Approximate Matching of Polygonal Shapes[◇]

Helmut Alt*
Bernd Behrends*
Johannes Blömer*

B 93-10
July 1993

Abstract

For two given simple polygons P, Q the problem is to determine a rigid motion I of Q giving the best possible match between P and Q, i.e. minimizing the Hausdorff-distance between P and $I(Q)$. Faster algorithms as the one for the general problem are obtained for special cases, namely that I is restricted to translations or even to translations only in one specified direction. It turns out that determining pseudo-optimal solutions, i.e. ones that differ from the optimum by just a constant factor can be done much more efficiently than determining optimal solutions. In the most general case the algorithm for the pseudo-optimal solution is based on the surprising fact that for the optimal possible match between P and an image $I(Q)$ of Q the distance between the centroids of the edges of the convex hulls of P and $I(Q)$ is a constant multiple of the Hausdorff-distance between P and $I(Q)$. It is also shown that the Hausdorff-distance between two polygons can be determined in time $O(n \log n)$, where n is the total number of vertices.

*Freie Universität Berlin, Fachbereich Mathematik und Informatik, Institut für Informatik, Takustr. 9, D-14195 Berlin, e-mail: alt@tcs.fu-berlin.de, bloemer@tcs.fu-berlin.de.
◇This research was supported by the Deutsche Forschungsgemeinschaft under Grant Al 253/1–2, Schwerpunktprogramm "Datenstrukturen und effiziente Algorithmen" and by the ESPRIT Basic Research Action No. 7141 (ALCOM II).

1 Introduction

The aim of this paper is to present methods from Computational Geometry solving standard problems in pattern recognition which can be intuitively formulated as follows:
Given two objects (shapes) P and Q, how much do they resemble each other? (or: Are they identical up to some tolerance $\delta > 0$?)

In many applications (e.g. character recognition) P will be the input and that Q out of a set of samples has to be determined which is most similar to P. Here we will assume that P and Q are simple polygons in the plane. Geometrically the problem above can be formulated as follows:
Given P, Q, find an *isometry* I such that the *distance* between P and $I(Q)$ is minimized (and determine that minimal distance). Here an isometry is an affine mapping in the plane which preserves distances. As is well known (see [M]) any isometry I can be represented as

$$I = r \circ \rho \circ t \quad \text{or} \quad I = \rho \circ t,$$

where r is the reflexion at the $x-$axis, ρ a rotation about the origin, and t a translation. In this article we will wlog. mean by *isometry* only isometries without reflexions (also called *rigid motions* or even *isometries*). Reflexions can easily be included by first matching optimally P and Q and then P and $r(Q)$ by rigid motions and taking the better of the two matches.
In this sense, any isometry I is of the form

$$I(x) = M \cdot x + t \tag{1}$$

where $M = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix}$ for some $\varphi \in [0, 2\pi[$ and $t \in \mathbf{R}^2$ is some fixed translation vector.

Throughout this article we will denote by $d(x, y)$, $x, y \in \mathbf{R}^2$, the *Euclidean-distance* between x and y, by $\|x\| := d(x, 0)$ the Euclidean norm of x. For a set $A \subset \mathbf{R}^2$ and $\epsilon > 0$ we denote by $U_\epsilon(A)$ the ϵ-*neighborhood* $\{x \in \mathbf{R}^2 | \exists a \in A \text{ such that } d(x, a) < \epsilon\}$ and for $x \in \mathbf{R}^2$ we will write $U_\epsilon(x)$ instead of $U_\epsilon(\{x\})$.

Now, for our polygons P and Q as a *distance* measure between P and $I(Q)$ we will use the so-called *Hausdorff-metric* δ_H that is defined by

$$\delta_H(A, B) = \max(\widetilde{\delta}_H(A, B), \widetilde{\delta}_H(B, A)), \tag{2}$$

where $\widetilde{\delta}_H(X, Y) = \sup_{x \in X} \inf_{y \in Y} d(x, y)$, is the *distance from* X *to* Y.

Notice that $\delta_H(A, B)$ is always defined if $A, B \subset \mathbf{R}^2$ are bounded and that

$$\delta_H(A, B) = \inf\{\epsilon > 0 | A \subset U_\epsilon(B) \text{ and } B \subset U_\epsilon(A)\}. \tag{3}$$

Figure 1 shows two polygons P, Q and an isometry I such that $\delta_H(P, I(Q))$ is minimized. (Note that throughout this paper, when considering a polygon P as a set, we always mean the set of points on the edges of P, not the ones in the interior.)

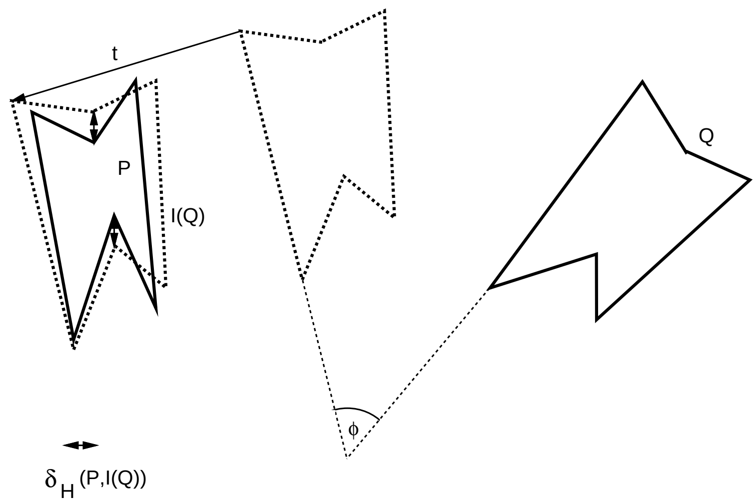

Figure 1:

There are some special cases of the general problem formulated above which are of independent interest. Let P3 denote the general problem then we define problems P2, P1, P0 by the following restrictions:

P2: Only translations are allowed, i.e. in (1) M is the identity matrix Id

P1: Only translations along one fixed direction t_0 are allowed, i.e. $M = Id$ and $t \in \{\lambda \cdot t_0 | \lambda \in \mathbf{R}\}$

P0: No isometries except for the identity are allowed $\left(M = Id, t = \binom{0}{0}\right)$, i.e. the problem is to measure the Hausdorff-distance between P and Q.

A standard example for the application of P2 (or even P1) is again character recognition. P0 is, of course, a very fundamental problem. A linear time algorithm for it in the case of convex polygons has been given by Atallah [At1]. An algorithm for P2 using parametric search was given in [AST].

In Section 2 we will give an $O((p+q)\log(p+q))$ algorithm for P0, where p and q are the numbers of vertices of P, Q, respectively. Then for problem P1 an $O(\lambda_{18}(pq)\log(pq))$ algorithm will be presented using techniques for computing upper envelopes of functions related to Davenport-Schinzel-sequences. Next for problems P2 and P3 algorithms will be briefly sketched whose approximate runtimes are polynomials of degree 7 and 9. Since that is not very efficient any more, we present in Section 5 algorithms giving pseudo-optimal solutions as an alternative. This means that they do not necessarily compute the optimal isometry, but one where the resulting Hausdorff-distance differs from the optimum only by a constant factor. For problem P3 the algorithm is based on the fact that if the minimum distance between P and an isometric image $I(Q)$ of Q is ϵ then the distance between the centroids of the edges of the convex hulls of P and the copy of Q giving the best possible match is at most 17ϵ.

For point sets instead of polygons similar questions as the ones considered in this paper have been investigated in [AMWW] and [S], and more recently, [AKMSW] and [HK]. Problems related to the ones here with respect to an alternative distance measure have been considered in [G] and [AG]. Approximation algorithms in this context have been developed in [ABGW] and [G].

This paper is the complete version of some parts of [ABB].

2 Determining the Hausdorff-distance of two polygons (Problem P0)

Let P, Q be two polygons with p, q vertices, respectively. In order to solve problem P0, i.e. determine the Hausdorff-distance between P and Q, we consider the *Voronoi-diagram* of P, $\text{Vor}(P)$.

$\text{Vor}(P)$ assigns to each edge and each vertex of P its Voronoi-cell, i.e. the set of points in the plane which are closer to this *element* (i.e. edge or vertex) than to any other one (see Figure 2). The edges of $\text{Vor}(P)$ are either line segments (if they separate the cells of two edges or two vertices of P) or parabolic segments (if they separate the cell of a vertex from the cell of an edge). $\text{Vor}(P)$ has $O(p)$ edges and vertices and can be constructed in time $O(p \log p)$ (see [Y], [F]). In order to obtain

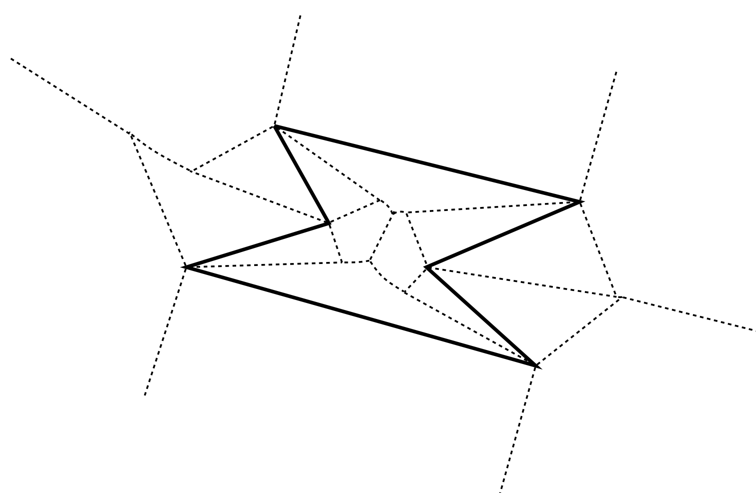

Figure 2: Voronoi Diagram of a Polygon P

a finite problem we observe the following:
First consider the intersection of a fixed Voronoi-cell C with Q (see Figure 3). Suppose that we move monotonically on an edge of Q within this Voronoi-cell C. As easily can be seen the distance to the corresponding element of P defining cell C is a bitonic function, i.e. first decreases and than increases monotonically (or is

just monotone increasing or just monotone decreasing). It follows that the maximal distance of a point of Q on this edge to P must be assumed at the endpoints of the edge or at the intersection point with some Voronoi-edge bounding cell C.

