Partial Matching of Planar Polylines Under Similarity Transformations*

Scott D. Cohen Leonidas J. Guibas
{scohen,guibas}@cs.stanford.edu
Computer Science Department
Stanford University
Stanford, CA 94305

Abstract

Given two planar polylines T and P with n and m edges, respectively, we present an $O(m^2n^2)$ time, $O(mn)$ space algorithm to find portions of the "text" T which are similar in shape to the "pattern" P. In the common case of a simple pattern, such as a line segment or corner, $m = O(1)$ and our algorithm requires $O(n^2)$ time and $O(n)$ space. We use the well-known arclength versus cumulative turning angle graph to judge how well a scaled, rotated, and translated version of the pattern matches a piece of the text. Our match scoring function balances the length of a match against the mean squared error in the match; given two matches with the same mean squared error (length), the longer (lower mean squared error) match will have a higher score. The match score is a function of the pattern scale, orientation, and position within the text, and our algorithm seeks to find local maxima of the scoring function. An analytic formula for the highest scoring pattern orientation in terms of scale and position leaves us with a search problem in scale-position space. This space is divided by a set of lines into combinatorially distinct regions in each of which we have an analytic formula for the scoring function. Our algorithm discovers local maxima of the scoring function by performing a topological line sweep over this scale-position space line arrangement.

1 Introduction

Shape comparison is a fundamental problem in computer vision because of the importance of shape information in performing recognition tasks. For example, many model-based object recognition algorithms work by matching boundary contours of imaged objects to boundary contours of models ([11][13][3][15]). In addition to this traditional application, shape information is also one of the major components in some content-based image retrieval systems ([14][8]). The goal in such a system is to find database images that look similar to a given query image or drawing. Images and queries are usually summarized by their color, shape, and texture content. For example, in the illustration retrieval system described in [4], we suggest a shape index which records *what* basic shapes (such as line segments, corners, and circular arcs) fit *where* in the drawing. The method in this paper can be used to index shape information in images once contours have been extracted.

In this paper, a shape is a polyline in the plane. We consider the following problem: Given two planar polylines, referred to as the text and the pattern, find all approximate occurrences of the pattern within the text, where such occurrences may be scaled and rotated versions of the pattern. We call this problem of locating a polyline pattern shape within a polyline text shape the *polyline shape search problem*, or *PSSP* for short. The PSSP is difficult because we allow both scaling and partial matching.

Two closely related problems to the PSSP are the *segment matching problem* ([12]) and the *polyline simplification problem* ([2][7][9]). The segment matching problem is to find approximate matches between a short polygonal arc and pieces of a longer polygonal arc. In searching for these matches, the short arc is allowed to rotate, but its length/scale remains constant. The PSSP generalizes the segment matching problem by allowing scaling. In the polyline simplification problem, a polyline and error bound are given, and we seek an approximation polyline with the fewest number of segments whose distance to the given polyline is within the error bound. (This problem is also known in the literature as the *min-#* problem.) For a polyline with n vertices, the planar polyline simplification problem can be solved in $O(n^2)$ time if the vertices of the approximation are required to be a subsequence of the vertices of the given polyline ([2]), and in $O(n)$ time if the vertices of the approximation can be arbitrary points in the plane ([16]). For a text polyline with n vertices, our algorithm for the PSSP requires $O(n^2)$ time

*Supported by ARPA grant DAAH01-95-C-R009 and by NSF grant CCR-9215219.

for any constant size pattern.

This paper is organized as follows. In section 2 we describe the framework for our solution to the PSSP, including the match score for a given scale, orientation, and position of the pattern within the text. In section 3 we derive the orientation of the pattern which gives the best match for a fixed pattern scale and position. This leaves us with a 2D search problem in scale-position space (the position is the text arclength at which to begin the comparsion of the pattern with the text). In section 4 we show how a certain set of lines divides up the scale-position plane into regions in each of which we have an analytic formula for our scoring function. In section 5 we fully describe our line sweep algorithm for the PSSP. Some results of this algorithm are shown in section 6. Finally, in section 7 we give some concluding remarks.

2 Problem Setup

Let T and P denote the text and pattern polylines, respectively. We will use the familiar arclength versus cumulative turning angle graph in judging the quality of a match. We denote these summary graphs for the text and pattern as $\Theta(s)$ and $\Psi(s)$, respectively. Figure 1 shows an example. If T consists of n segments and P consists of m segments, then $\Theta(s)$ and $\Psi(s)$ are piecewise constant functions with n and m pieces, respectively. We denote the text arclength breakpoints as $0 = c_0 < c_1 < \cdots < c_n = L$, where L is the length of the text. The value of $\Theta(s)$ over the interval (c_i, c_{i+1}) is denoted by θ_i, $i = 0, \ldots, n-1$. Similarly, we denote the pattern arclength breakpoints as $0 = a_0 < a_1 < \cdots < a_m = l$, where l is the length of the pattern, and the value of $\Psi(s)$ over the interval (a_j, a_{j+1}) is ψ_j, $j = 0, \ldots, m-1$.

Rotating the pattern by angle γ simply shifts its summary graph by γ along the turning angle axis, while scaling the pattern by a factor α stretches its summary graph by a factor of α along the arclength axis. Hereafter, a scaled, rotated version of the input pattern will be referred to as the transformed pattern. The comparison between the transformed pattern and the text will be done in the summary coordinate system. The text arclength at which to begin the comparison will be denoted by β. Since the length of the transformed pattern is αl, the transformed pattern summary graph is compared to the text summary graph from β to $\beta + \alpha l$. Finding the pattern within the text means finding a stretching, right shift, and up shift of the pattern summary graph $\Psi(s)$ that makes it closely resemble the corresponding piece of the text summary graph. Figure 2 illustrates this intuition. The stretching (α), up shifting (γ), and right shifting (β) of the pattern summary graph $\Psi(s)$ correspond to scaling, rotating, and sliding the pattern along the text, respectively. The problem of finding the pattern within the text is thus a search problem in the scale-position-rotation space (α, β, γ).

