International Journal of Computational Geometry & Applications

COMPUTING THE HAUSDORFF DISTANCE BETWEEN CURVED OBJECTS*

HELMUT ALT
Freie Universität Berlin, Fachbereich Mathematik und Informatik, Takustr. 9
14195 Berlin, Germany
e-mail:alt@mi.fu-berlin.de

LUDMILA SCHARF
Freie Universität Berlin, Fachbereich Mathematik und Informatik, Takustr. 9
14195 Berlin, Germany
e-mail:[scharf@mi.fu-berlin.de

Received 03 April 2006
Revised 29 August 2006
Communicated by Kokichi Sugihara and Deok-Soo Kim, Guest Editor

ABSTRACT

The Hausdorff distance between two sets of curves is a measure for the similarity of these objects and therefore an interesting feature in shape recognition. If the curves are algebraic computing the Hausdorff distance involves computing the intersection points of the Voronoi edges of the one set with the curves in the other. Since computing the Voronoi diagram of curves is quite difficult we characterize those points algebraically and compute them using the computer algebra system SYNAPS. This paper describes in detail which points have to be considered, by what algebraic equations they are characterized, and how they actually are computed.

Keywords: Hausdorff distance; freeform curves; shape matching.

1. Introduction

Analysis and comparison of geometric shapes are of importance in various application areas within computer science, such as pattern recognition and computer vision, but also in other disciplines concerned with the shape of objects such as cartography, molecular biology, medicine, or biometric signal processing.

The general situation is that we are given two objects A, B modeled as subsets of 2- or 3-dimensional space and we want to know how much they resemble each other [3].

*This research was supported by the European Union under contract No. IST-2000-26473, Project ECG

For this purpose we need a similarity measure defined on pairs of shapes indicating the degree of resemblance of these shapes. A frequently used similarity measure is the Hausdorff distance, which is defined for arbitrary non-empty compact sets A and B. It assigns to each point of one set the distance to its closest point in the other set and takes the maximum over all these values. Formally, we define the *one-sided Hausdorff distance* from A to B as

$$\tilde{\delta}_H(A,B) = \max_{a\in A} \min_{b\in B} d(a,b) \quad ,$$

where $d(a,b)$ denotes a distance measure between points a and b.

Here, we will assume that A and B are planar shapes, i.e., $A, B \subset \mathbb{R}^2$ and that d is the Euclidean distance.

The (bidirectional) *Hausdorff distance* between A and B is defined as maximum over both one-sided distances:

$$\delta_H(A,B) = \max\left\{\tilde{\delta}_H(A,B), \tilde{\delta}_H(B,A)\right\} \quad .$$

Here, we will consider only the computation of the one-sided Hausdorff distance from A to B. On one hand, it is of independent interest since it measures how close A is to some subset of B. On the other hand, we can easily compute the Hausdorff distance itself once we know how to do it for the one-sided Hausdorff distance.

Hausdorff distance is one of the most well studied measures, and efficient algorithms have been developed for different object sets. The Hausdorff distance between two finite point sets of size n and can be computed using Voronoi diagrams in time $O(n \log n)$ [1]. For two polygonal chains or arbitrary sets of n line segments we can compute the Hausdorff distance also in time $O(n \log n)$ using the Voronoi diagram of line segments and a line sweep algorithm [1].

The aim of this paper is to find algorithms for more general shapes which are modeled by two sets of *algebraic curves*. When we speak about the Hausdorff distance between these two sets we actually mean the Hausdorff distance between the two sets of points lying on these curves. We will restrict to curves that are given by *rational parameterizations*, i.e., each curve is represented by a parameterization

$$c : I \to \mathbb{R}^2, \; c(t) = (x(t), y(t)) \quad ,$$

where $I \subset \mathbb{R}$ is a closed interval and x and y are rational functions, which have no poles in the interval I. We assume that the enumerator and denominator of these functions are polynomials whose degree is bounded by a constant. Further, we restrict the input curves to be C^1 continuous, curves with C^1 discontinuities can be preprocessed and split at the corresponding points.