It follows that the distance $\widetilde{\delta}_H(Q,P)$ must be assumed at a vertex of Q or an intersection point of an edge of Q with a Voronoi-edge of P. Furthermore if we move monotonically on a Voronoi-edge e of P the distance to the elements whose cells are separated by this edge is a bitonic function as described before. Summarizing we have

Lemma 1 *The distance of Q to P, $\widetilde{\delta}_H(Q,P)$ is assumed either at some vertex of Q or at some intersection point of Q with some Voronoi-edge e of P having either the smallest or largest $x-$coordinate among the intersection points of Q with e (see Figure 3).*

(In the lemma we assume that parabolic segments having a vertical tangent are cut into two pieces at the point where the vertical tangent occurs.) Notice that the number of points in Lemma 1 is $O(p+q)$. It remains to show how to find these points and their nearest neighbours on P, that is we have to determine the cells of Vor(P) containing the vertices of Q and the elements of P closest to the critical intersection points. We do this by a plane sweep across the arrangement obtained by the edges

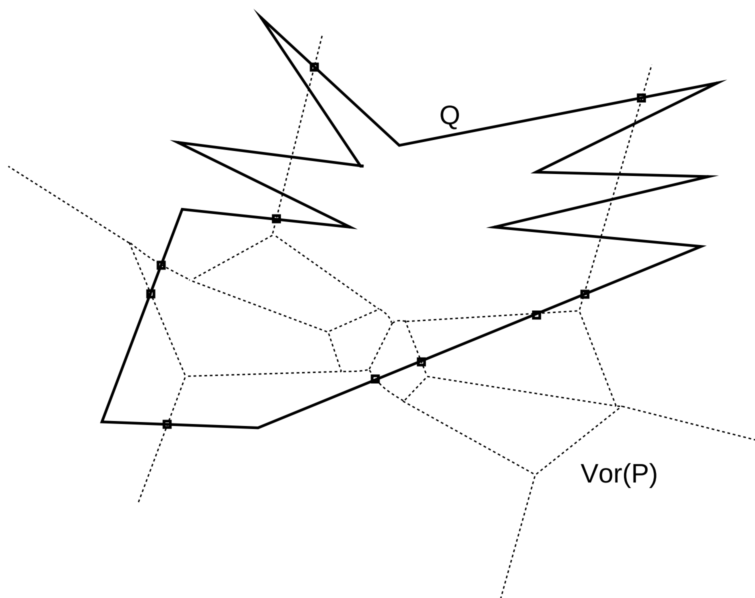

Figure 3:
Vor(P) and Q $\square$ - extreme intersection points.

of Vor(P) and Q. In order to obtain only the extreme intersection points of each edge e of Vor(P), we delete e from the data structure (e.g. 2-3-tree) as soon as the first intersection point with Q has been found. Two such sweeps, one from left to right and one from right to left, are necessary. Since there are $O(p+q)$ event points we obtain an $O((p+q)\log(p+q))$-algorithm for determining all candidates in the sense of Lemma 1. By determining their distance to P and taking their maximum we get $\tilde{\delta}_H(Q,P)$. Analogously, $\tilde{\delta}_H(P,Q)$ and thus $\delta_H(P,Q)$ can be determined.

3 An algorithm for P1

For problem P1 we can assume wlog. that the direction of the allowed translations is parallel to the x-axis, i.e. translation vectors are of the form $(\lambda, 0)$, $\lambda \in \mathbf{R}$. For $\lambda \in \mathbf{R}$ and an egde e of Q we denote by $I_\lambda(e)$ the image of e.

Suppose e' is an edge of $\mathrm{Vor}(P)$ bounding some cell C. For any fixed value of λ $I_\lambda(e)$ has at most two intersection points with e'. We consider the square of the distance of such an intersection point to the object in P defining cell C as a function in λ. Since e' is a parabolic or a straight line segment this function is clearly algebraic and a detailed analysis shows that its order is at most 4. It is not hard to verify that each pair e, e', $e \in Q$, e' an edge of $\mathrm{Vor}(P)$, generates at most 3 such algebraic functions in λ, whose domains are intervals (see Figure 4). Likewise we

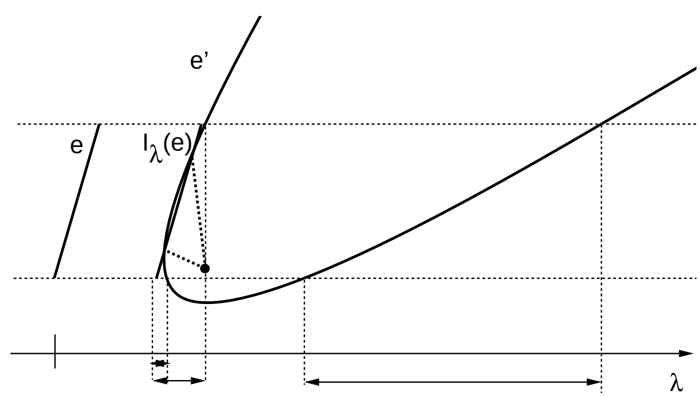

Figure 4:

define for each pair (a, C), where a is an endpoint of some edge e of Q and C a cell of Vor(P), the function $f_{a,C}$, i.e. if the corresponding endpoint of $I_\lambda(e)$ is contained in C $f_{a,C}(\lambda)$ is defined as the square of the distance of this point to the site defining cell C.

Obviously, $f_{a,C}(\lambda)$ is a quadratic function. According to Lemma 1 the Hausdorff distance $h(\lambda) := \widetilde{\delta_H}(I_\lambda(Q), P)$ is the maximum of all functions described previously, i.e. h is the *upper envelope* of all these functions (see Figure 5).