The preceeding discussion tacitly assumes that the pattern and text are both *open* polylines. If, for example, the text is closed (i.e. the text is a polygon), we can repeat the underlying text summary to ensure that a pattern match that crosses the arbitrary start and finish text arclengths $s = 0$ and $s = L$ will not be missed. In what follows, however, we simply assume that our pattern and text are open polylines.

In judging the quality of a match at a given scale, orientation, and position, we need to consider both the error of the match and the length of the match. When the mean squared error (length) of two matches is equal, the longer (lower mean squared error) match will have a higher score than the shorter (higher mean squared error) match. Of course, there is still the issue of how to compare a match to a shorter (longer), lower mean squared error (higher mean squared error) match. There is no one correct answer to this balancing question – the answer depends, for example, on the underlying input noise model. Here we opt for a simple balancing of match length versus match error which, as we shall see, yields very good results *and* is amenable to analysis via standard calculus optimization techniques. Obviously, other match score functions are possible.

In moving toward our scoring metric, we define the mean squared error $e(\alpha, \beta, \gamma)$ as

$$e(\alpha, \beta, \gamma) = \frac{\int_{s=\beta}^{\beta+\alpha l} \left(\Theta(s) - \left(\Psi\left(\frac{s-\beta}{\alpha}\right) + \gamma \right) \right)^2 \, ds}{\alpha l}.$$

Note that $\Psi\left(\frac{s-\beta}{\alpha}\right) + \gamma$, $s \in [\beta, \beta + \alpha l]$, is the summary graph of the transformed pattern, starting at text arclength β. The score $S(\alpha, \beta, \gamma)$ of a match is defined in terms of the mean squared error e as

$$S(\alpha, \beta, \gamma) = \frac{\alpha l}{L(1 + e(\alpha, \beta, \gamma))}.$$

The product αl is the length of the match (α, β, γ). Our goal is to find local maxima of the score function S over a suitable domain D (in which $S \in [0, 1]$). This domain is defined by restricting the values of α and β so that the domain of definition of the stretched, shifted pattern summary graph is completely contained in $[0, L]$ (the domain of definition of the text summary graph):

$$D = \{ \ (\alpha, \beta) \mid \alpha > 0, \ \beta \geq 0, \ \text{and} \ \alpha l + \beta \leq L \ \}.$$

Although the range of the mean squared error e is $[0, \infty)$, the range of the score S over the domain D is

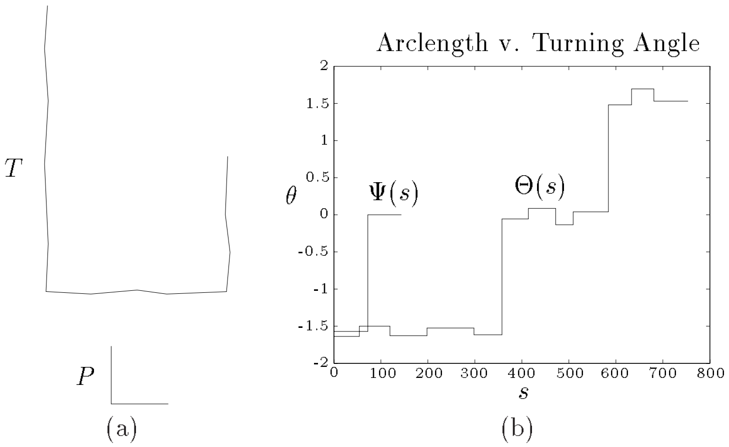


Figure 1: (a) Text T above the corner pattern P. (b) Arclength versus cumulative turning angle functions $\Theta(s)$ and $\Psi(s)$ for T and P, respectively (s in points, θ in radians).