Observe, that this definition includes some important families of free-form parametric curves, in particular B-splines which can be considered as sets of curves with a polynomial parameterization, as well as curves with piecewise rational parameterization. In theory, we can also include Bézier-splines but in this case our algorithms are not feasible, because the degree of the polynomials involved is not bounded.

For simplicity we assume a certain general position of the input curves. In particular, we assume that any two curves intersect in at most finitely many points and any curve intersects the medial axis of another curve or the bisector between two other curves in at most finitely many points. We will describe other special cases that we exclude later. In principle, our methods should work for degenerate inputs, as well, but that would require many case distinctions and technical considerations.

Conceptually, our algorithm is quite straightforward. The difficulty lies in developing algebraic equations for the individual steps and proving their correctness. We implemented the algorithm using the computer algebra system SYNAPS [9] to solve the algebraic equations involved. In fact, the main purpose of this paper is not to develop a competitive software for comparing shapes, but to demonstrate, how this problem can be formulated using an algebraic approach, so that it can be solved using a computer algebra system.

The problem of curve-curve or surface-surface processing has been addressed also in many publications in the CAGD community usually in connection with collision detection or computing intersections. Some also use an algebraic approach, see e.g. [10,11].

2. Basic Cases

The objects we are concerned with are curves, some of which can degenerate to a single point. Before we turn to sets of objects, we will in this section investigate how the directed Hausdorff distance between two single objects can be computed.

2.1. *Point-curve*

The Hausdorff distance from a point $P = (u, v)$ to a curve $c(t) = (x(t), y(t)),\ t \in I$ is by definition $\min_{t \in I} d(P, c(t))$.

In order to find those parameters t where the minimum is attained, we consider the zeros of the derivative of the squared distance $\frac{\mathrm{d}}{\mathrm{d}\,t}\left[d^2(P, c(t))\right]$, i.e., the solutions to the equation

$$2 \cdot (u - x(t)) \cdot x'(t) + 2 \cdot (v - y(t)) \cdot y'(t) = 0 \ . \qquad (1)$$

This equation has constantly many solutions if the degree of c is bounded. From this equation we see that the line segment from the given point P to its nearest neighbor $c(t)$ on the curve c is perpendicular to the tangent vector $(x'(t), y'(t))$ to the curve c at the point $c(t)$, see Figure 1. We will call a point with this property a *footpoint* of P on c.

In addition to the points determined by the solutions of equation (1) the Hausdorff distance from P to c can be attained at the endpoints of c.

We determine the distances from all these candidate points to P and obtain the Hausdorff distance as minimum of all these values.

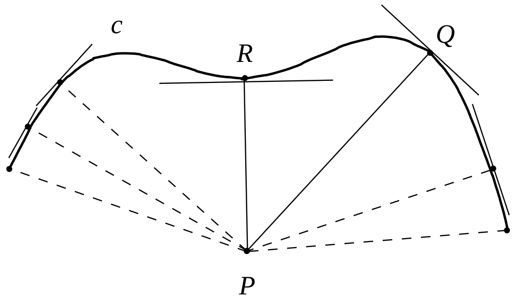

Fig. 1. The Hausdorff distance from a point P to a curve c is the distance to its nearest footpoint R. The Hausdorff distance from a curve c to a point P is the distance between P and its farthest footpoint Q or some endpoint of c.

2.2. *Curve-point*

The Hausdorff distance from a curve c to a point P is by definition the maximum Euclidean distance from any point on the curve to P. So inside c again the points satisfying equation (1), i.e., P's footpoints on c are the candidates and, in addition, the endpoints of c, see Figure 1.

2.3. *Curve-curve*

Like before we reduce the problem of determining the Hausdorff distance from a curve a, $a(t) = (x_a(t), y_a(t))$ to a curve b, $b(s) = (x_b(s), y_b(s))$ to determining the distance of constantly many candidate or critical points on a to the curve b and then taking the maximum over these distances. To characterize all candidate points some theoretical considerations are necessary.

There are four different types of candidate points. Firstly, the Hausdorff distance can be assumed at the endpoints of a, which we call the candidate points of *type EA*, see Figure 2.

