

Discrete Curve Evolution with Hausdorff Distance

Roland Opfer & Longin Ian Latecki

Abstract

In this paper we present a discrete curve evolution with a new knot relevance measure. This new measure has the advantage, that we can determine the Hausdorff distance between the original and the evolved curve without additional computation. The basic idea is, to parametrize the curve by two linear spline functions. We calculate a good approximation of these two spline functions, by spline functions with fewer knots. This approximation is the parametric representation of our evolved curve. For the purpose of approximating spline function by spline function with fewer knots we apply a knot removal strategy for linear spline functions, which we present in the first section of this paper. This knot removal strategy is formulated in very general terms, so that it could also be applied to spline functions of an arbitrary order and even to radial basis functions in the multivariate case.

1 General strategy for knot removal

In [Latecki 1998, Latecki 1999, Latecki 2000] a greedy style algorithm for a discrete curve evolution is presented. This algorithm works astonishing well and leads to a noise free representation of the curve as well as to a good shape abstraction. In [Latecki 2000] also some theoretical properties of this curve evolution are proved. However, up to now it was a major problem to find good criteria for terminating the evolution. In this paper we present a similar curve evolution as in [Latecki 1998], but with a different knot relevance measure. This new measure has the advantage, that we can determine the Hausdorff distance between the original and the evolved curve without additional computation.

We begin with the definition of a function space, which depends in a certain way on some given points in \mathbb{R}^n . We give two examples of such spaces. The second one is introduced in the second section. We need this second example of a data dependent space to establish our new, so called curve evolution.

Definition 1.1 Let Ω be a subset of \mathbb{R}^n and $X := \{x_1, x_2, \dots, x_m\}$ a set of pairwise distinct points in Ω . We shall call a finite dimensional subspace of $\{f : \Omega \rightarrow \mathbb{R}\}$ a *data dependent space* and denote it by \mathbb{S}_X if

- (a) for every subset \tilde{X} of X , $\mathbb{S}_{\tilde{X}}$ is a vector subspace of \mathbb{S}_X ,
- (b) \mathbb{S}_X depends on the data set X ,
- (c) \mathbb{S}_X is the null space iff $X = \emptyset$.

The elements of X are also called *knots*. In many applications problems arise, where one has a function f of a data dependent space and one wants to find an approximation of f depending on a smaller data set. In this section we present a very general knot removal algorithm for functions of a data dependent space, which can solve such problems.

First of all we want to give an examples of a data dependent space. In the next section we will work with a space of piecewise linear functions, which is also a data dependent space. For the first example of a data dependent space we need an extra definition.

Definition 1.2 Let Ω be a subset of \mathbb{R}^n . A function $\Phi : \Omega \times \Omega \rightarrow \mathbb{R}$ is called *positive definite with respect to Ω* if for all choices of finite sets $X := \{x_1, x_2, \dots, x_m\}$ of pairwise distinct points in Ω the matrix

$$\mathbf{A} := (\Phi(x_i, x_j))_{i,j}, \quad i, j = 1, 2, \dots, m, \quad m \in \mathbb{N}$$

is positive definite.

The most common positive definite functions have the form $\Phi(x, y) = \phi(\|x - y\|)$, where ϕ is a function which maps \mathbb{R}_+ to \mathbb{R} and $\|\cdot\|$ denotes the ordinary Euclidean norm. This type of positive definite function is also called *radial basis function* and some frequently used examples of these functions are $\phi(t) := e^{-\alpha t^2}$, $\alpha > 0$ and $\phi(t) := (t^2 + \alpha)^{-\frac{1}{2}}$, $\alpha > 0$. These functions have the astonishing property that for arbitrarily many pairwise distinct points $x_i \in \mathbb{R}^n$ the matrix $\phi(\|x_i - x_j\|)_{i,j}$ is positive definite, *independent* of the space dimension n . The Function $\phi(t) := (t^2 + \alpha)^{-\frac{1}{2}}$ leads to the *inverse multiquadrics* interpolation, which is often used in geophysics. Also very useful in this context are *conditionally positive definite* functions like $\phi(t) := \text{identity}$ or $\phi(t) := (t + 1)^{-1}$ were the above defined matrix \mathbf{A} is only positive definite on a certain subspace of \mathbb{R}^m . Readers who want to obtain some more information about positive definite or radial basis functions should read [Cheney 2000] and [Schaback 1997].

Definition 1.3 Let the function Φ be positive definite with respect to some domain $\Omega \subset \mathbb{R}^n$ and let $X := \{x_1, x_2, \dots, x_m\}$ be a set of pairwise distinct points in Ω . Then we introduce the following notation:

$$\mathbb{S}_{\Phi, X} := \{f : \Omega \rightarrow \mathbb{R} : f(x) := \sum_{i=1}^m \alpha_i \Phi(x_i, x), \quad \alpha_i \in \mathbb{R}\}.$$

Obviously, $\mathbb{S}_{\Phi, X}$ is an example of a data dependent space according to definition 1.1, because (a) $\mathbb{S}_{\Phi, X}$ is a finite dimensional vector space, (b) $\mathbb{S}_{\Phi, X}$ depends on the data X , (c) $\mathbb{S}_{\Phi, \tilde{X}}$ is a subspace of $\mathbb{S}_{\Phi, X}$ if \tilde{X} is a subset of X , (d) $\mathbb{S}_{\Phi, X}$ is the null space iff $X = \emptyset$. Note, that the interpolation problem, to find an $f \in \mathbb{S}_{\Phi, X}$ which satisfies $f(x_i) = c_i$ for some given set of real numbers c_i , has always a unique solution, because the function Φ is positive definite.