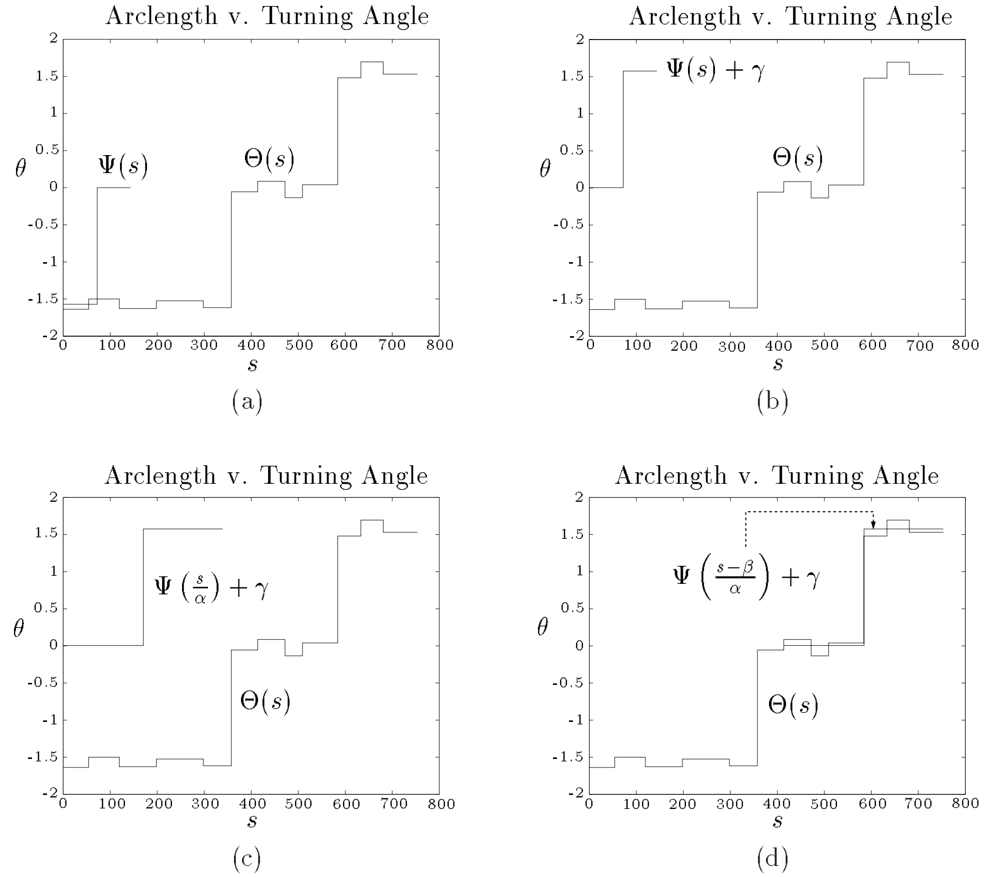


Figure 2: (a) $\Theta(s)$ and $\Psi(s)$ from figure 1b. (b) Rotating the pattern by γ shifts its summary graph up by γ. (c) Scaling the rotated pattern by α stretches the summary graph by a factor of α. (d) Finally, we slide the transformed pattern summary graph over by an amount β to obtain a good match.

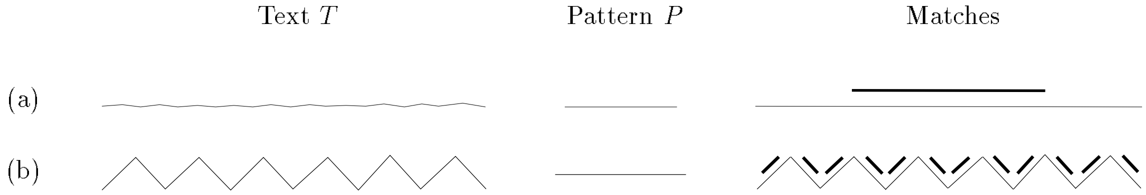


Figure 3: Match error versus match length. In both examples, the pattern is a line segment and the maximum absolute error input parameter $mae_{\max} = 9°$. To help make individual matches clear, we show a darker, smaller scale version of the pattern slightly offset from each match. (a) One match over the length of the entire "noisy" straight line text is found. (b) Twelve matches, one for each of the sides of the "mountain range", are found.

$[0,1]$. A match of length L with zero mean squared error receives the highest score of one. Instead of trying to locate local maxima of S in D, we will try to find local minima of its reciprocal

$$R(\alpha,\beta,\gamma) \equiv \frac{1}{S(\alpha,\beta,\gamma)} = \frac{L}{\alpha l}(1+e(\alpha,\beta,\gamma)).$$

Note that the rotation γ affects only the mean squared error portion of the score.

At a local maximum location $(\alpha_*,\beta_*,\gamma_*)$ of S, a small change in pattern scale, orientation, or position within the text decreases the match score. We do not, however, want to report all local maxima because two very similar matches may be reported. In a sense, we want to report a complete set of independent matches. By independent matches, we mean that any two reported matches should differ significantly in scale, orientation, or position. The matches shown for the two inputs in figure 3 are complete sets of independent matches. These results also illustrate the balancing of match error versus match length. Our matching algorithm and the user-supplied input error bound $mae_{\max}$ are described in section 5.

3 The Best Rotation

In this section we fix (α,β) and derive the rotation angle $\gamma = \gamma_*(\alpha,\beta)$ which minimizes the mean squared error $e(\alpha,\beta,\gamma)$ and, hence, the reciprocal match score $R(\alpha,\beta,\gamma)$. This is straightforward because e is differentiable with respect to γ. The derivative $\partial e/\partial\gamma$ is equal to zero exactly when

$$\gamma = \gamma_*(\alpha,\beta) = \frac{\int_{s=\beta}^{\beta+\alpha l}\left(\Theta(s) - \Psi\left(\frac{s-\beta}{\alpha}\right)\right)\,ds}{\alpha l},$$

the mean value of the difference $\Theta - \Psi$ (more precisely, $\Theta(s) - \Psi\left(\frac{s-\beta}{\alpha}\right)$) over the arclength interval of the match. Since $\partial^2 e/\partial\gamma^2(\alpha,\beta,\gamma) \equiv 2 > 0$, we conclude that for fixed α and β, the value that minimizes the mean squared error is $\gamma = \gamma_*(\alpha,\beta)$ given above. If we define $e_*(\alpha,\beta) \equiv e(\alpha,\beta,\gamma_*(\alpha,\beta))$, then

$$\begin{aligned} e_*(\alpha,\beta) \;=\; & \frac{\int_{s=\beta}^{\beta+\alpha l}\left(\Theta(s) - \Psi\left(\frac{s-\beta}{\alpha}\right)\right)^2 ds}{\alpha l} - \\ & \left(\frac{\int_{s=\beta}^{\beta+\alpha l}\left(\Theta(s) - \Psi\left(\frac{s-\beta}{\alpha}\right)\right) ds}{\alpha l}\right)^2 . \end{aligned}$$

The function $e_*(\alpha,\beta)$ is the variance of $\Theta - \Psi$ over the arclength interval of the match.