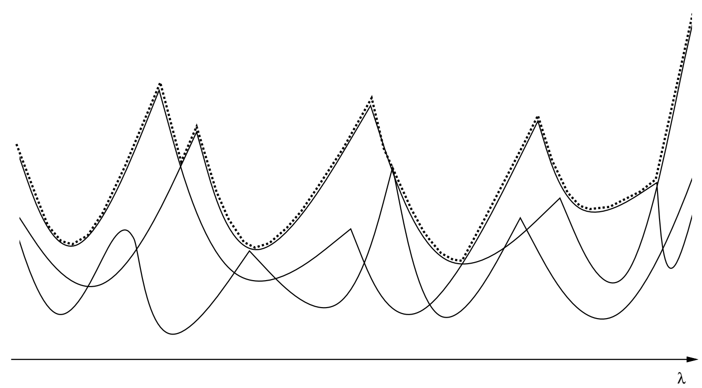

Figure 5:

Problem P1 can now be reduced to finding the minimum of the function $h(\lambda)$. Clearly $h(\lambda)$ is a piecewise algebraic function. Constructing upper envelopes of sets of functions is well studied in the theory of *Davenport-Schinzel sequences* (see [ASS], [At2]). There the number of pieces of the upper envelope of n functions from which any pair can intersect at most k times is denoted by $\lambda_k(n)$. The upper envelope can be constructed (see [At2]) in time $O(\lambda_k(n) \log n)$. No explicit expression is known for $\lambda_k(n)$ if $k > 4$, but it is known that the growth rate is only slightly above linear for any constant k. In fact $\lambda_k(n) = o(n \log^* n)$ (where $\log^* n$ is the number of times log has to be applied to get down from n to some value ≤ 1). In our case $h(\lambda)$ is the upper envelope of $O(pq)$ algebraic functions of degree at most 4, consequently any two of them intersect in at most 16 points by Bezout's theorem (see [Fu]). Since the domain of these functions is not necessarily the whole of $\mathbf{R}$ but some interval we additionaly have to take into consideration the interval endpoints and get $O(\lambda_{18}(pq))$ as the number of pieces $h(\lambda)$ consists of and

$$O(\lambda_{18}(pq) \log pq)$$

for the time to construct it (and, thus to find its minimum).

In the same way, we can determine the distance from P to $I_\lambda(Q)$ as a function of λ. By merging the two functions we can determine the optimal λ in time

$$O\big(\lambda_{18}(pq) \log(pq)\big).$$

4 Pseudo-optimal solutions for P2 and P3

In [ABB] problems P2 and P3 were solved by observing that for the optimal placement of Q the Hausdorff-distance must occur at at least 3 for P2 and 4 for P3 different places (except for degenerate cases). This observation led to brute force algorithms of runtimes $O\left((pq)^3(p+q)\log(p+q)\right)$ for P2 and $O\left((pq)^4(p+q)\log(p+q)\right)$ for P3. Meanwhile in [AST] an algorithm for P2 of runtime $O\left((pq)^2\log^3(pq)\right)$ has been found using the technique of *parametric search*.

In this section, we will present a different approach, which gives much more efficient and practical algorithms. However, it does not necessarily find the optimal solution but one which is not too bad in the following sense:

Definition 2 *An algorithm is said to produce a* pseudo-optimal *solution for problem P2 (P3), iff there is a constant $c > 0$ such that on input P, Q the algorithm finds a translation (isometry) I with $\delta_H(P, I(Q)) \leq c\delta$, where δ is the minimal Hausdorff-distance determined by the optimal solution.*

A pseudo-optimal solution for P2 can be found very easily:
For a polygon P let $r_P := (x_P, y_P)$ where $x_P(y_P)$ is the smallest x-coordinate (y-coordinate) of any point in P (see Figure 6). Let P, Q be two polygons and I a

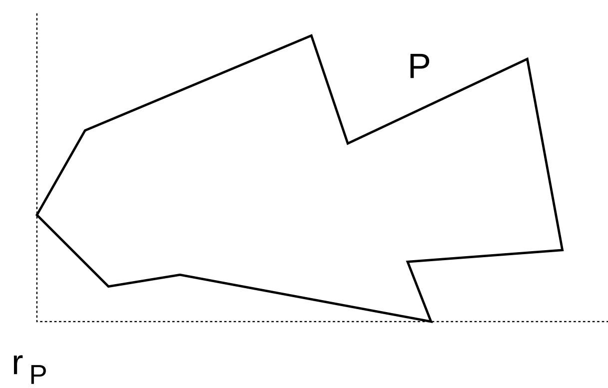

Figure 6:

solution to P2, i.e. $\delta := \delta_H(P, I(Q))$ is minimal. Obviously $d(r_P, r_{I(Q)}) \leq \sqrt{2}\delta$. Therefore, if $\widetilde{I}$ is the translation mapping r_Q *onto* r_P, its difference to the optimal one is a vector of length at most $\sqrt{2}\delta$. Hence,

$$\delta_H(P, \widetilde{I}(Q)) \leq (1+\sqrt{2})\delta$$

i.e. $\widetilde{I}$ is a pseudo-optimal solution. Since r_P, r_Q can be determined in time $O(p+q)$, the same holds for $\widetilde{I}$; if we also want the value of $\delta_H(P, \widetilde{I}(Q))$ we have to apply the algorithm for P0 and finally get a runtime of $O\big((p+q)\log(p+q)\big)$.