If M is any finite set, we denote by $|M|$ the number of its elements. Now we consider the following problem: For a given data dependent space \mathbb{S}_X , let $\|\cdot\|$ be an arbitrary norm on \mathbb{S}_X . For a given tolerance $\varepsilon > 0$ and a given $f \in \mathbb{S}_X$ with $\|f\| \geq \varepsilon$, we try to find a subset \tilde{X} of X and a function $\tilde{f} \in \mathbb{S}_{\tilde{X}}$ so that the following two properties hold:

- (1) $\|f - \tilde{f}\| = \min_{\tilde{g} \in \mathbb{S}_{\tilde{X}}} \|f - \tilde{g}\| < \varepsilon,$
- (2) $\min_{g \in \mathbb{S}_Y} \|f - g\| \geq \varepsilon$ for all $Y \subset X$ with $|Y| < |\tilde{X}|.$

Definition 1.4 We call the just stated problem the *knot approximation problem* for the function $f \in \mathbb{S}_X$.

Theorem 1.5 The knot approximation problem always has a solution.

Proof: For every fixed subset \tilde{X} of X there is an element $\tilde{f} \in \mathbb{S}_{\tilde{X}}$ with

$$\|f - \tilde{f}\| = \min_{\tilde{g} \in \mathbb{S}_{\tilde{X}}} \|f - \tilde{g}\|,$$

since $\mathbb{S}_{\tilde{X}}$ is a finite dimensional vector space and $\varrho(\tilde{g}) := \|f - \tilde{g}\|$ is a continuous function. Let \mathcal{X} the set of all subsets of X , so that for each $\tilde{X} \in \mathcal{X}$ we have $\min_{\tilde{g} \in \mathbb{S}_{\tilde{X}}} \|f - \tilde{g}\| < \varepsilon$. The set \mathcal{X} is not empty because a least X is in \mathcal{X} . Since X has only finite many points, \mathcal{X} has finite many elements. We choose a smallest set \tilde{X} of \mathcal{X} and a $\tilde{f} \in \mathbb{S}_{\tilde{X}}$ that satisfies $\|f - \tilde{f}\| = \min_{\tilde{g} \in \mathbb{S}_{\tilde{X}}} \|f - \tilde{g}\|$. The set \tilde{X} contains a least one point because $\|f\| \geq \varepsilon$. Therefore \tilde{X} and \tilde{f} satisfy (1) and (2). \square

In general, it is very hard to compute an exact solution of the knot approximation problem, in particular, if the data set X is large. We now present a greedy algorithm, which can deliver a fairly good approximation of the true solution. The strategy of the presented algorithm is very similar to the strategy of the discrete curve evolution first presented in [Latecki 1998]. For each point in X we calculate a *relevance measure*,

which gives us some information about the importance of the point for the function $f \in \mathbb{S}_X$. We remove in each step the point with the lowest relevance. With $\mathbf{G}(X, \tilde{X}) : \mathbb{S}_X \rightarrow \mathbb{S}_{\tilde{X}}$ we denote an approximation operator, which maps for every subset \tilde{X} a function $f \in \mathbb{S}_X$ to an approximation $\mathbf{G}(X, \tilde{X})f \in \mathbb{S}_{\tilde{X}}$ of f . The approximation operator \mathbf{G} is arbitrary, for instance \mathbf{G} could be an interpolation operator or a mapping to the best approximation, if it is unique. In the next section we define a \mathbf{G} which is suitable for our purpose. If X is a given data set and f a function of \mathbb{S}_X , then we calculate the *relevance measure* of the point $x \in X$ according to the formula

$$(3) \quad \omega := \|f - \mathbf{G}(X, X \setminus \{x\})f\|.$$

The real number ω measures, how good we can approximate the function f without the point x . We now describe the new algorithm.

Algorithm 1.6 Given finite set X , a function $f \in \mathbb{S}_X$, a tolerance $\varepsilon > 0$ and $l \in \mathbb{N}$ with $l < |X|$.

0. Start: $X^0 := X$, $f^0 := f$, $j := 0$ and $\text{test}^0 := 0$.
1. If $|X^j| = l$ then stop;
2. determine the relevance measure of all points in X^j , let $x(j)$ be a point in X^j with the lowest relevance $\omega(j) := \|f^j - \mathbf{G}(X^j, X^j \setminus \{x(j)\})f^j\|$;
3. if $\text{test}^j + \omega(j) < \varepsilon$ go to 4, else stop;
4. put

$$\begin{aligned} f^{j+1} &:= \mathbf{G}(X^j, X^j \setminus \{x(j)\})f^j, \\ X^{j+1} &:= X^j \setminus \{x(j)\}, \\ \text{test}^{j+1} &:= \text{test}^j + \omega(j), \\ j &:= j + 1, \end{aligned}$$

and go to 1;

Theorem 1.7 *The given algorithm 1.6 terminates after a finite number of, say j_1 , iterations. We obtain a sequence of subsets of X with $X^{j_1} \subset X^{j_1-1} \subset X^{j_1-2} \subset \dots \subset X^0$ and in every step $j, 0 \leq j \leq j_1$ the function $f^j \in \mathbb{S}_{X^j}$ satisfies the estimation*

$$\|f^j - f^0\| < \varepsilon.$$

Proof: In each executed step 4 the algorithm sets $X^{j+1} := X^j \setminus \{x(j)\}$, so it has the claimed inclusion relation, and in addition, we obtain $|X^j| = |X| - j$. Therefore, the algorithm must terminate, either because of $|X^j| = l$ or because of $\text{test}^j + \omega(j) \geq \varepsilon$. Finally, for $0 \leq j \leq j_1$ we obtain the estimation

$$\begin{aligned} \|f^j - f^0\| &= \|f^j - f^{j-1} + f^{j-1} - \dots - f^1 + f^1 - f^0\| \\ &\leq \|f^j - f^{j-1}\| + \dots + \|f^1 - f^0\| \\ &= \|\mathbf{G}(X^{j-1}, X^{j-1} \setminus \{x(j-1)\})f^{j-1} - f^{j-1}\| + \dots \\ &\quad \dots + \|\mathbf{G}(X^0, X^0 \setminus \{x(0)\})f^0 - f^0\| \\ &= \omega(j-1) + \dots + \omega(0) \\ &= \omega(j-1) + \text{test}^{j-1} \\ &< \varepsilon. \end{aligned} \quad \square$$