4 The 2D Search Problem

The result of the previous section allows us to eliminate the rotation parameter from consideration in our score and reciprocal score functions. We define $R_*(\alpha,\beta) \equiv R(\alpha,\beta,\gamma_*(\alpha,\beta))$. Our goal now is to find local minima in the domain D of

$$(4.1)\quad R_*(\alpha,\beta) = \frac{L}{\alpha l}\left(1 + \frac{I_2(\alpha,\beta)}{\alpha l} - \left(\frac{I_1(\alpha,\beta)}{\alpha l}\right)^2\right),$$

where

$$(4.2)\quad I_2(\alpha,\beta) = \int_{s=\beta}^{\beta+\alpha l}\left(\Theta(s) - \Psi\left(\frac{s-\beta}{\alpha}\right)\right)^2 ds,$$

$$(4.3)\quad I_1(\alpha,\beta) = \int_{s=\beta}^{\beta+\alpha l}\left(\Theta(s) - \Psi\left(\frac{s-\beta}{\alpha}\right)\right) ds.$$

Consider the evaluation of the integral $I_1(\alpha,\beta)$ for a fixed pair (α,β). Since Θ and Ψ are piecewise constant functions, this integral can be reduced to a finite summation of terms such as the product of $(\theta_i - \psi_j)$ with the length of the overlap of the ith arclength interval (c_i, c_{i+1}) of $\Theta(s)$ and the jth arclength interval $(a_j\alpha+\beta, a_{j+1}\alpha+\beta)$ of $\Psi(\frac{s-\beta}{\alpha})$. In precise mathematical terms, we have

$$I_1(\alpha,\beta) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}(\theta_i - \psi_j) \times X_{ij},$$

where $X_{ij} = |(c_i, c_{i+1}) \cap (a_j\alpha + \beta, a_{j+1}\alpha + \beta)|$ and $|(a,b)| = \min\{b-a, 0\}$ is the length of the interval (a,b).

Let l_{ij} denote the line $a_j\alpha + \beta = c_i$, $i = 0,\ldots,n$, $j = 0,\ldots,m$, in the (α,β) plane. The four lines $l_{ij}, l_{i+1,j}, l_{i,j+1}$, and $l_{i+1,j+1}$ divide the (α,β) plane into regions in which we may write down an explicit analytic formula for X_{ij}. In each region, the formula for the size of the intersection is (at worst) a degree one polynomial in α and β. For example, in the region where $a_j\alpha+\beta < c_i$, $a_j\alpha+\beta < c_{i+1}$, $a_{j+1}\alpha+\beta > c_i$, and $a_{j+1}\alpha+\beta < c_{i+1}$, it is easy to check that $X_{ij} = a_{j+1}\alpha + \beta - c_i$. Now let $\mathcal{L}$ denote the set of $(n+1)(m+1)$ lines l_{ij} and let $\mathcal{A} = \mathcal{A}(\mathcal{L})$ denote the arrangement ([5]) in (α,β) space of the lines in $\mathcal{L}$. In each face f of $\mathcal{A}$, we have a degree one polynomial formula for X_{ij}, $X_{ij}^f = u_{ij}^f\alpha + v_{ij}^f\beta + w_{ij}^f$. As explained above, the formula for $X_{ij} = X_{ij}(\alpha,\beta)$ is determined by the above–below relationship of (α,β) and each of the four lines $l_{ij}, l_{i+1,j}, l_{i,j+1}$, and $l_{i+1,j+1}$. From this fact, it is easy to see that the above–below relationship between (α,β) and the line l_{ij} affects only the four intersection formulae $X_{ij}, X_{i-1,j}, X_{i,j-1}$, and $X_{i-1,j-1}$.

An arrangement vertex v_{ijpq} is the intersection of l_{ij} and l_{pq}. The scaling and sliding $(\alpha,\beta) = v_{ijpq}$ of the pattern lines up exactly two pairs of breakpoints: $a_j\alpha + \beta$ coincides with c_i and $a_q\alpha + \beta$ coincides with c_p. An arrangement edge e is an open segment along some line l_{ij}. A scaling and sliding $(\alpha,\beta) \in e$ lines up exactly one pair of breakpoints: $a_j\alpha + \beta$ coincides with c_i. For an open arrangement face f, any scaling and sliding $(\alpha,\beta) \in f$ lines up no pairs of breakpoints.

Now fix a face f of the arrangement $\mathcal{A}$ and let $Y_{ij} = \theta_i - \psi_j$. Then for (α,β) in the closure $\bar{f}$, the integrals (4.2), (4.3) in the formula (4.1) for R_* can be written as

$$(4.4) \qquad I_2(\alpha,\beta) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} X_{ij}^f (Y_{ij})^2$$

$$(4.5) \qquad I_1(\alpha,\beta) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} X_{ij}^f Y_{ij}.$$

Substituting $X_{ij}^f = u_{ij}^f\alpha + v_{ij}^f\beta + w_{ij}^f$ into these formulae and gathering like terms gives

$$(4.6) \qquad I_2(\alpha,\beta) = \hat{u}^f\alpha + \hat{v}^f\beta + \hat{w}^f$$

$$(4.7) \qquad I_1(\alpha,\beta) = \tilde{u}^f\alpha + \tilde{v}^f\beta + \tilde{w}^f,$$

where $\hat{u}^f = \sum_{ij} u_{ij}^f (Y_{ij})^2$, $\tilde{u}^f = \sum_{ij} u_{ij}^f Y_{ij}$, and similarly for $\hat{v}^f$, $\tilde{v}^f$, $\hat{w}^f$, and $\tilde{w}^f$. Combining (4.1), (4.6), and (4.7), we can write R_* in the closed region $\bar{f}$ as

$$(4.8) \quad R_*^f(\alpha,\beta) = \frac{L}{\alpha^3 l^3}(A^f\alpha^2 + B^f\alpha\beta + C^f\beta^2 + D^f\alpha + E^f\beta + F^f)$$

for constants A^f, B^f, C^f, D^f, E^f, F^f.