Of course, the point r_P is not a suitable choice for problem P3 since its position relative to P is not invariant under rotations. Instead, we define for a polygon P

S_P to be the centroid of the edges of the convex hull $\widetilde{P}$ of P. (In the following $\widetilde{P}$ will always denote the *boundary* of the convex hull of P not its interior.) One way to compute S_P (in time $O(p)$) is to assign for each edge e of $\widetilde{P}$ the length of e as a weight to the midpoint of e and compute the weighted arithmetic mean of all these midpoints.

An alternative definition of S_P, which we will use here, is by *parametrizations* of $\widetilde{P}$, i.e. continuous mappings $\alpha : [a, b] \rightarrow \mathbf{R}^2$, where $[a, b]$ is a real interval such that the image of α equals $\widetilde{P}$ (see [E]). In addition to the standard definition we will assume here that α is *injective* everywhere, except that, since we are considering closed curves, $\alpha(a) = \alpha(b)$. In particular, we will consider *natural parametrizations*, i.e. parametrizations $\alpha : [0, L_{\widetilde{P}}] \rightarrow \mathbf{R}^2$, where $L_{\widetilde{P}}$ is the length of $\widetilde{P}$, i.e. the total length of its edges. Furthermore for any $t \in [0, L_{\widetilde{P}}]$ the arc-length from point $\alpha(0)$ on $\widetilde{P}$ to $\alpha(t)$ on $\widetilde{P}$ equals t. Now elementary geometric considerations show that

$$S_P = \frac{1}{L_{\widetilde{P}}} \int_0^{L_{\widetilde{P}}} \alpha(t) dt,$$

if α is a natural parametrization of $\widetilde{P}$.

The following lemma states that S_P is indeed a suitable choice for finding a pseudo-optimal solution:

Lemma 3 *Let P, Q be polygons and I the isometry minimizing $\delta := \delta_H(P, I(Q))$. Assume furthermore wlog. that $I(Q)$ contains the origin. Then*

$$d(S_P, S_{I(Q)}) \leq (4\pi + 3)\delta.$$

For proving Lemma 3 we need a few facts about parametrized curves. First we consider an alternative distance measure for curves, the so-called *Fréchet-distance* (see also [E], [AG]):

Definition 4 *Let C_1, C_2 be curves. Then the Fréchet-distance is defined as*

$$\delta_F(C_1, C_2) := \inf_{\alpha}(\max\{d(n_{C_1}(t), \alpha(t)) | t \in [0, L_{C_1}]\}), \tag{4}$$

where α ranges over all possible injective parametrizations

$$\alpha : [0, L_{C_1}] \rightarrow \mathbf{R}^2$$

of C_2 and n_{C_1} is a natural parametrization of C_1.

δ_F can be visualized as follows:
Suppose there is a man walking his dog, the man walking on curve C_1, the dog on C_2. $\delta_F(C_1, C_2)$ is the minimal length of a leash that is possible.
It has been proven in [ABGW]:

Lemma 5 *For any pair of convex closed curves* $C_1, C_2 : \delta_F(C_1,C_2) = \delta_H(C_1,C_2)$, *in fact to any natural parametrization* n_{C_1} *of* C_1 *there exists a parametrization* $\alpha : [0, L_{C_1}] \to \mathbf{R}^2$ *of* C_2 *with* $d(n_{C_1}(t), \alpha(t)) \le \delta_H(C_1, C_2)$ *for all* $t \in [0, L_{C_1}]$.

Lemma 6 *[Ben, Thm. 14.12] Let* C_1 *and* C_2 *be convex closed curves,* L_{C_1}, L_{C_2} *their lengths and* $\delta = \delta_H(C_1, C_2)$. *Then* $|L_{C_1} - L_{C_2}| \le 2\pi\delta$.

Lemma 7 *Let* $A, B \subset \mathbf{R}^2$ *compact, and* $\widetilde{A}, \widetilde{B}$ *their convex hulls. Then* $\delta_H(\widetilde{A}, \widetilde{B}) \le \delta_H(A, B)$.

Proof: Let $\delta := \delta_H(A,B)$. $A \subset U_\delta(B)$ implies $\widetilde{A} \subset \widetilde{U_\delta(B)}$. Since $U_\delta(B) \subset U_\delta(\widetilde{B})$ and $U_\delta(\widetilde{B})$ is convex, it follows $\widetilde{U_\delta(B)} \subset U_\delta(\widetilde{B})$, so $\widetilde{A} \subset U_\delta(\widetilde{B})$. Analogously, $\widetilde{B} \subset U_\delta(\widetilde{A})$, which proves the lemma. $\square$

Now we can give the **proof of Lemma 3:**
Let $R{:=}\widetilde{P}$, $T{:=}\widetilde{I(Q)}\left(= I(\widetilde{Q})\right)$, α a natural parametrization of R, and $\widetilde{\beta}$ a parametrization of T such that according to Lemma 5:

$$
\begin{aligned}
d\left(\alpha(t), \widetilde{\beta}(t)\right) &\le \delta_H(R,T) \quad \text{for all } t \in [0, L_R] && (5)\\
&\le \delta \quad \text{by Lemma 7.} && (6)
\end{aligned}
$$