Given a set X , a function $f \in \mathbb{S}_X$, a tolerance $\varepsilon > 0$, and an integer l the algorithm 1.6 delivers a subset \tilde{X} of X , and an approximation $\tilde{f} \in \mathbb{S}_{\tilde{X}}$ with $\|\tilde{f} - f\| < \varepsilon$. We shall denote this input-output relation

by $(\tilde{X}, \tilde{f}) = \mathbf{ALG}(X, f, \varepsilon, l)$. Generally, the sum of the relevance measures is a very bad upper bound for the true approximation class of \tilde{f} to f and $\|\tilde{f} - f\|$ is much smaller than ε . To improve the result of algorithm 1.6 we add a “global” approximation step. In algorithm 1.8 we describe the improved version of algorithm 1.6. One could summarize algorithm 1.8 as follows: We remove successively the points with the lowest relevance, until the sum of relevances exceeds the given tolerance ε . One could stop now with removing points, and one would obtain a function f^j which satisfies $\|f - f^j\| < \varepsilon$. But generally, the sum of the relevance measures is a too bad upper bound for the true approximation class of f^j to f . Therefore, we compute in step two algorithm 1.8 a better approximation by applying the approximation operator \mathbf{G} to f^0 with the computed point set X^j . If $\|f^0 - \mathbf{G}(X, X^j)f^0\|$ does not exceed the given tolerance, we can continue removing knots in step one of algorithm 1.8. The algorithm stops when in step one no knot has been removed (then we have $X^j = X^{j-1}$).

Algorithm 1.8 Given a finite set X , a function $f \in \mathbb{S}_X$, a tolerance $\varepsilon > 0$, and $l \in \mathbb{N}$ with $l < |X|$.

0. Start: $X^0 := X$, $f^0 := f$, $j := 0$ and $\text{test}^0 := 0$;
1. $(X^{j+1}, f^{j+1}) := \mathbf{ALG}(X^j, f^j, \varepsilon - \text{test}^j, l)$;
2. if $\|f^0 - \mathbf{G}(X, X^{j+1})f^0\| < \varepsilon$, $|X^{j+1}| \neq |X^j|$ and $|X^{j+1}| \neq l$, put

$$\begin{aligned} f^{j+2} &:= \mathbf{G}(X, X^{j+1})f^0, \\ \text{test}^{j+2} &:= \|f^0 - \mathbf{G}(X, X^{j+1})f^0\|, \\ X^{j+2} &= X^{j+1} \\ j &:= j + 2, \end{aligned}$$

and go to 1, else stop.

Theorem 1.9 *The given algorithm 1.8 terminates after a finite number of, say j_1 , iterations. We obtain a sequence of subsets of X with $X^{j_1} \subset X^{j_1-1} \subset X^{j_1-2} \subset \dots \subset X^0$ and in every step j , $0 \leq j \leq j_1$ the function $f^j \in \mathbb{S}_{X^j}$ satisfies the estimation*

$$\|f^j - f^0\| < \varepsilon.$$

Proof: Due the conditions in step two, the algorithm terminates after a finite number of iteration. For all $0 \leq j \leq j_1$ we obtain the estimation

$$\begin{aligned} \|f^j - f^0\| &= \|f^j - f^{j-1} + f^{j-1} - f^0\| \\ &\leq \|f^j - f^{j-1}\| + \|f^{j-1} - f^0\| \\ &= \underbrace{\|f^j - f^{j-1}\|}_{< \varepsilon - \text{test}^{j-1}} + \underbrace{\|\mathbf{G}(X, X^{j-2})f^0 - f^0\|}_{=\text{test}^{j-1}} \\ &< \varepsilon. \end{aligned} \quad \square$$

In the next section we do not need this algorithm in this general setting. So we specialize it to one dimensional periodic linear spline functions.

A slightly different strategy for the multivariate case is tested in [Opfer 2002] with the radial basis function $\Phi(x, y) := \|x - y\|$.

In [Lyche 1987] and [Lyche 1988] a different knot removal strategy for B-spline functions is presented. In [Lyche 1986] and [Böhm 1980] you can find some very fast algorithm for approximating B-spline functions by B-spline function defined on a different but fixed knot sequence.

2 Approximation of piecewise linear curves

In this section we present the new discrete curve evolution. A great amount of experiments has shown, that the new curve evolution has the same nice properties as the evolution presented in [Latecki 1998]. The advantage of the new evolution is, that we can determine an upper bound for the Hausdorff distance between the original and evolved curve without additional computation. With the Hausdorff distance we have a good termination criteria at hand, so we can stop the evolution if the Hausdorff distance exceeds a given tolerance. The basic idea is, to parametrize the curve by two functions of the space $\mathbb{S}_{\mathbf{t}}^p$ (defined below). By applying the knot removal algorithm to these functions we obtain a good approximation by spline functions with fewer knots. This approximation is the parametric representation of our evolved curve.

In order to present our new, so called curve evolution in this section, we now introduce the next example of a data dependent space.

Definition 2.1 Let $\mathbf{t} := \{t_1 < t_2 < \dots < t_n\}$ be a given real knot sequence and p a real number with $p > t_m - t_1$. We call

$$\mathbb{S}_{\mathbf{t}}^p := \{s \in C(\mathbb{R}) : s|_{[t_i, t_{i+1}]}, i = 1, \dots, m-1 \text{ and } s|_{[t_m, t_1+p]} \text{ are linear, } s(t) = s(t+p) \forall t \in \mathbb{R}\}$$

space of *piecewise linear p -periodic functions with respect to the knot sequence \mathbf{t}* . If $\mathbf{t} = \emptyset$ then $\mathbb{S}_{\mathbf{t}}^p$ is the null function. In case $m = 1$, the space $\mathbb{S}_{\mathbf{t}}^p$ consists of the constant functions.

If one uses spline spaces of an higher order, it is very useful to work with B-spline bases. See [Nürnberger 1980, Schumaker 1981, de Boor 1978] to obtain informations about spline spaces, for example.