Our 2D search problem is to find pairs $(\alpha,\beta) \in D$ at which $R_*(\alpha,\beta)$ is a local minimum. It turns out that there are no local minima of R_*^f in the interior of face f (see theorem A.1 in appendix A). So now consider an edge e that bounds face f. We want to know whether R_* has a local minimum at some $(\alpha,\beta) \in e$ in the direction of e. The edge e is part of a line $l_{ij} : a_j\alpha + \beta = c_i$. Combining this line equation with the equation (4.8) for R_*^f, we get a function R_*^e (R_* restricted to the edge e) which is a rational cubic in α (the numerator is quadratic, but the denominator is cubic). A few simple manipulations show that there are at most two local minimum points $(\alpha_*, \beta_*) \in e$ for the function R_*^e, and these locations can be determined in constant time. Local minima of R_* will also commonly occur at arrangement vertices.

5 The Algorithm

The user specifies a minimum and maximum match length $matchlen_{\min}$ and $matchlen_{\max}$ (default maximum is L), along with a bound on the maximum mean absolute error $mae_{\max}$ of a reported match. The bound $mae_{\max}$ can be guaranteed as long as we require the reported match to have a mean squared error which is less than or equal to $mse_{\max} = mae_{\max}^2$ (see theorem B.1 in appendix B). We say that (α,β) is *admissible* if the match length $\alpha l \in [matchlen_{\min}, matchlen_{\max}]$ and the mean squared error $e_*(\alpha,\beta) \le mse_{\max}$. Of all admissible locations (α,β), we report only those which are locally the best. An admissible vertex is reported iff its reciprocal score is less than the reciprocal scores of all adjacent admissible vertices and of all admissible edge minima locations on adjacent edges. An admissible edge minimum is reported iff its reciprocal score is less than the reciprocal scores of its at most two admissible vertices and the other admissible edge minimum (if one exists) on the same edge. Using topological closeness instead of geometric closeness is only a heuristic for reporting independent matches.

Our algorithm outputs matches during a topological sweep ([6]) over the $O(mn)$ lines l_{ij} in the arrangement $\mathcal{A}$. During an elementary step, the topological sweep line moves from a face f_1 into a face f_2 through an elementary step vertex v. Please refer to figure 4.

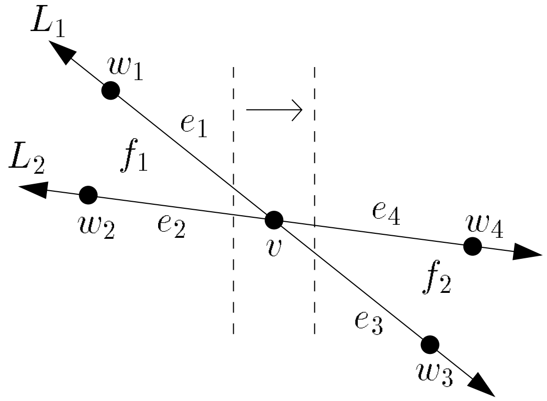


Figure 4: Elementary step notation.

During this step, we decide whether or not to report the vertex v and the local edge minima (if any) on e_3 and e_4. The former decision requires the values $R_*(v)$, $R_*(w_1)$, $R_*(w_2)$, $R_*(w_3)$, $R_*(w_4)$, as well as the values of R_* at all local edge minima of e_1, e_2, e_3, and e_4. The latter decision requires R_* at the local edge minima of e_3 and e_4, and the values $R_*(v)$, $R_*(w_3)$, and $R_*(w_4)$. The values of R_* at local edge minima on e_1 and e_2 follow in constant time from the formulae for $R_*^{e_1}$ and $R_*^{e_2}$. These formulae, as well as the values $R_*(v)$, $R_*(w_1)$, and $R_*(w_2)$ are obtained in constant time from the (already known) formula for $R_*^{f_1}$. Similarly, the values $R_*(w_3)$, $R_*(w_4)$, and R_* at local edge minima of e_3 and e_4 follow in constant time from the formula for $R_*^{f_2}$. Next, we argue that this formula can be computed in $O(1)$ time from the formula for $R_*^{f_1}$.

Computing the formula for $R_*^{f_2}(\alpha,\beta)$ requires computing the coefficients in the formulae (4.6) and (4.7) for the integrals $I_2(\alpha,\beta)$ and $I_1(\alpha,\beta)$, $(\alpha,\beta) \in \bar{f}_2$. Note that computing and summing the $O(mn)$ terms in (4.4) and (4.5) during each elementary step would require total time $O(m^3n^3)$ because there are $O(m^2n^2)$ elementary steps. Fortunately, only a constant number of terms in (4.4) and (4.5) change when we move from face f_1 to face f_2. This is because only a constant number (at most eight) of intersection formulae X_{ij}^f are affected by above–below relationships involving L_1 and L_2. Hence, the values $\hat{u}^{f_2}, \tilde{u}^{f_2}, \hat{v}^{f_2}, \tilde{v}^{f_2}, \hat{w}^{f_2}, \tilde{w}^{f_2}$ in (4.6) and (4.7) can be computed in constant time from the values $\hat{u}^{f_1}, \tilde{u}^{f_1}, \hat{v}^{f_1}, \tilde{v}^{f_1}, \hat{w}^{f_1}, \tilde{w}^{f_1}$, and the formula for $R_*^{f_2}(\alpha,\beta)$ can be computed in $O(1)$ time from the formula for $R_*^{f_1}(\alpha,\beta)$. The latter formula was computed when the sweep line first entered face f_1.

The above discussion shows that the elementary step work specific to our setting may be performed in $O(1)$ time. Thus, the total time to perform the topological sweep over the $O(mn)$ lines l_{ij} is $O(m^2n^2)$. The total space required by our algorithm is the $O(mn)$ storage required by a generic topological line sweep which does not store the discovered arrangement. In the common case of a simple pattern, such as a line segment or corner, $m = O(1)$ and our algorithm requires $O(n^2)$ time and $O(n)$ space.