Let $\beta : [0, L_T] \to \mathbf{R}^2$ be the natural parametrization of T with $\beta(0) = \widetilde{\beta}(0)$, and the orientation in which β traverses T is the same as the one of $\widetilde{\beta}$. Now,

$$
\begin{aligned}
d(S_P, S_{I(Q)}) &= \left\|\frac{1}{L_R}\int_0^{L_R} \alpha(t)dt - \frac{1}{L_T}\int_0^{L_T} \beta(t)dt\right\| \\
&\le \frac{1}{L_R}\int_0^{L_R} \|\alpha(t) - \beta(t)\| dt \\
&+ \left\|\left(\frac{1}{L_R} - \frac{1}{L_T}\right)\int_0^{L_R} \beta(t)dt\right\| \\
&+ \frac{1}{L_T}\left\|\int_{L_R}^{L_T} \beta(t)dt\right\|,
\end{aligned}
$$

assuming wlog. that $L_T \ge L_R$. Let us denote the three terms in the last expression by J_1, J_2, J_3, respectively.

In order to get an upper bound on J_2 and J_3 observe first that T is a closed curve, hence its length L_T is at most twice its diameter. Since T also contains the origin it follows

$$\|\beta(t)\| \le L_T/2 \tag{7}$$

for all $t \in [0, L_T]$. Hence

$$\begin{aligned} J_2 &\leq \left(\frac{1}{L_R} - \frac{1}{L_T}\right) \int_0^{L_R} \|\beta(t)\| dt \\ &\leq \frac{L_T - L_R}{L_T L_R} L_R \frac{L_T}{2} && \text{by (7)} \qquad (8) \\ &\leq \pi\delta && \text{by Lemma 6} \end{aligned}$$

and

$$\begin{aligned} J_3 &\leq \frac{1}{L_T} \int_{L_R}^{L_T} \|\beta(\alpha)\| dt \\ &\leq \frac{1}{L_T} (L_T - L_R) \frac{L_T}{2} \\ &\leq \pi\delta \end{aligned}$$

again by (7) and Lemma 6. In order to get an upper bound for J_1, we show

Claim: $\|\alpha(t) - \beta(t)\| \leq (2\pi + 3)\delta$ for all $t \in [0, L_R]$

Proof: For a fixed $t \in [0, L_R]$ consider the curve segments from $\alpha(0)$ to $\alpha(t)$ of R and from $\widetilde{\beta}(0)$ to $\widetilde{\beta}(t)$ of T and close them by line segments ℓ_R and ℓ_T (see Figure 7). The resulting curves R', T' have Hausdorff-distance $\leq \delta$. By definition (see (5)) this

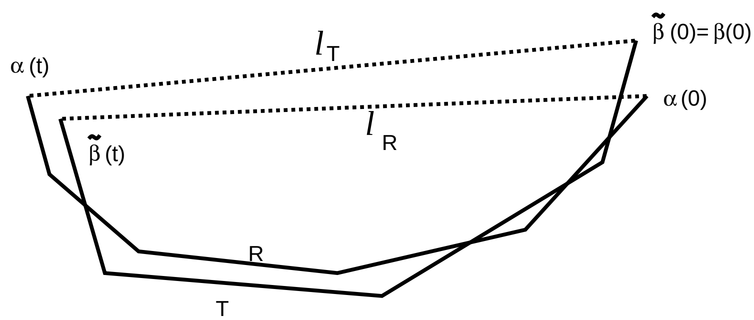

Figure 7:

is correct for the curve segments themselves, for the line segments ℓ_R, ℓ_T it holds because their respective endpoints have distance at most δ. By Lemma 6 it follows

$$|L_{R'} - L_{T'}| \leq 2\pi\delta \tag{9}$$

Now, if b is the arc-length of T from $\widetilde{\beta}(0)$ to $\widetilde{\beta}(t)$, and l_R, l_T the lengths of ℓ_R, ℓ_T, respectively, then

$$\begin{aligned} |L_{R'} - L_{T'}| &= |t + \ell_R - b - \ell_T| \\ &\geq |t - b| - |\ell_R - \ell_T| \end{aligned} \tag{10}$$

Since $|l_R - l_T| \leq 2\delta$, we have by (9) and (10):

$$|t - b| \leq (2\pi + 2)\delta$$

On the other hand since b is the arc-length of T between $\widetilde{\beta}(0)$ and $\widetilde{\beta}(t)$ and t the arc-length of T between $\beta(0)$ $\left(= \widetilde{\beta}(0)\right)$ and $\beta(t)$, we get

$$\begin{array}{rrcl} & \|\widetilde{\beta}(t) - \beta(t)\| & \leq & |t - b| \\ \text{So} & \|\beta(t) - \alpha(t)\| & \leq & \|\beta(t) - \widetilde{\beta}(t)\| + \|\widetilde{\beta}(t) - \alpha(t)\| \\ & & \leq & (2\pi + 2)\delta + \delta \end{array}$$

and the claim follows.

Clearly, the claim implies that $J_1 \leq (2\pi + 3)\delta$, hence

$$d(S_P, S_{I(Q)}) \leq (4\pi + 3)\delta,$$

which finishes the proof of Lemma 3. $\square$

From Lemma 3 we obtain with the same arguments as for problem P2:

Lemma 8 *Let $\widetilde{I}$ be an isometry which gives a minimal Hausdorff-distance among the ones mapping S_Q onto S_P. Then*

$$\delta_H(P, \widetilde{I}(Q)) \leq (4\pi + 4)\delta;$$

where δ is the optimal solution, i.e. $\widetilde{I}$ is a pseudo-optimal solution.