If τ is a subsequence of \mathbf{t} then obviously \mathbb{S}_{τ}^p is a subspace of $\mathbb{S}_{\mathbf{t}}^p$. So we obtain that for a fixed real number p , the set $\mathbb{S}_{\mathbf{t}}^p$ is also a data dependent space according to definition 1.1.

Let $P := \{p_1, p_2, \dots, p_n, p_{n+1}\}$, $p_i \in \mathbb{R}^2$ be the vertices of a closed (that means $p_1 = p_{n+1}$) piecewise linear curve in \mathbb{R}^2 . With $\text{trace}(P)$ we mean the curve, which one obtains by connecting p_i with p_{i+1} by a straight line, while the set P are the vertices, which represent the curve $\text{trace}(P)$. We always assume that P has length one, if not so we rescale P such that

$$(4) \quad \sum_{i=1}^n \|p_i - p_{i+1}\|_2 = 1.$$

Now we define a knot sequence $\mathbf{t} := \{t_1 = 0 < t_2 < \dots < t_n\}$ by the property

$$(5) \quad |t_i - t_{i+1}| = \|p_i - p_{i+1}\|_2 \text{ for } i = 1, 2, \dots, n-1.$$

Doing this, we can choose two functions $x, y, \in \mathbb{S}_{\mathbf{t}}^p$ with $p = 1$ such that

$$(x(t_i), y(t_i)) = p_i \text{ for } i = 1, \dots, n \text{ and} \\ \text{trace}(P) = \text{trace}(x, y).$$

($\text{trace}(x, y) := \{(x(t), y(t)) \in \mathbb{R}^2 : t \in \mathbb{R}\}$)

According to (4) and (5), (x, y) is an arc-length parametrization of the curve $\text{trace}(P)$. Now we want to apply the knot removal algorithm to the functions x, y , so we have to specialize the norm and the approximation operator \mathbf{G} used in the upper description of the algorithm. Our approximation should be done in the standard max-norm

$$\|x\|_{\infty} := \max_{t \in \mathbb{R}} |x(t)|.$$

The evolution should have the property that in every step the evolved curve \tilde{P} is a subset of P . For that reason, we define the following interpolation operator. If f is a function of $\mathbb{S}_{\mathbf{t}}^p$ and τ is a subset of \mathbf{t} , then $\mathbf{G}(\mathbf{t}, \tau) : \mathbb{S}_{\mathbf{t}}^p \rightarrow \mathbb{S}_{\tau}^p$ is defined by the property: For every knot

$$(6) \quad \tau_0 \in \tau \text{ we have } \mathbf{G}(\mathbf{t}, \tau)f(\tau_0) = f(\tau_0)$$

Sine $f \in \mathbb{S}_t^p$ is piecewise linear, we just have to store the values of f in the knots of \mathbf{t} to describe f completely. So, if we want to compute the interpolation $\mathbf{G}(\mathbf{t}, \tau)f$, we just drop the knots of \mathbf{t} which are not in τ . The norm of $\mathbf{G}(\mathbf{t}, \tau)f - f$ can also be computed in a very easy way:

$$\|\mathbf{G}(\mathbf{t}, \tau)f - f\|_\infty = \max_{\tau \in \mathcal{T}} |\mathbf{G}(\mathbf{t}, \tau)f(\tau) - f(\tau)|.$$

If we applied the knot removal algorithm to x and y separately, it could happen that after a few steps x^j and y^j are defined on some different knot sequences. For avoiding that, we remove points of the knot sequence of x and y only simultaneously. Therefore, we modify the knot removal algorithm as follows: Let x, y be two functions of \mathbb{S}_t^p . For every knot $t \in \mathbf{t}$ we define

$$(7) \quad \omega := \max\{\|x - \mathbf{G}(\mathbf{t}, \mathbf{t} \setminus \{t\})x\|_\infty, \|y - \mathbf{G}(\mathbf{t}, \mathbf{t} \setminus \{t\})y\|_\infty\}.$$

The real number ω measures how good we can approximate the functions *simultaneously* without the knot t . We now describe the modified version of algorithm 1.8.

Algorithm 2.2 Given knot sequence \mathbf{t} , two functions $x, y \in \mathbb{S}_t^1$, and a tolerance $\varepsilon > 0$.

0. Start: $\tau^{-1} := \tau^0 := \mathbf{t}$, $x^0 := x, y^0 := y$, $j := 0$ and $\text{test}^0 := 0$.
1. If $|\tau^j| = 3$ then stop;
2. determine the relevance measure of all knots in τ^j , let $\tau(j)$ be a knot from τ^j with the lowest relevance $\omega(j) := \max\{\|x^j - \mathbf{G}(\tau^j, \tau^j \setminus \{\tau(j)\})x^j\|_\infty, \|y^j - \mathbf{G}(\tau^j, \tau^j \setminus \{\tau(j)\})y^j\|_\infty\}$;
3. if $\text{test}^j + \omega(j) < \varepsilon$ go to 4, else go to 5;
4. put

$$\begin{aligned} x^{j+1} &:= \mathbf{G}(\tau^j, \tau^j \setminus \{\tau(j)\})x^j, \\ y^{j+1} &:= \mathbf{G}(\tau^j, \tau^j \setminus \{\tau(j)\})y^j, \\ \tau^{j+1} &:= \tau^j \setminus \{\tau(j)\}, \\ \text{test}^{j+1} &:= \text{test}^j + \omega(j), \\ j &:= j + 1, \text{ and go to 1;} \end{aligned}$$

5. if $\tau^{j-1} \neq \tau^j$, put

$$\begin{aligned} x^{j+1} &:= \mathbf{G}(\mathbf{t}, \tau^j)x, \\ y^{j+1} &:= \mathbf{G}(\mathbf{t}, \tau^j)y, \\ \tau^{j+1} &:= \tau^j, \\ \text{test}^{j+1} &:= \max\{\|x - \mathbf{G}(\mathbf{t}, \tau^j)x\|_\infty, \|y - \mathbf{G}(\mathbf{t}, \tau^j)y\|_\infty\}, \text{ and go to 1;} \end{aligned}$$

else stop.