At the moment when we decide to report a location (α,β), we compute in $O(1)$ time $\gamma = \gamma_*(\alpha,\beta) = I_1(\alpha,\beta)/\alpha l$ and actually report the triple (α,β,γ). The α and γ components give the scaling and rotation parameters of a similarity transformation of the pattern which makes it look like a piece of the text. The β component tells us where along the text this similar piece is located. To get the translation parameters of the similarity transformation, we sample the transformed pattern and the corresponding similar piece of the text, and find the translation parameters which minimize the mean squared error between the translated pattern point set and the text point set.

6 Results

In practice, local edge minima are very rarely reported because there is almost always a smaller admissible minimum at one of the two edge vertices. Essentially, the algorithm reports admissible vertices which have a reciprocal score which is lower than any other adjacent admissible vertex. Recall that arrangement vertices (α,β) give scalings and shifts of the pattern which cause two of its arclength breakpoints to line up with two of the text arclength breakpoints. Our experimentation has thus showed that the best matches of arclength versus turning angle graphs are usually those that line up two pairs of breakpoints (as opposed to one pair for points on arrangement edges and zero pairs for points in arrangement faces).

Figure 5 shows some results of our algorithm. An noteworthy example is figure 5b, which clearly shows that the order of the vertices in the pattern and text makes a difference in the matches found by our algorithm — the three left turn text corners are found, but the right turn is missed. In the example shown in figure 6, we use our method to summarize the straight line content of an image.

7 Conclusion

In this paper we developed an algorithm to find where a planar "pattern" polyline fits well into a planar "text" polyline. By allowing the pattern to rotate and scale, we find portions of the text which are similar in shape to the pattern. All comparisons were performed on the arclength versus cumulative turning angle representations of the polylines. This

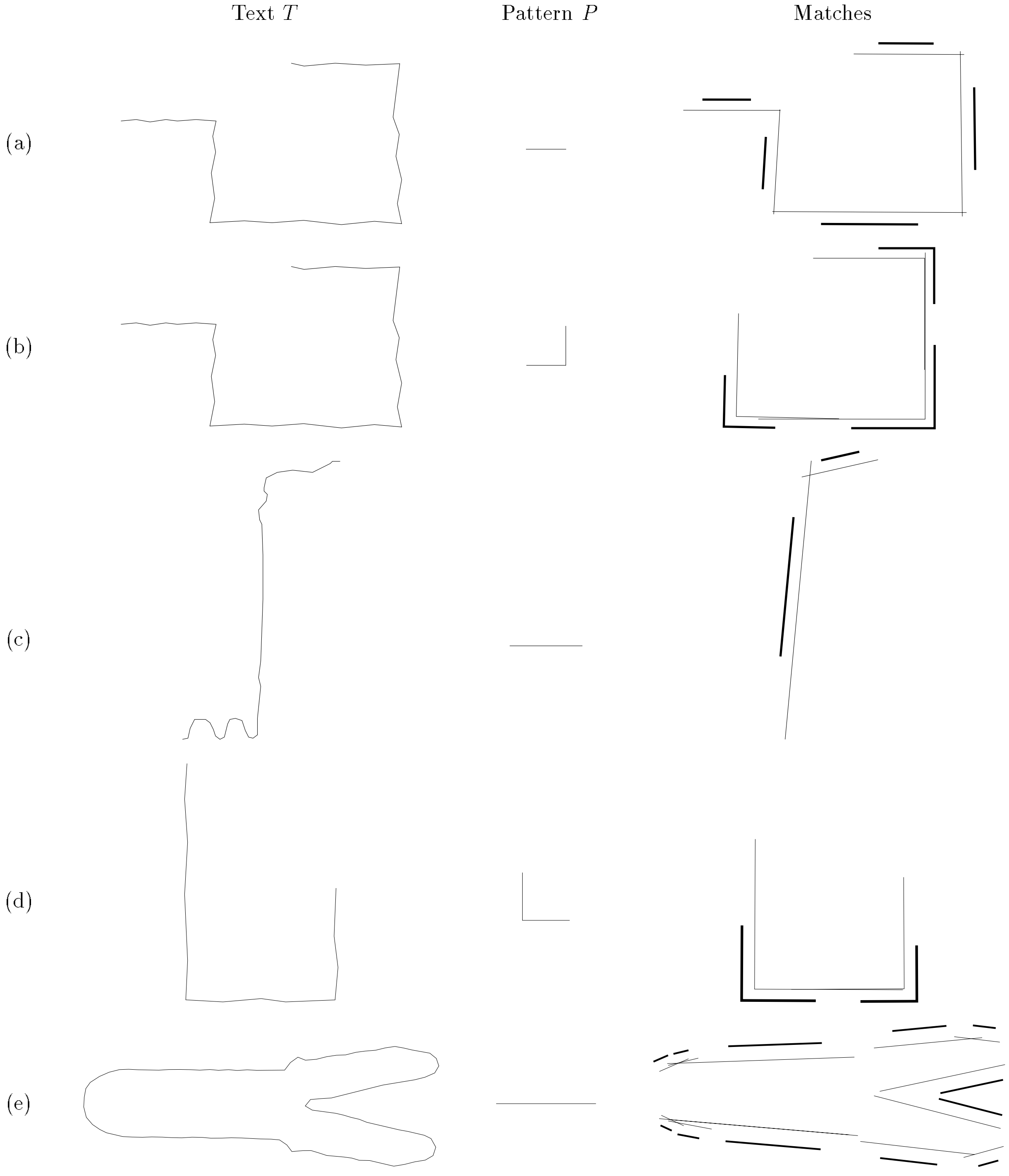
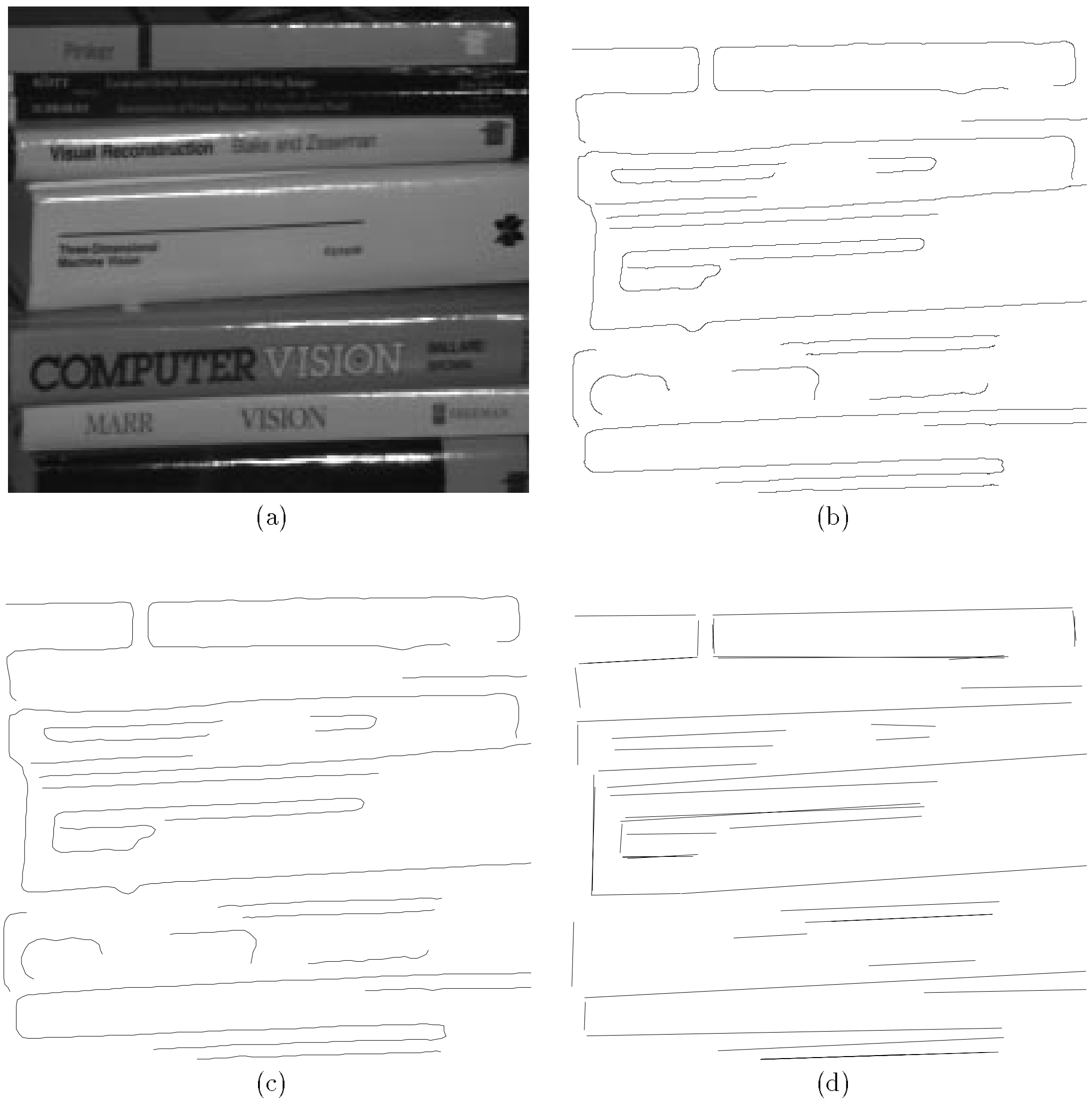