$\widetilde{I}$ can be found by translating Q such that S_Q is mapped onto S_P and then rotating the image of Q around S_P. The angle $\widetilde{\varphi}$ of rotation which gives the optimal solution $\widetilde{I}$ can be determined by a technique analogous to the one used for solving problem P1.

In fact, let us first assume that $\widetilde{\varphi} \in [0, \pi]$, the case $\widetilde{\varphi} \in [\pi, 2\pi]$ can be solved analogously. Rather than using the angle φ itself as a parameter for the rotation, we use $c := \cos\varphi \in [-1, 1]$ which is bijective on the interval considered. Also, by applying a simple translation, we may assume that the rotation is about the origin and, thus, is described by some rotation matrix

$$I_c = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$

where

$$c^2 + s^2 = 1. \tag{11}$$

Analogously to P1, we want to describe the one-sided Hausdorff-distance $\widetilde{\delta_H}\,(I_c(Q), P)$ as a function in c. As for P1 this function is the upper envelope of $O(pq)$ functions obtained from pairs e, e' where e is an edge of Q and e' one of $\mathrm{Vor}(P)$ and $O(pq)$ functions obtained from pairs v, f where v is a vertex of Q and f a vertex or edge of P.

Let us first consider the intersection points of edges e, e'. If e' is a parabolic segment (see Figure 8) let

$$q(x, y) = 0 \tag{12}$$

be a quadratic equation describing the corresponding parabola. Let $\ell(x, y) = 0$ be

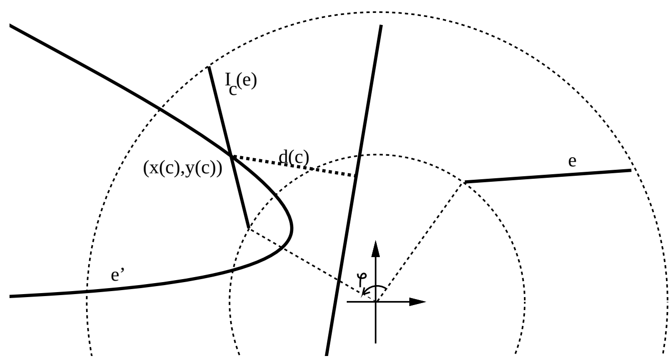

Figure 8:

a linear equation describing the straight line through e. Then $I_c(e)$ is described by $\ell\left(I_c^{-1}(x, y)\right) = 0$ which is a (inhomogeneous) bilinear form in x, y, c and s, i.e.

$$B(x, y, c, s) = 0. \tag{13}$$

From (11), (12) and (13) with four unknowns we can eliminate y and s and obtain x as (constantly) many branches of an algebraic function in c. Then from (12) we obtain y as function in c. The function we are finally looking for is the distance $d(c)$ of $(x(c), y(c))$ to the edge of P whose Voronoi cell is bounded by e'. It has the form

$$d(c) = a_1 x(c) + a_2 y(c) + a_3 \tag{14}$$

for constansts a_1, a_2, a_3 and therefore is also an algebraic function of constant degree (A detailed analysis shows that its degree is at most 32.).

In the case where e' is a straight line segment, equation (12) is linear instead of quadratic and equation (14) the square root of a quadratic function instead of linear function. It can be shown that $d(c)$ is then an algebraic function of degree less than 32.

The functions resulting from the distances between $I_c(v)$ and f, v a vertex of Q and f a vertex or edge of P are, as a detailed analysis shows, algebraic functions in c of degree at most 4. Altogether, we have $O(pq)$ algebraic functions of constant degree so they intersect pairwise in constantly many points (using Bezout's theorem, see [Fu]). Since in our case the functions are only defined on finite intervals, like in the analysis for P1 we have to add the endpoints of these intervals as critical points. So the number of segments the upper envelope consists of is $\lambda_k(pq)$ for some constant k and it can be constructed in time

$$O\left(\lambda_k(pq)\log(pq)\right) \tag{15}$$

(see [At2]). Likewise, within the same runtime we can construct $\widetilde{\delta_H}(P, I_c(Q))$ as a function in c, determine the maximum f of both functions and the minimum of f which is $\delta_H(P, \widetilde{I}(Q))$. Since S_P and S_Q can be found in linear time this bounds also the runtime of the whole algorithm.

A detailed analysis shows that $k = 1026$ is sufficient in (15). As was mentioned before $\lambda_k(pq) = O\left(pq\log^*(pq)\right)$ for any constant k, but the constant in the O-term may become quite large. But although the analysis is rather complicated the algorithm of [At2] is simple and it should behave reasonably in practice for our problem.