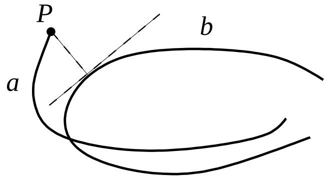

Fig. 2. The Hausdorff distance from the curve a to b occurs between the endpoint P of a and its footpoint on b. P is a candidate point of type EA.

Secondly, it can happen that the Hausdorff distance is attained between an endpoint of b and a point on a, see Figure 3. Therefore, we determine on a all footpoints of the endpoints of b by equation (1) and include them into our list of candidates (*type EB*).

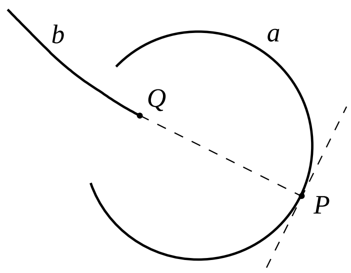

Fig. 3. The Hausdorff distance from the curve a to the curve b is attained between an endpoint Q of b and its footpoint P on a. Point P is a critical point of type EB.

For the third type of candidate points we consider the *medial axis* of a curve, which is a set of all points in the plane whose minimal distance to the curve is attained at more than one point on the curve. As can be seen in Figure 4 the medial axis of a curve segment can consist of several algebraic pieces depending on whether the two closest points are both endpoints, one of them is an endpoint, or both are interior points of the curve. (In the latter case, the medial axis can end in an isolated point, which is the center of an osculating circle touching the curve in a point where the curvature has a local maximum [2].)

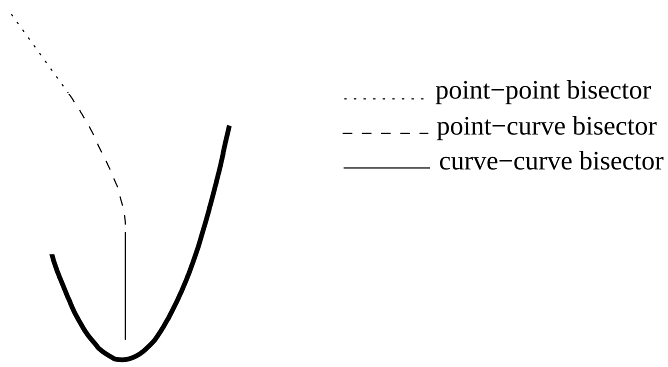

Fig. 4. Medial axis of a parabola segment.

As is demonstrated in Figure 5 an intersection point of the curve a with the medial axis of b can be the point where the Hausdorff distance occurs. Such candidate points of *type MA* can be characterized and determined by a system of algebraic equations.

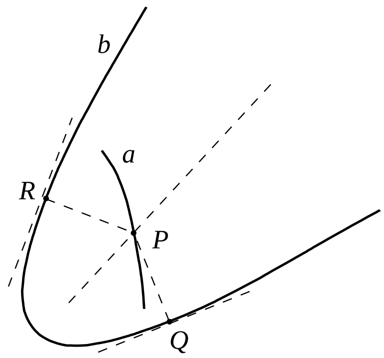

Fig. 5. The Hausdorff distance from the curve a to curve b is attained at an intersection point P of a with the medial axis of the curve b. The point P has two different closest points points Q and R on b. Point P is a critical point of type MA.

We begin with the simpler cases where at least one of the closest points is an endpoint. Let $Q = (u, v)$ and $R = (w, z)$ be the endpoints of the curve b, then an intersection point of the curve a with the bisector between Q and R can be described as a point on a which is equidistant to the both endpoints:

$$[(x_a(t) - u)^2 + (y_a(t) - v)^2] - [(x_a(t) - w)^2 + (y_a(t) - z)^2] = 0 \tag{2}$$

We only have to consider this case if the endpoints do not coincide.

A point of the curve a lying on the bisector between an endpoint of b, say $Q = (u, v)$, and the internal part of b should again be equidistant to Q and its footpoint and, as we observed in section 2.1, the line segment between this point and its footpoint should be perpendicular to the curve b:

$$\begin{aligned} \left[(x_a(t) - x_b(s))^2 + (y_a(t) - y_b(s))^2\right] - \\ \left[(x_a(t) - u)^2 + (y_a(t) - v)^2\right] = 0 \end{aligned} \tag{3}$$

$$(x_a(t) - x_b(s)) \cdot x_b'(s) + (y_a(t) - y_b(s)) \cdot y_b'(s) = 0 \tag{4}$$

Of course, we need to solve this system twice, once for each endpoint of b.