Figure 5: Results. As in figure 3, each match is accompanied by a darker, smaller scale version of the pattern which is slightly offset from the match. (a) $mae_{\max} = 15°$. Each of the five noisy lines gives rise to exactly one segment match. (b) $mae_{\max} = 9°$. The three left turn text corners are found with the left turn corner pattern. (c) $mae_{\max} = 20°$. Our algorithm finds the two (relatively) long, straight portions of the text. (d) $mae_{\max} = 9°$. Both left turn text corners are found. (e) $mae_{\max} = 9°$. The straight pieces of the pliers contour are found.



(a)

(b)

(c)

(d)

Figure 6: Image summary by straight segments. (a) The image to be summarized is 512 × 480. (b) The result of Canny edge detection ($\sigma = 6$ pixels) and edgel linking is a set of polylines with a total of 7219 vertices. (c) The result of subsampling each of the polylines by a factor of 6 leaves a total of 1212 vertices. (d) Finally, fitting a straight segment to each of the subsampled polylines using our PSSP algorithm gives a set of 50 segments. As mentioned in the text, checking only a constant number of topologically neighboring elements before reporting a match (α, β) does not guarantee that two very similar matches (which are geometrically close) will not be reported. This heuristic is responsible for the "double edges" in the image summary.

allowed us to reduce the complexity of the problem: To compare two planar polylines, we compare their one-dimensional arclength versus turning angle graphs. Another reduction in complexity was gained by using the L_2 norm to compare the graphs. This allowed us to eliminate the rotation parameter from our search space, leaving a 2D scale-position space. Thus, we converted a four dimensional search problem in the space of similarity transformations to a two dimensional search problem in scale-position space.

Our line sweep strategy essentially examines all possible pattern scales and positions within the text. If, however, the pattern does not fit well at a certain scale and location, then it will not fit well at nearby scales and locations. Finding "certificates of dissimilarity" which would allow us to prune our (α, β) search space is a topic for future research.

Acknowledgements

We would like to thank Carlo Tomasi for carefully reading the manuscript and providing useful comments.

References

[1] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 129–137, 1990.

[2] W. S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum errors. *International Journal of Computational Geometry & Applications*, 6(1):59–77, Mar. 1996.

[3] F. S. Cohen, Z. Huang, and Z. Yang. Invariant matching and identification of curves using b-splines curve representation. *IEEE Transactions On Image Processing*, 4(1):1–10, Jan. 1995.