Theorem 2.3 The given algorithm 2.2 terminates after a finite number of, say j_1 , iterations. We obtain a sequence of subsets of τ with $\tau^{j_1} \subset \tau^{j_1-1} \subset \tau^{j_1-2} \subset \dots \subset \tau^0$ and in every step $j, 0 \leq j \leq j_1$ the functions $x^j \in \mathbb{S}_{\tau^j}^p, y^j \in \mathbb{S}_{\tau^j}^p$ satisfy the estimation

$$\|x^j - x\|_\infty < \varepsilon \quad \text{and} \quad \|y^j - y\|_\infty < \varepsilon.$$

Proof: Same arguments as in theorem 1.9. □

As mentioned (x, y) is an arc-length parametrization of the curve P . Due to the interpolation property (6) of the approximation operator \mathbf{G} we have, that in each step j and for all $\tau \in \tau^j$

$$(x^j(\tau), y^j(\tau)) \in P.$$

We now want to obtain some information about the Hausdorff distance between $\text{trace}(x, y) =: P$ and $\text{trace}(x^j, y^j) =: P^j$. For readers who are not familiar with the notion of Hausdorff distance, we give the definition here for convenience. We need few steps.

Definition 2.4 Let B be a non empty, convex, bounded and point symmetric (that means $x \in B \Rightarrow -x \in B$) subset of \mathbb{R}^2 . Let $x_0 \in \mathbb{R}^2$ and $\delta > 0$ be given. We shall call

$$B_\delta(x_0) := \{x \in \mathbb{R}^2 : \exists \tilde{x} \in B \text{ with } x = x_0 + \delta \tilde{x}\}$$

the *generalized ball with radius δ and center x_0* .

For a subset $S \subset \mathbb{R}^2$ and $\delta > 0$ we define the *Hausdorff parallel set* by

$$S_\delta := \bigcup_{x \in S} B_\delta(x).$$

With the help of the Hausdorff parallel set we can construct a metric on the set of all non empty, compact subsets of \mathbb{R}^2 . Let \mathcal{C} be the family of all non empty, compact subsets of \mathbb{R}^2 . Then the function $\delta : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}_+$ defined by

$$\delta(S_1, S_2) := \inf\{\delta > 0 : S_1 \subset (S_2)_\delta, S_2 \subset (S_1)_\delta\}$$

is a metric on \mathcal{C} . A proof can be found in [Hausdorff 1927] Paragraph 28.

For the next theorem the set B in the upper definition has the special form $B := \{x \in \mathbb{R}^2 : \|x\|_\infty \leq 1\}$.

Theorem 2.5 Let $x, y, \tilde{x}, \tilde{y} \in C[a, b]$ with $\|x - \tilde{x}\|_\infty < \varepsilon$ and $\|y - \tilde{y}\|_\infty < \varepsilon$. Then for the sets $\mathbf{P}_1 := \text{trace}(x(t), y(t))$ and $\mathbf{P}_2 := \text{trace}(\tilde{x}(t), \tilde{y}(t))$ we have

$$\delta(\mathbf{P}_1, \mathbf{P}_2) < \varepsilon.$$

Proof: Let $\varepsilon_0 := \max(\|x - \tilde{x}\|_\infty, \|y - \tilde{y}\|_\infty)$, then $\varepsilon_0 < \varepsilon$. We show that both $\mathbf{P}_1 \subseteq (\mathbf{P}_2)_{\varepsilon_0}$ and $\mathbf{P}_2 \subseteq (\mathbf{P}_1)_{\varepsilon_0}$ hold. Let p be an arbitrary point of \mathbf{P}_1 . Due to $p \in \mathbf{P}_1 = \text{trace}(x, y)$, we have a $\tilde{t} \in [a, b]$ with $p = (x(\tilde{t}), y(\tilde{t}))$. We choose the point $\tilde{p} := (\tilde{x}(\tilde{t}), \tilde{y}(\tilde{t})) \in \mathbf{P}_2$. From the assumption of the theorem and the nature of the special chosen set B we have $|x(\tilde{t}) - \tilde{x}(\tilde{t})| \leq \varepsilon_0, |y(\tilde{t}) - \tilde{y}(\tilde{t})| \leq \varepsilon_0$. Therefore, $p \in B_{\varepsilon_0}(\tilde{p}) \subseteq (\mathbf{P}_2)_{\varepsilon_0}$. Analogously, we show that $\mathbf{P}_2 \subseteq (\mathbf{P}_1)_{\varepsilon_0}$. \square

We can summarize our considerations in the following way.

Theorem 2.6 Let $x, y \in \mathbb{S}_t^1$ be an arc-length parametrization of a given piecewise linear curve $\text{trace}(P)$. If we apply algorithm 2.2 to x, y with a given tolerance $\varepsilon > 0$, we obtain a sequence of functions $x^j, y^j \in \mathbb{S}_{\mathcal{T}^j}^1$ with following properties:

- (8) for all $\tau \in \mathcal{T}^j$ we have $(x^j(\tau), y^j(\tau)) \in P$,
(9) $\delta(\text{trace}(x^j, y^j), \text{trace}(P)) < \varepsilon$.

Proof: Since the in (6) defined approximation operator \mathbf{G} has interpolation properties we get (8). According to theorem 2.3 we have $\|x^j - x\|_\infty < \varepsilon$ and $\|y^j - y\|_\infty < \varepsilon$. With theorem 2.5 we obtain (9). \square

3 Examples

The algorithm 2.2 has been implemented in MATLAB, and a great amount of experiments have been done. In this section we present some examples. In the first two examples, figures 3.1 and 3.4, we see an original curve and its coordinate functions, followed by their evolved versions obtained by algorithm 2.2 with tolerance values $\varepsilon = 0.01$ and $\varepsilon = 0.03$.