Now, consider the case, where the nearest points to the intersection point of a with the medial axis of b are both interior points. Suppose that $P = a(t)$, and that $Q = b(s)$ and $R = b(r)$ are the points on b closest to P. Then we obtain the following system of equations for parameter values t, s and r:

$$\begin{aligned} \left[(x_a(t) - x_b(s))^2 + (y_a(t) - y_b(s))^2\right] - \\ \left[(x_a(t) - x_b(r))^2 + (y_a(t) - y_b(r))^2\right] = 0 \end{aligned} \tag{5}$$

$$(x_a(t) - x_b(s)) \cdot x_b'(s) + (y_a(t) - y_b(s)) \cdot y_b'(s) = 0 \tag{6}$$

$$(x_a(t) - x_b(r)) \cdot x_b'(r) + (y_a(t) - y_b(r)) \cdot y_b'(r) = 0 \tag{7}$$

Equation (5) says that the distances from P to Q and from P to R should be equal, and equations (6) and (7) say that the two line segments from P to its closest points on b are perpendicular to the curve b (see section 2.1).

The problem we immediately recognize here is that with $s = r$ we can get infinitely many points on a as solutions. This problem can be overcome by introducing a new variable u and adding one more equation to the system to ensure that $s \neq r$ for all solutions:

$$1 - u(s - r) = 0 \tag{8}$$

We add all points defined by the constantly many solutions of this system to the candidate list.

For the fourth type of possible candidate points we observe that the Hausdorff distance can occur between two interior points $P \in a$ and $Q \in b$ without one of them lying on the medial axis of the other curve, see Figure 6.

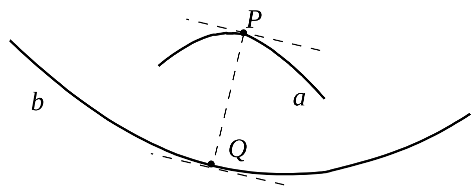

Fig. 6. Hausdorff distance at interior points. P is a critical point of type I.

Since Q is a footpoint on b for point P, the line segment $\overline{PQ}$ must be perpendicular to the tangent line to curve b at point Q. In addition, the tangent line to the curve a at point P must be parallel to the tangent line to the curve b at point Q and, thus, also perpendicular to the line segment $\overline{PQ}$. Therefore, we obtain the following equations for $P = a(t)$ and $Q = b(s)$:

$$(x_a(t) - x_b(s)) \cdot x_b'(s) + (y_a(t) - y_b(s)) \cdot y_b'(s) = 0 \tag{9}$$

$$(x_a(t) - x_b(s)) \cdot x_a'(t) + (y_a(t) - y_b(s)) \cdot y_a'(t) = 0 \tag{10}$$

Candidate points satisfying these equations are called *type I*.

It should be noted that, for simplicity, we exclude inputs where the previous systems of equations are singular, i.e., have infinitely many solutions. This problem can, in fact, occur only for very special inputs. More precisely, equations (3) and (4) become singular if an endpoint of the curve b also lies in the interior of b and a is part of the normal of b in that point. Equations (5)–(8) become singular if b is self-intersecting but has a unique tangent in the intersection point and a is part of the normal in that point, or if the curve b is a circular arc and a contains the center of the corresponding circle. Equations (9) and (10) can become singular if a and b are concentric circular arcs or parallel line segments. Additionally, if one of the curves is a circular arc and an endpoint of the other curve coincides with the center of the circle equation (1) becomes singular when finding candidate points of type EA or EB. We consider these cases, which could be treated by special consideration, as "degenerate inputs" (see introduction).