[4] S. D. Cohen and L. J. Guibas. Shape-based illustration indexing and retrieval - some first steps. In *Proceedings of the ARPA Image Understanding Workshop*, pages 1209–1212, Feb. 1996.

[5] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, 1987.

[6] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, 38(1):165–194, Feb. 1989.

[7] D. Eu and G. T. Toussaint. On approximating polygonal curves in two and three dimensions. *CVGIP: Graphical Models and Image Processing*, 56:231–246, May 1994.

[8] L. J. Guibas and C. Tomasi. Image retrieval and robot vision research at stanford. In *Proceedings of the ARPA Image Understanding Workshop*, pages 101–108, Feb. 1996.

[9] H. Imai and M. Iri. Polygonal approximations of a curve – formulations and algorithms. In G. T. Toussaint, editor, *Computational morphology : a computational geometric approach to the analysis of form*, pages 71–86. Elsevier Science Publishers, 1988.

[10] R. Johnsonbaugh and W. E. Pfaffenberger. *Foundations of Mathematical Analysis*, pages 288–291. Marcel Dekker, inc., 1981.

[11] A. Kalvin, E. Schonberg, J. T. Schwartz, and M. Sharir. Two-dimensional model-based, boundary matching using footprints. *The International Journal of Robotics Research*, 5(4):38–55, Winter 1986.

[12] B. Kamgar-Parsi, A. Margalit, and A. Rosenfield. Matching general polygonal arcs. *CVGIP: Image Understanding*, 53(2):227–234, Mar. 1991.

[13] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson. Object recognition by affine invariant matching. In *Proceedings of Computer Vision and Pattern Recognition*, pages 335–344, 1988.

[14] W. Niblack et al. The QBIC project: querying images by content using color, texture, and shape. In *Proceedings of the SPIE*, volume 1908, pages 173–187, 1993.

[15] E. J. Pauwels, T. Moons, L. J. Van Gool, and A. Oosterlinck. Recognition of planar shapes under affine distortion. *International Journal of Computer Vision*, 14(1):49–65, Jan. 1995.

[16] S. Suri. On some link distance problems in a simple polygon. *IEEE Transactions on Robotics and Automation*, 6(1):108–113, Feb. 1990.

A The Reciprocal Score Function $R_*^f(\alpha, \beta)$

THEOREM A.1. *The function $R_*^f(\alpha, \beta)$ has no local minima.*

Proof. Consider formula (4.8) for $R_*^f(\alpha, \beta)$. It is easy to verify that $C^f = -(\tilde{v}^f)^2 \leq 0$. If $C^f = 0$, then $R_*^f(\alpha, \beta)$ is linear in β and, therefore, cannot have a local minimum in the β direction at any (α, β). Of course, this means that $R_*^f(\alpha, \beta)$ cannot have a local minimum. The other possibility is $C^f < 0$. The first and second partial derivatives of R_*^f with respect to β are

$$\begin{aligned} \frac{\partial R_*^f}{\partial \beta} &= \frac{L}{\alpha^3 l^3}(B^f \alpha + 2C^f \beta + E^f) \\ \frac{\partial^2 R_*^f}{\partial \beta^2} &= \frac{2L}{\alpha^3 l^3} C^f. \end{aligned}$$

Since $\partial^2 R_*^f / \partial \beta^2 < 0$ (because $C^f < 0$), $R_*^f(\alpha, \beta)$ cannot have a local minimum in the β direction at any (α, β). Therefore, $R_*^f(\alpha, \beta)$ cannot have a local minimum when $C^f < 0$.

B Mean Absolute Error Versus Mean Squared Error

THEOREM B.1. *Let $mae(\alpha,\beta,\gamma)$ and $mse(\alpha,\beta,\gamma)$ denote the mean absolute error and mean squared error, respectively, for a match (α,β,γ). Then $mae^2(\alpha,\beta,\gamma) \leq mse(\alpha,\beta,\gamma)$.*

Proof. It is easy to check that

$$< f, g >= \int_{s=\beta}^{\beta+\alpha l} f(s)g(s)\ ds$$

defines an inner product on $\mathcal{R}[\beta, \beta+\alpha l]$, the set of functions which are integrable on $[\beta, \beta+\alpha l]$, with the minor exception that $< f, f >= 0$ only implies that $f = 0$ almost everywhere in $[\beta, \beta+\alpha l]$ ([10]). Even with this modification, we still have Cauchy's inequality:

$$| < f, g > | \leq ||f||\ ||g||,$$

where $||f|| = \sqrt{< f, f >}$. Applying Cauchy's inequality with $f(s) = |\Theta(s) - \left(\Psi\left(\frac{s-\beta}{\alpha}\right) + \gamma\right)|$ and $g(s) \equiv 1$ gives

$$\int_{s=\beta}^{\beta+\alpha l} \left|\Theta(s) - \left(\Psi\left(\frac{s-\beta}{\alpha}\right) + \gamma\right)\right|\ ds \leq \sqrt{\int_{s=\beta}^{\beta+\alpha l} \left(\Theta(s) - \left(\Psi\left(\frac{s-\beta}{\alpha}\right) + \gamma\right)\right)^2\ ds} \cdot \sqrt{\alpha l}.$$

Squaring both sides of the previous inequality and then dividing by $\alpha^2 l^2$ gives the desired result:

$$\left(\frac{\int_{s=\beta}^{\beta+\alpha l} \left|\Theta(s) - \left(\Psi\left(\frac{s-\beta}{\alpha}\right) + \gamma\right)\right|\ ds}{\alpha l}\right)^2 \leq \frac{\int_{s=\beta}^{\beta+\alpha l} \left(\Theta(s) - \left(\Psi\left(\frac{s-\beta}{\alpha}\right) + \gamma\right)\right)^2\ ds}{\alpha l}.$$