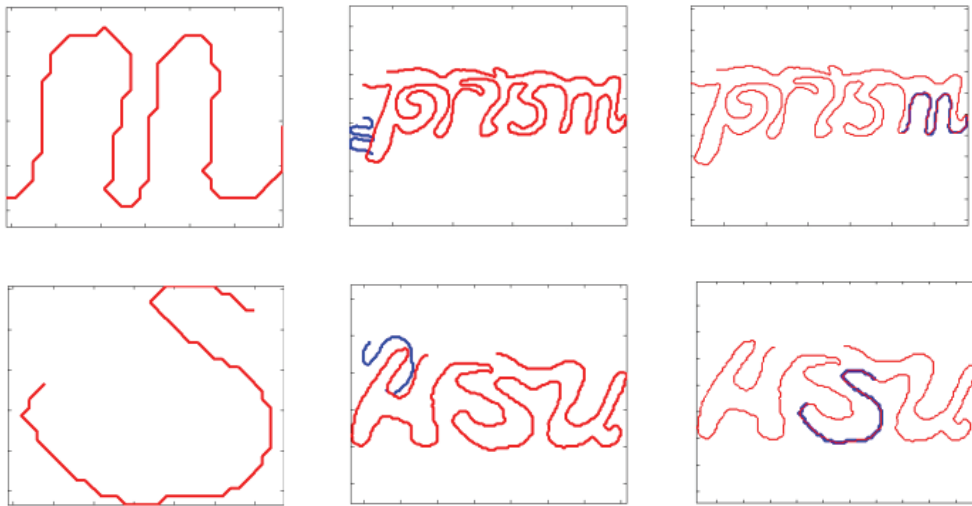


Figure 10: Examples of whole-to-part matching for handwritten strokes. For each row, the left column is the smaller curve that is embedded in the curve shown in middle column, which also shows the two curves in the same coordinate system. The last column shows the matched parts with the similarity transform estimated using our algorithm.

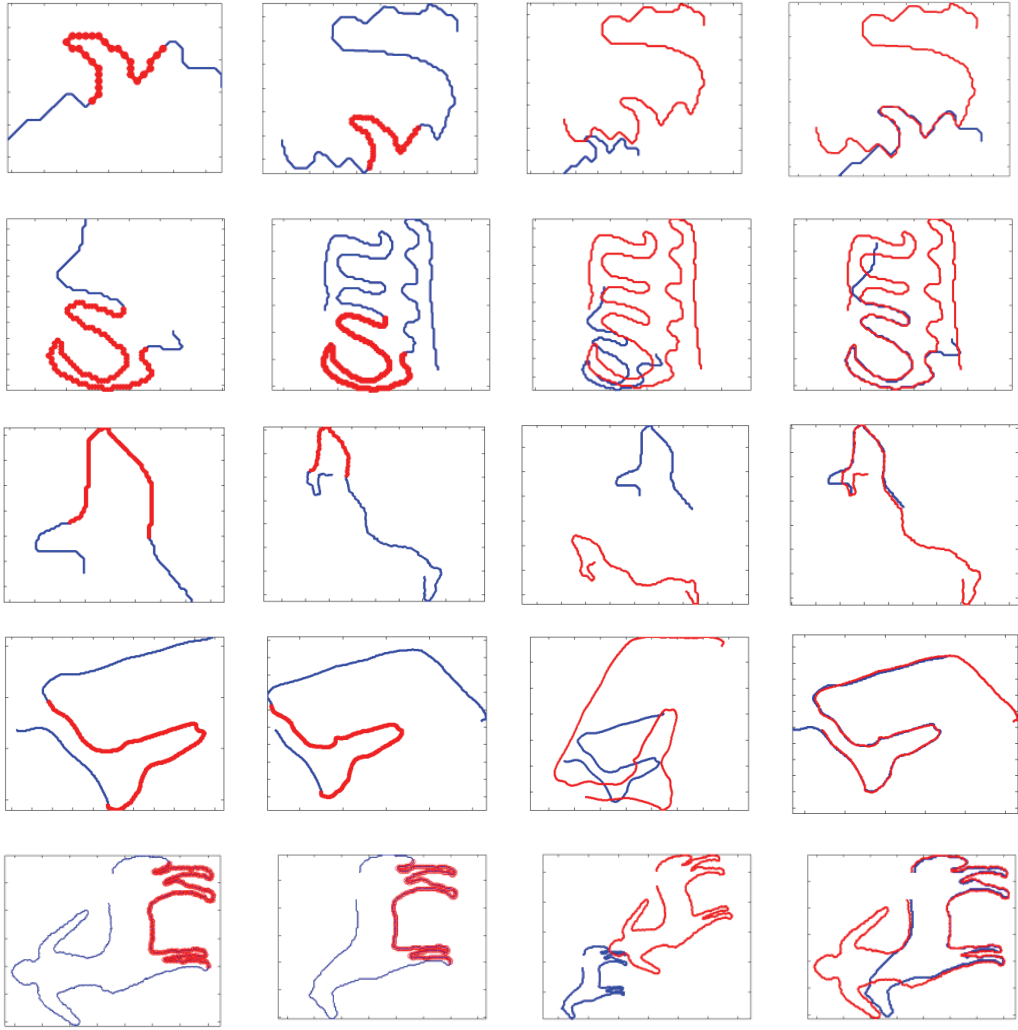


Figure 11: Examples of part-to-part matching. For each row, the first two columns are the input curves that also show the matched parts highlighted in red, the third columns shows the curves in the same coordinate system thereby highlighting the scale difference. The last column shows the aligned curves after our algorithm finds the matching parts and estimates the similarity transform.