Now we have characterized all candidate point pairs, where the Hausdorff distance from a curve a to a curve b can be attained. In fact, it holds

Theorem 1. *The directed Hausdorff distance from a curve a to a curve b can only occur at points that are critical of type EA, EB, MA, or I.*

Proof. Consider the function $\sigma : I_a \times I_b \to \mathbb{R}$ with $\sigma(t,s) = (x_a(t) - x_b(s))^2 + (y_a(t) - y_b(s))^2$ which is the squared Euclidean distance between the points $a(t)$ and $b(s)$. σ is continuously differentiable and its graph is a smooth surface Σ in $\mathbb{R}^3$. For an example, see the two curves in Figure 7 and the corresponding function σ in Figure 8 which have been produced with the computer algebra system MAPLE.

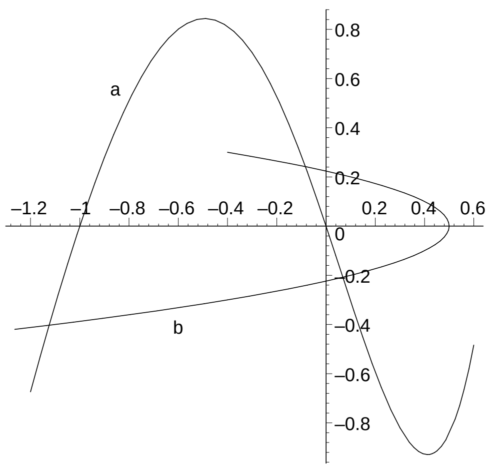

Fig. 7. Two curves

Let $\gamma : I_a \to \mathbb{R}$ be the function that assigns to $t \in I_a$ the distance of the point $a(t)$ to its nearest point on b, i.e., $\gamma(t) = \min_{s \in I_b} \sigma(t,s)$. The graph Γ of γ, therefore, can be obtained by projecting the surface Σ to the (t,z)-plane and taking the lower envelope of this projection, see Figure 9. Here, z is besides s and t the third coordinate axis.

The directed Hausdorff distance from a to b is then $\max_{t \in I_a} \gamma(t)$ and, thus, must be one of the local maxima of $\gamma(t)$. Now the critical points of type EA correspond to the endpoints of I_a. The candidate points of type EB correspond to the maxima

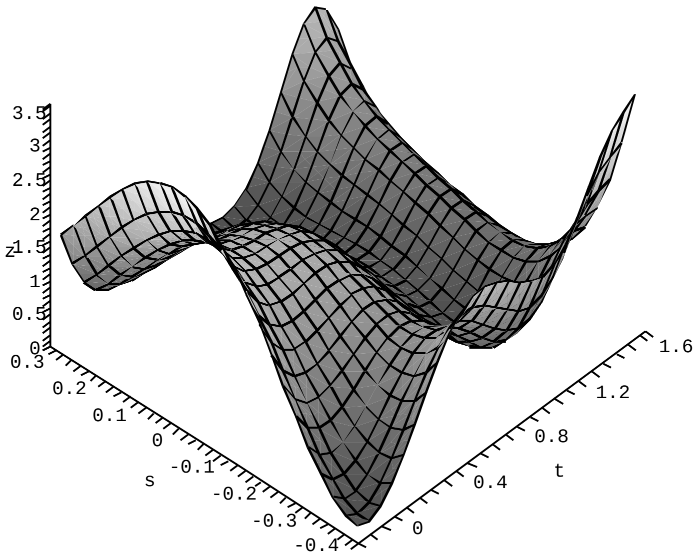

Fig. 8. The graph Σ of σ.

in the portions of Γ that are projections of the boundary of Σ. The candidates of type MA correspond to the points in the lower envelope of the projection resulting from different parts of Σ. It remains to show that the remaining local maxima of γ correspond to points of type I, or equivalently, □

Lemma 1. *Let t_0 be a local maximum of γ such that $a(t_0)$ is not of one of the types EA, EB, or MA. Then the point $p_0 = (t_0, \gamma(t_0)) \in \Gamma$ is the projection of some point $q_0 = (s_0, t_0, \gamma(t_0)) \in \Sigma$ with the property that the partial derivatives $\partial\sigma/\partial t(s_0, t_0) = 0$ and $\partial\sigma/\partial s(s_0, t_0) = 0$.*

Proof. All points of Σ that are projected to Γ and do not lie on the boundary must have the property that the partial derivative of σ in s-direction is zero, since those points correspond to the minima of σ in s-direction. Therefore, $\partial\sigma/\partial s(s_0, t_0) = 0$.