Also the constant of $4\pi + 4 \approx 17$ in Lemma 8 may seem large, but with the following idea (cf. [S]) it can be reduced to any fixed constant $c > 1$ without increasing the asymptotic runtime:
We know by Lemma 3 that the optimal isometry I maps S_Q into the $(4\pi + 4)\delta$-neighborhood U of S_P. We place onto U a sufficiently small grid so that no point in U has distance greater than $(c-1)\delta$ from a gridpoint. Since c is fixed, there are constantly many gridpoints within U. We place S_Q instead of onto S_P only, onto each one of these gridpoints and proceed as described before. It follows from the previous discussion that for the solution $\widetilde{I}$ found this way it holds:

$$\delta_H(P, \widetilde{I}(a)) \leq c\delta.$$

5 Conclusion

Let us summarize the results of this paper, using explicit upper bounds for $\lambda_k(pq)$, k constant:

Theorem 9 *The different versions of the problem of measuring the resemblance between polygons P, Q with p, q vertices respectively, can be solved within the following time bounds:*

$$
\begin{aligned}
P0: &\quad O\left((pq)\log(pq)\right)\\
P1: &\quad O\left((pq)\log(pq)\log^*(pq)\right)\\
P2: &\quad O\left((pq)\log(pq)\right)\\
P3: &\quad O\left((pq)\log(pq)\log^*(pq)\right),
\end{aligned}
$$

where the algorithms for P0 and P1 give optimal, the ones for P2 and P3 pseudo-optimal solutions.

Finally observe that we never really used in our algorithm that P and Q are polygons. In fact, we obtain:

Corollary 10 *Theorem 9 not only holds for polygons, but also for more general structures like polygonal chains, in fact, for arbitrary sets of nonintersecting line segments.*

Acknowledgement We want to thank Günter Rote for several ideas and helpful comments concerning this article.

References

[ABB] H. Alt, B. Behrends, J. Blömer, "Approximate Matching of Polygonal Shapes", Proc. 7th Annual ACM Symp. on Computational Geometry, 1992, pp. 186-193.

[ABGW] H. Alt, J. Blömer, M. Godau, H. Wagener, "Approximations of Convex Polygons", Proc. 17th ICALP, Springer Lecture Notes in Computer Science 443, 1990, pp. 703-716.

[AG] H. Alt, M. Godau, "Measuring the Resemblance of Polygonal Curves", Proc. 8th Annual ACM Symp. on Computational Geometry, 1992, pp. 102-109.

[AKMSW] E. M. Arkin, K. Kedem, J. S. B. Mitchell, J. Sprinzak, M. Werman, "Matching Points into Noise Regions: Combinatorial Bounds and Algorithms", Proc. 2nd Annual ACM-SIAM Symposium on Discrete Algorithms, 1991, pp. 42-51.

[AMWW] H. Alt, K. Mehlhorn, H. Wagener, E. Welzl, "Congruence, Similarity and Symmetries of Geometric Objects", *Discrete Comp. Geom.* 3, 1988, pp. 237-256.

[ASS] P. K. Agarwal, M. Sharir, P. Shor, "Sharp Upper and Lower Bounds on the Length of General Davenport-Schinzel Sequences", *J. Combin. Theory*, Ser. A 52, 1989, pp. 228-274.

[AST] P. K. Agarwal, M. Sharir, S. Toledo, "Applications of Parametric Searching in Geometric Optimization", Proc. 3rd ACM-SIAM Symp. Discrete Algorithms (SODA '92), 1992, pp. 72-82.

[At1] M. J. Atallah, "A Linear Time Algorithm for the Hausdorff-distance between Convex Polygons", *Information Processing Letters* 17, 1983, pp. 207-209.

[At2] M. Atallah, "Dynamic Computational Geometry", Proc. 24th Annual Symp. on Foundations of Computer Science, 1983, pp. 92-99 also in *Comp. Math. Appl.* 11, 1985, pp. 1171-1181.

[Beh] B. Behrends, "Algorithmen zur Erkennung der ϵ-Kongruenz von Punktmengen und Polygonen", Diplomarbeit, FB Mathematik, FU Berlin, 1990.

[Ben] R.V. Benson, *Euclidean Geometry and Convexity*, McGraw-Hill, 1966.

[E] G.M. Ewing, *Calculus of Variations with Applications*, Dover Publ., New York 1985.

[EPSS] H. Edelsbrunner, J. Pach, J. T. Schwartz, M. Sharir, "On the Lower Envelope of Bivariate Functions and its Applications", Proc. 28th Annual Symp. on Foundations of Computer Science, 1987, pp. 27-37.

[F] S. Fortune, "A Sweepline - Algorithm for Voronoi-Diagrams", *Algorithmica 2*, 1987, pp. 153-174.

[Fu] W. Fulton, *Algebraic Curves*, W.A. Benjamin Inc., New York, Amsterdam, 1969.

[G] M. Godau, "A Natural Metric for Curves - Computing the Distance for Polygonal Chains and Approximation Algorithms", Proc. 8th STACS, Springer Lecture Notes in Computer Science 480, 1991, pp. 127-126.

[HK] D. P. Huttenlocher, K. Kedem, "Computing the Minimum Hausdorff Distance for Point Sets under Translation", Proc. 6th Annual Symp. on Computational Geometrie, 1990, pp. 340-349.

[M] G. E. Martin, *Transformation Geometry*, Springer-Verlag, New York, Heidelberg, Berlin, 1982.

[S] S. Schirra, "Über die Bitkomplexität der ϵ-Kongruenz", Diplomarbeit, Fachbereich Informatik, Universität des Saarlandes, 1988.

[Y] C. K. Yap, "An $O(n \log n)$ Algorithm for the Voronoi Diagram of a Set of Simple Curve Segments", *Discrete Comp. Geom.* 2, 1987, pp. 365-393.