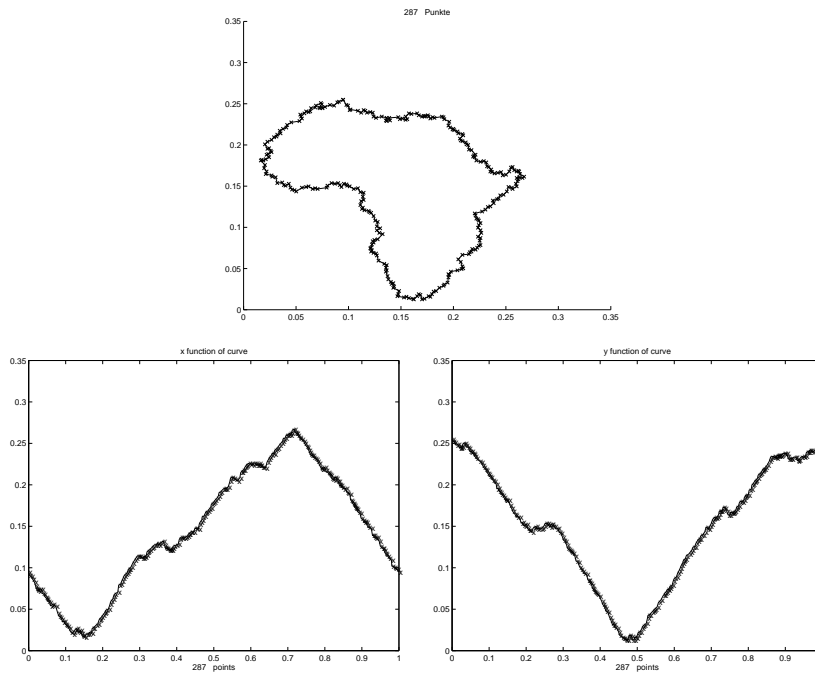


Figure 3.1 A given piecewise linear curve with 287 point and its arc-length parametrization.

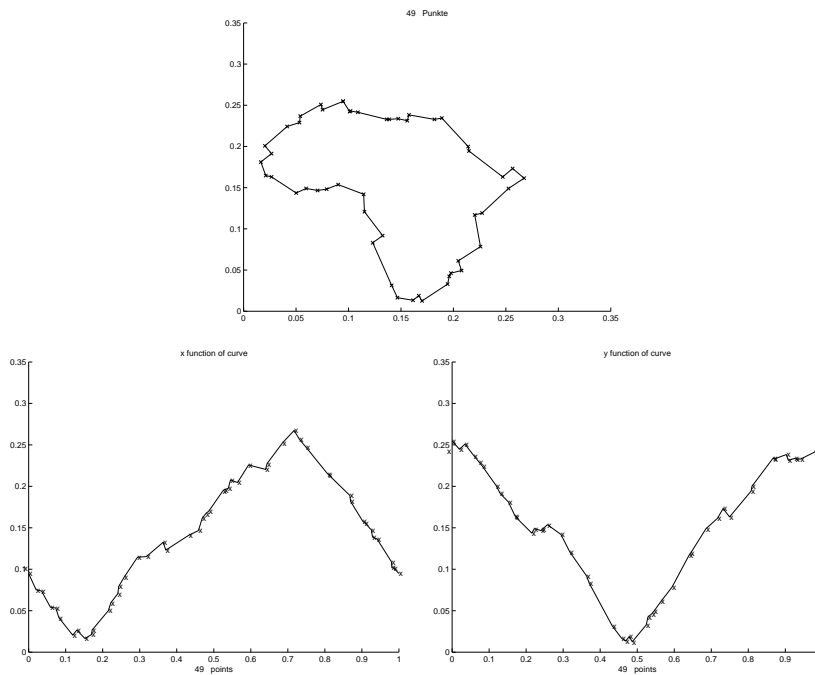


Figure 3.2 Evolved version of curve in figure 3.1 with tolerance $\varepsilon = 0.01$. Number of points: 49.

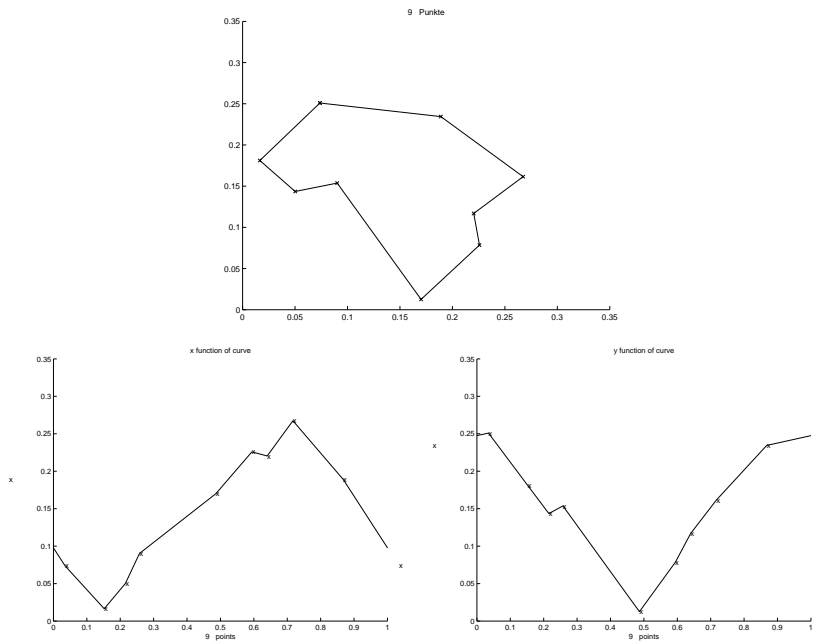


Figure 3.3 Evolved version of curve in figure 3.1 with tolerance $\varepsilon = 0.03$. Number of points: 9.