It remains to show the same property for the t-direction. Observe that, since $a(t_0)$ is not a candidate of type MA, it does not lie on the medial axis of b. The same is true for all points in an open neighborhood of $a(t_0)$, so they all have a unique nearest neighbor on b. This means, that there exists some open interval J containing t_0 such that the segment $\{(t, \gamma(t)) | t \in J\}$ of Γ has a continuous curve

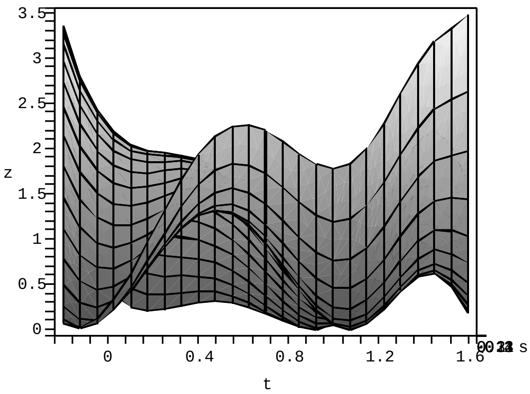

Fig. 9. The graph Γ of γ is the lower envelope of the projected image. Type EA parameters are the two endpoints,a type EB parameter is (approximately) $t = 0.5$, type MA parameters are $t = 0.05$ and $t = 1$, and a type I parameter is $t = 1.45$.

segment κ on Σ as its preimage under the projection from Σ to the (t, z)-plane. Furthermore, κ contains q_0.

Suppose that $\partial\sigma/\partial t(s_0, t_0) \neq 0$. Then the tangent plane T to Σ in q_0 is perpendicular to the (z, t)-plane but not parallel to the (s, t)-plane. Consider two planes T_1 and T_2 that are also perpendicular to the (z, t)-plane and pass through q_0. In addition, they lie on opposite sides of T and separate T from the plane T_0 through q_0 parallel to the (s, t)-plane (see Figure 10). Then there must be some open neighborhood U of q_0 so that inside U, Σ lies completely in the wedge W between T and T'. Also the curve segment $\kappa \cap U$ which maps to some open segment of Γ that contains p_0, must lie in the wedge W which is a contradiction to p_0 being a local maximum on Γ.

This proves Lemma 1. In fact, q_0 is an interior saddle point of Σ. Observe that the partial derivatives of σ being 0 is equivalent to equations (9) and (10), so $a(t_0)$ is a candidate point of type I, which proves Theorem 1. $\square$

3. The General Case

As we said before, the general problem we consider is to find the Hausdorff distance between two point sets given by two sets A and B of rationally parameterized algebraic curves.

In order to find $\tilde{\delta}_H(A, B)$ we first consider the *Voronoi diagram* of the different

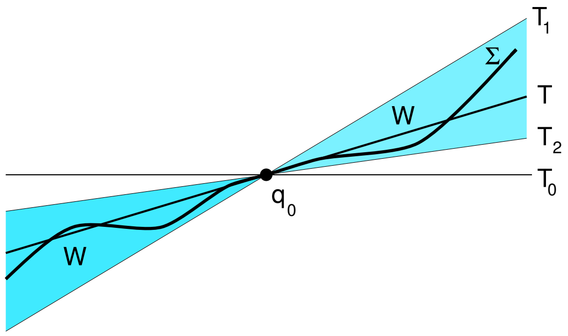

Fig. 10. Cross section at $s = s_0$ showing the wedge W that locally must contain the surface Σ.

curves in B. Obviously, the Hausdorff distance from A to B can be attained at intersection points of A with the Voronoi edges of B. Therefore, we simply split the curves of A at these intersection points with the Voronoi edges of B. The resulting set A' of curves has the property that each curve lies in one Voronoi cell of B, so its distance to B is the distance to one curve and can be obtained using the techniques of section 2.

Computing the complete Voronoi diagram of algebraic curves is a very difficult task in practice and there are many open questions about the bisectors of algebraic curves [4,5,6,7,8]. For example, the bisector of two curves with rational parameterization does not have a rational parameterization in general. In [7] an approximation algorithm for Voronoi diagrams of curves is given.