Table 1: Comparison of running times for different algorithms. Pair 1,2 and 3 are data shown in the 3rd, 4th, and 5th rows in figure 11 respectively. Notice that the number of samples N are different. For our method, N is linearly proportional to the sum absolute curvature, which is invariant to scale. For Wolfson’s and naive method, N is linearly proportional to arclength, thus also to scale.

Dataset	Our Method		extended Wolfson’s [19]		Naive Search	
	N(samples)	T(time)	N(samples)	T(time)	N(samples)	T(time)
Pair 1	164/435	0.13sec	170/219	0.8sec	170/219	27sec
Pair 1 (X10)	167/434	0.14sec	1704/2198	53sec	1704/2198	>30min
Pair 2	342/719	0.68sec	636/1199	19sec	636/1199	>30min
Pair 2 (X10)	341/723	0.68sec	6362/11995	287sec	6362/11995	>30min
Pair 3	939/993	8.3sec	2064/1956	75sec	2064/1956	>30min
Pair 3 (X10)	927/981	8.2sec	20640/19561	1405sec	20640/19561	>30min

In Table 1, we compare the running times of our algorithm against extended Wolfson’s [19] and the naive search approach. Each data set labeled $PairYY(X10)$ is scaled ten times the original curve $PairYY$. N denotes the number of sample points on the curves, which in turn determines the size of the input. Contrary to our method, the number of sample points required in arclength based methods are dependent on the length of the curve. The running time for our method is invariant to the scale of the curve. For other methods, when the length changes due to scaling of the curve, the size of the input changes, hence the running time is impacted.

6 Conclusions

In this paper we derived a framework with a new curve signature and a parametrization that is invariant under similarity transform. Using this signature, we solve curve matching problems for open 2D curves. Our novel approach is a significant improvement over previous methods. In the future we would like to extend the signature to include affine invariant signature and deal with curve matching under affine transform. Another extension would be to use this for 3D/space curves.

7 Acknowledgment

This work was funded in part by the National Geospatial Agency (NGA), grant # HM1582-05-1-2004), and, National Science Foundation (NSF), grant # IIS-0612269.

References

- [1] K. Arbter, W. E. Snyder, H. Burhardt, and G. Hirzinger. Application of affine-invariant fourier descriptors to recognition of 3-d objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):640–647, 1990.
- [2] K. Briechle and U. D. Hanebeck. Template matching using fast normalized cross correlation. In *Optical Pattern Recognition XII*, volume 4387, pages 95–102, March 2001.
- [3] S. Cohen and L. Guibas. Partial matching of planar polylines under similarity transformations. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 441–444, 1997.
- [4] Berthold K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *A Journal of the Optical Society of America*, 4:629, 4 1987.
- [5] Serkan Kaygin and M. Mete Bulut. Shape recognition using attributed string matching with polygon vertices as the primitives. *Pattern Recogn. Lett.*, 23(1-3):287–294, 2002.
- [6] J.P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, pages 120–123. Canadian Image Processing and Pattern Recognition Society, 1995.
- [7] Simon X. Liao and Mirosław Pawlak. On image analysis by moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(3):254–266, 1996.
- [8] H.-C. Liu and M. D. Srinath. Partial shape classification using contour matching in distance transformation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(11):1072–1079, 1990.

- [9] Siddharth Manay and Byung-Woo Hong. Integral invariants for shape matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1602–1618, 2006.
- [10] Farzin Mokhtarian and Alan K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(8):789–805, 1992.
- [11] Tomas Pajdla and Luc J. Van Gool. Matching of 3d curves using semi-differential invariants. In *ICCV*, pages 390–395, 1995.
- [12] William J. Rucklidge. Efficiently locating objects using the hausdorff distance. *Int. J. Comput. Vision*, 24(3):251–270, 1997.
- [13] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. On aligning curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(1):116–125, 2003.
- [14] Du-Ming Tsai and Chien-Ta Lin. Fast normalized cross correlation for defect detection. *Pattern Recogn. Lett.*, 24(15):2625–2631, 2003.
- [15] Yih-Tay Tsay and Wen-Hsiang Tsai. Attributed string matching by split-and-merge for on-line chinese character recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(2):180–185, 1993.
- [16] Brown university. Kimia shape database.
- [17] Remco C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Proceedings of the International Conference on Shape Modeling & Applications*, page 188, 2001.
- [18] B.B Weixin Kong; Kimia. On solving 2d and 3d puzzles using curve matching. In *CVPR*, pages II–583 – II–590 vol.2, 2001.
- [19] H. J. Wolfson. On curve matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):483–489, 1990.
- [20] H.J. Wolfson and Y. Lamdan. Geometric hashing: A general and efficient model-based recognition scheme. pages 238–249, 1988.

- [21] Xilin Yi and Octavia I. Camps. Line-based recognition using a multidimensional hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(9):901–916, 1999.
- [22] Y. Zhu and L. D. Seneviratne. On the recognition and location of partially occluded objects. *J. Intell. Robotics Syst.*, 25(2):133–151, 1999.
- [23] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.