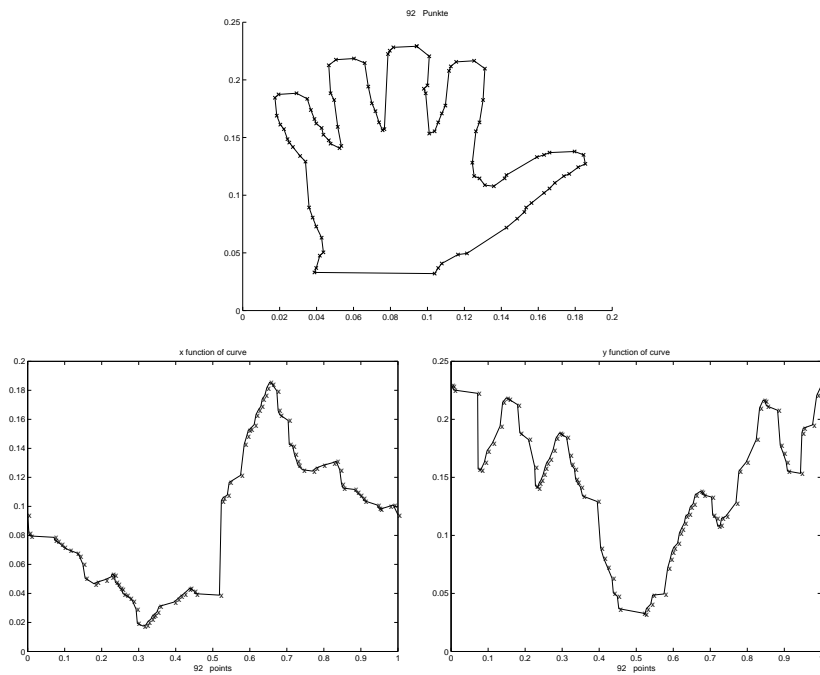


Figure 3.4 A given piecewise linear curve with 92 point and its arc-length parametrization.

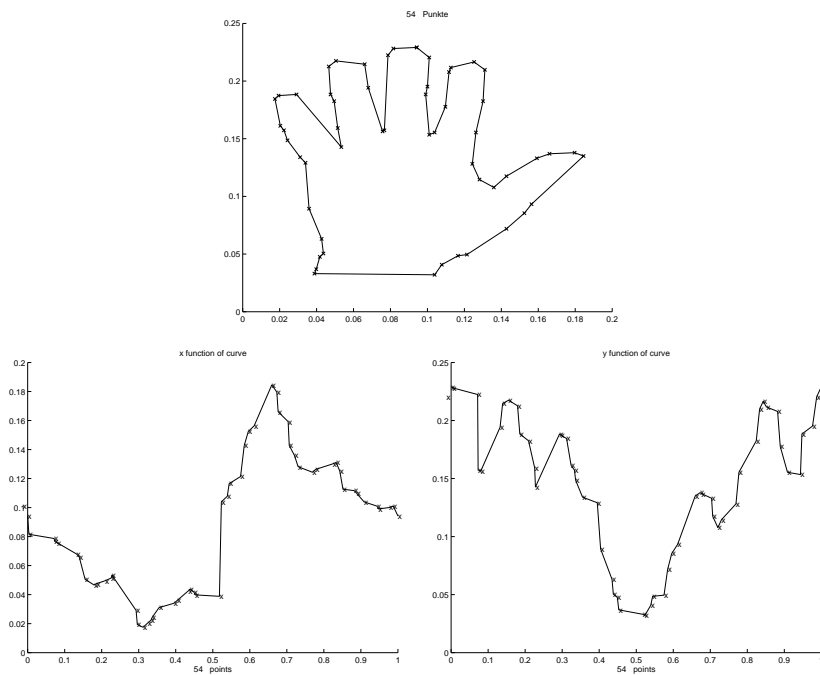


Figure 3.5 Evolved version of curve in figure 3.4 with tolerance $\varepsilon = 0.01$. Number of points: 54.

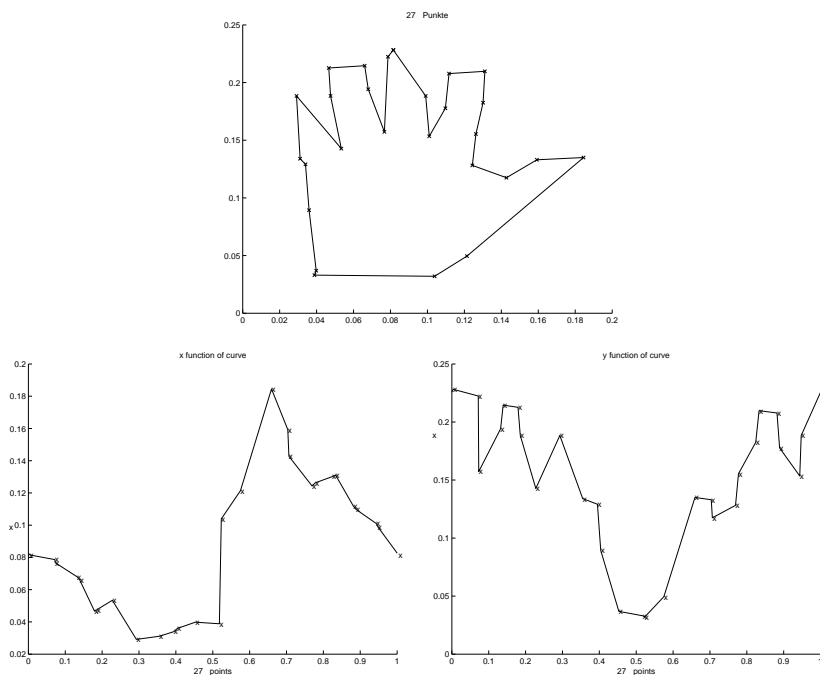


Figure 3.6 Evolved version of curve in figure 3.4 with tolerance $\varepsilon = 0.03$. Number of points: 27.

In the following three figures we see an original curve, and its two evolved versions.