Instead of constructing the complete diagram, we just compute the intersection points described. The splitting of each curve a in A is done incrementally. Suppose that we have a list of the intersection points of a with a Voronoi diagram of a subset of B and we want to add a new curve $b \in B$. We do this by scanning the current segments of a. Each segment s of a curve a lies in a Voronoi cell of some curve $c \in B$ which has already been processed. First we determine all intersection points of the bisector between b and c with a and find out which ones lie inside s. s is split further with these points and some portions are labeled with b as nearest neighbor, others with c. After this has been done it might be necessary to merge neighboring segments which are both marked with b but are separated by a split-point from a previous step.

Similar to the case of the medial axis, the intersection points of a with the bisector between b and c can be found as solutions of systems of equations. We only give this system here for the "interior" bisector of two curves b and c, since the equation for the bisector between two endpoints is exactly the equation (2) with corresponding point coordinates and the system of equations (3) and (4) is used for a bisector between an endpoint of one curve and the interior part of another curve, as described in section 2.3.

The system for the "interior" bisector uses the property that if a point $a(t)$ lies on the bisector between b and c and the corresponding footpoints are $b(s)$ and $c(r)$, then the line segment $\overline{a(t)b(s)}$ is perpendicular to the tangent vector $b'(s)$, and $\overline{a(t)c(r)}$ is perpendicular to $c'(r)$:

$$\begin{aligned}&\left[(x_a(t) - x_b(s))^2 + (y_a(t) - y_b(s))^2\right] - \\ &\qquad \left[(x_a(t) - x_c(r))^2 + (y_a(t) - y_c(r))^2\right] = 0\end{aligned} \tag{11}$$

$$(x_a(t) - x_b(s)) \cdot x_b'(s) + (y_a(t) - y_b(s)) \cdot y_b'(s) = 0 \tag{12}$$

$$(x_a(t) - x_c(r)) \cdot x_c'(r) + (y_a(t) - y_c(r)) \cdot y_c'(r) = 0 \tag{13}$$

Observe that the worst case running time of our algorithm is $O(nm^2)$ if A consists of n and B of m curves, assuming that the degree of curves are bounded and, therefore, the systems of equations can be solved in constant time. It should be worthwhile to improve the combinatorial complexity of the algorithm (see also [2]), but our major intent was to find a simple algorithm that can be put into practice with a reasonable amount of effort.

4. Implementation

We implemented the algorithm described in C++ using computer algebra library SYNAPS (SYmbolic Numeric APlicationS), see `http://www-sop.inria.fr/galaad/logiciels/synaps/` and [9], for solving the systems of polynomial equations.

Table 1 shows computation time statistics for sets of curves with polynomial parameterization, with polynomials of degree 2, 3 and 4. Time measurements were performed on a Pentium 4 3GHz Linux machine. A significant amount of the computation time (ca. 95% on average) thereby is spent by the computer algebra system and the figures in Table 1 are not really convincing for practical applications. However, as was mentioned before, the major purpose of the implementation is to demonstrate that our algebraic concept can be realized *in principle*. Furthermore, Table 1 shows the polynomial (closer to quadratic than to cubic) dependence of the runtime on the number of curves, and its exponential dependence on the degree of the curves.

For visualization purposes a graphical user interface was developed using the GTK library. It includes the visualization of the curves, the input and editing of the parameterization and the intervals of the parameter values, the computation of the Hausdorff distance and the graphical indication of the candidate points considered for the computation. Figure 11 shows the interface with an example.

5. Conclusion

In this paper we presented an algorithm based on symbolic methods for computing the Hausdorff distance between two sets of planar freeform curves A and B. Conceptually, it exploits the Voronoi diagram of the curves in B, but avoids direct

set	polynomial degree		
size	2	3	4
3	0.949	5.311	20.852
5	3.159	16.677	61.888
10	13.486	79.557	299.284
15	34.550	192.330	730.295
20	59.222	644.815	
25	93.263		
30	127.175		

Table 1. Computation time in seconds for sets of B-spline curves.

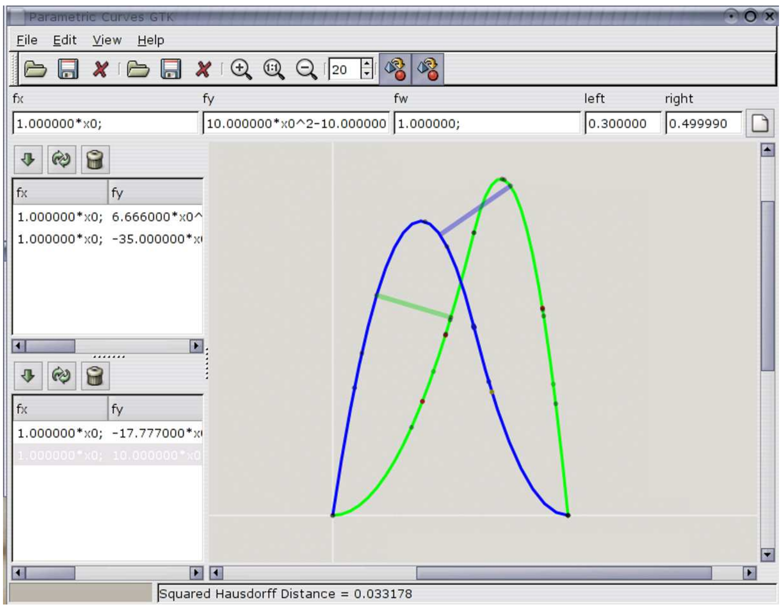

Fig. 11. Result of the Hausdorff distance computation with two B-splines of degree 2. Both directed distances and all candidate points are shown.

computation of the Voronoi diagram since that is not a trivial task. Instead, we introduce criteria for partitioning each curve in A into the pieces resulting from the intersections with the Voronoi regions of the curves in B. We also define algebraic constraints for the interior points of the curves where the Hausdorff distance can be attained. These points together with the points of intersection with the Voronoi diagram and the original endpoints of curves in A form a set of so-called candidate points. For every candidate point we compute its distance to the set B and take the maximum over all these distances.

The implementation of the algorithm was intended as a demonstration that the concept can be realized. For a speed up in real applications it might be reasonable to combine our algebraic methods with numeric approximation techniques.

References

1. H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, 13:251–265, 1995.
2. H. Alt, O. Cheong, and A. Vigneron. The voronoi diagram of curved objects. *Discrete and Computational Geometry*, 34:439–453, 2005.
3. H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.-R. Sack and J. Urrutia, editors, *Handbook of computational geometry*. Elsiever Science Publishers B.V. North-Holland, Amsterdam, 1999.
4. G. Elber and M.-S. Kim. Bisector curves of planar rational curves. *Computer-Aided Design*, 30(14):1089–1096, 1998.
5. G. Elber and M.-S. Kim. Computing rational bisectors. *IEEE Computer Graphics and Applications*, pages 76–81, November/December 1999.
6. R. T. Farouki and R. Ramamurthy. Degenerate point/curve and curve/curve bisectors arising in medial axis computations for planar domains with curved boundaries. *Internat. J. Comput. Geom. Appl.*, 8:599–617, 1998.
7. R. Ramamurthy and R. T. Farouki. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries i. theoretical foundations. *Journal of Computational and Applied Mathematics*, 102(1):119–141, February 1999.
8. R. Ramamurthy and R. T. Farouki. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries ii. detailed algorithm description. *Journal of Computational and Applied Mathematics*, 102(1):253–277, 1999.
9. G. D. Reis, B. Mourrain, R. Rouillier, and Ph. Trébuchet. An environment for symbolic and numeric computation. In *Proc. of the International Conference on Mathematical Software*, pages 239–249, 2002.
10. T. W. Sederberg, H. N. Christiansen, and S. Katz. Improved test for closed loops in surface intersections. *Comput. Aided Des.*, 21(10):505–508, 1989.
11. J.-K. Seong, D. E. Johnson, and E. Cohen. A higher dimensional formulation for robust and interactive distance queries. In *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 197–205, New York, NY, USA, 2006. ACM Press.