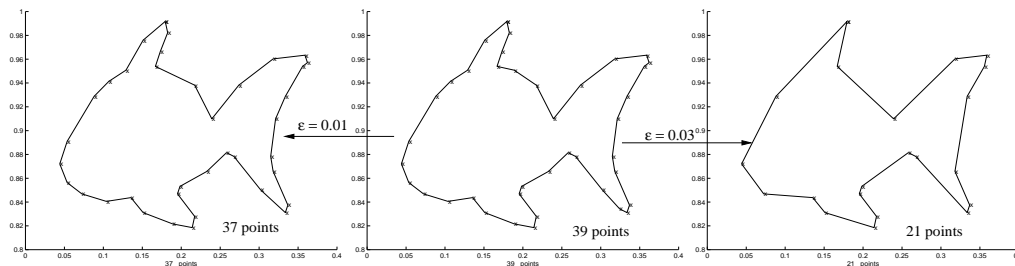


Figure 3.7 A given piecewise linear curve with 39 point and its evolved versions.

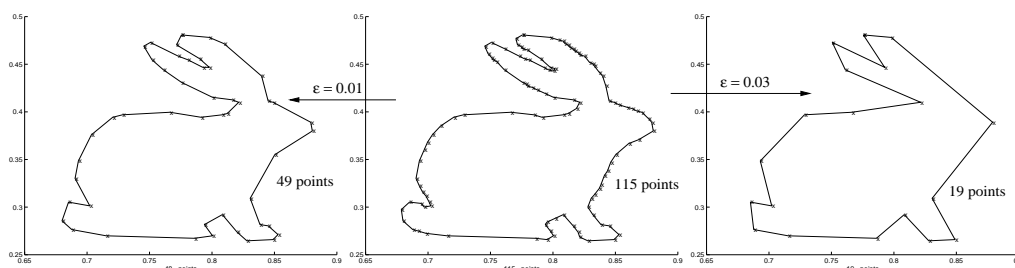


Figure 3.8 A given piecewise linear curve with 115 point and its evolved versions.

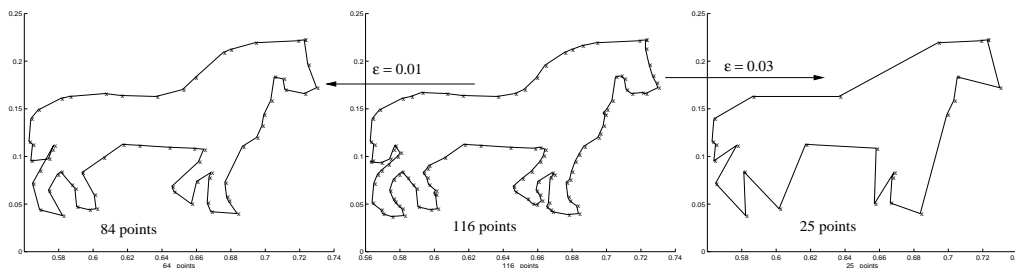


Figure 3.9 A given piecewise linear curve with 116 point and its evolved versions.

References

- [Böhm 1980] W. Böhm, (1980) Inserting new knots into B-spline curves. Computer Aided Design 12, pp. 199-201.
- [Cheney 2000] E. W. Cheney, W. Light, (2000) *A Course in Approximation Theory*, Brooks/Cole series in advanced mathematics.
combination of B-Splines. A survey, L.L., Eds.,
- [de Boor 1978] C. de Boor, (1978) *A Practical Guide to Splines*, Springer-Verlag, New York.
- [Hausdorff 1927] F. Hausdorff, (1927) *Mengenlehre*, New York, Dover Publications.
Identifikation zweidimensionaler Objekte,

- [Latecki 1998] L. J. Latecki, R. Lakämper, (Sep. 1998) Discrete Approach to Curve Evolution, Proc. DAGM Conf. on Pattern Recognition, Stuttgart, Seite 85-92.
- [Latecki 1999] L. J. Latecki, R. Lakämper, (Sep. 1999) Polygon Evolution by Vertex Deletion, Proc. of Int. Conf. on Scale-Space Theories in Computer Vision, M. Nielsen, P. Johanson, O. F. Olsen, J. Weikert.
- [Latecki 2000] L. J. Latecki, R.-R. Ghadially, R. Lakämper, and U. Eckhardt, (July 2000) Continuity of the discrete curve evolution. *Journal of Electronic Imaging* 9 (3), pp. 317-326.
- [Latecki 2000] L. J. Latecki, R. Lakämper, (October 2000) Shape Similarity Measure Based on Correspondence of Visual Parts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, pp. 1185-1190.
Evolution, noch nicht erschienen.
- [Lyche 1986] T. Lyche and K. Mörken, (1986) Making the Oslo Algorithm more efficient, *SIAM J. Numer. Anal.* 23, Seite 663-675.
- [Lyche 1987] T. Lyche and K. Mörken, (1987) Knot removal parametric B-Splines curves and surfaces, *Comut. Aided Geom. Design* 4, Seite 217-230.
- [Lyche 1988] T. Lyche and K. Mörken, (1988) A data-reduction strategy for splines with applications to the approximation of functions and data, *IMA J. Numer. Anal.* 8, Seite 185-208.
- [Nürnbergger 1980] G. Nürnbergger, (1980) *Approximation by Spline Functions*, Springer-Verlag.
- [Opfer 2002] R. Opfer, (2002) Anwendungen der diskreten Kurvenevolution und deren Verallgemeinerung auf höherdimensionale Flächen mittels radialer Basisfunktionen, Master Thesis, Universität Hamburg.
- [Schaback 1997] R. Schaback, Reconstruction of Multivariate Functions from Scattered Data, home page Schaback (2002) : <http://www.num.math.uni-goettingen.de/schaback/>.
- [Schumaker 1981] L.L. Schumaker, (1981) *Spline Functions: Basic Theory*, Wiley, New York.

Addresses

Longin Jan Latecki
 Dept. of Computer and Information Sciences
 Temple University
 1805 North Broad St.
 Philadelphia, PA 19122, USA
 e-mail: latecki@temple.edu

Roland Opfer
 Department of Mathematics
 Georg-August-Universität Göttingen
 Lotzestr. 16-18
 37073 Göttingen, Germany
 e-mail: opfer@math.uni-goettingen.de