On the min DSS problem of closed discrete curves

F. Feschet* L. Tougne†

Technical report LLAIC - 2002/06

Abstract

Given a discrete eight-connected curve, it can be represented by discrete eight connected segments. In this paper, we try to determine the minimal number of necessary discrete segments. This problem is known as the min DSS problem. We propose to use a generic curve representation by discrete tangents, called a tangential cover which can be computed in linear-time. We introduce a series of criteria each having a linear-time complexity to progressively solve the min DSS problem. This results in an optimal algorithm both from the point of view of optimization and of complexity, outperforming the previous quadratic bound.
keyword Min DSS Problem, discrete curves and segments, tangential cover, optimal complexity.

1 Introduction

Let us given a closed discrete curve $\mathcal{C}$ which is eight connected and coded as a list of consecutive points M_i with $i \in \{1, .., n\}$. In the sequel we note $M_i \ \leq, =, \geq \ M_j$ if and only if $i \ \leq, =, \geq \ j$. If $\mathcal{C}$ is a Jordan curve then the Freeman chain coding of its boundary can be used as the input list of the present work. In case of self-intersecting curves, we only suppose the list to be available whatever the way to compute it. For the strict definition of a eight connected curve, we refer to [1]. It is well known that an eight connected discrete curve, with width one, can be locally and globally represented by discrete segments.

There are various definitions of a discrete segment, such as the one of Bresenham [2], of Rosenfeld [3] with the chord property, of various authors with word processing approaches [4] or by Reveillès [5] with arithmetical studies. Surveys can be found in [6, 4]. In the present work, we choose to use the definition of Reveillès for its algorithmic properties.

Definition 1.1 *A discrete segment S is a set of points $(x_i, y_i)_{1\leq i\leq n}$ of $\mathbb{Z}^2$ which verifies*

$$\forall i \in \{1, \dots, n\}, \quad \mu \leq ax_i - by_i < \mu + \omega \tag{1}$$

*{feschet@llaic3.u-clermont1.fr} LLAIC1 - IUT Clermont-Ferrand - 63172 Aubière, France.

†{Laure.Tougne@univ-lyon2.fr} Laboratoire LIRIS, Université Lumière Lyon2, 5, avenue Pierre-Mendès-France F-69676 Bron, France.

where the quadruplet $(a, b, \mu, \omega) \in \mathbb{Z}^3 \times \mathbb{N}$ are the parameters of a discrete line (an unbounded set). The ratio a/b is the slope of the line, the μ parameter is its shift and ω is its thickness.

We consider usual eight connected segments given by $\omega = \max(|a|, |b|)$. The limiting points (x_1, y_1) and (x_n, y_n) of S will be denoted by $\pi_1(S)$ and $\pi_2(S)$ respectively.

Given a curve $\mathcal{C}$, a discrete segment S composed of points of $\mathcal{C}$ will be called maximal if and only if no connected point P to S in $\mathcal{C}$ can be found such that $S \cup \{P\}$ is a discrete segment.

In the sequel, any curve $\mathcal{C}$ will be oriented with respect to the natural ordering of the indices of its Freeman chain codes. Thus, the right will refer to points with higher indices, the left being the converse.

Definition 1.2 *The polygonalization P_P of the curve $\mathcal{C}$ starting at the point P is a graded set of elements of $\mathcal{C}$, $P_P = \{S_j,\ j \in [1, k]\}$, such that,*

1. $\forall j \in [1, k], \quad S_j$ *is a discrete segment,*
2. $\forall j \in [1, k-1], \quad S_j \cap S_{j+1} = \pi_2(S_j) = \pi_1(S_{j+1}),$
3. $S_k \cap S_1 = \pi_2(S_k) = \pi_1(S_1) = P.$

$|P_P| = k$ *is called the length of the polygonalization.*

A polygonalization is called locally maximal if and only if $\forall j \in [1, k-1]$, S_j is a maximal discrete segment with respect to the right only. There is a special case here for the last segment because it might not be maximal. From the point of view of complexity, computing a polygonalization is linear in time [7, 8]. We can now formulate the problem we try to solve,

Min DSS problem: What is the minimal length of a locally maximal polygonalization of a given eight connected curve $\mathcal{C}$?

Two different locally maximal polygonalizations can have different lengths (see Fig. 1). Of course, as noted by Rosenfeld and Klette [6], the use of maximal discrete segments implies that the lengths of two different locally maximal polygonalizations are either the same or differ by one. Thus, the problem to solve is equivalent to determining whether or not there exist two locally maximal polygonalizations with different lengths.

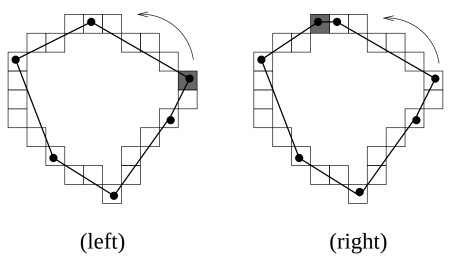

Figure 1: Two locally maximal polygonalizations with 6 (left) or 7 (right) discrete segments. The starting point is indicated by a gray point.

2 Tangential cover and polygonalization

Let us first introduce the notion of tangential cover and remember the outline of an algorithm to compute the tangential cover of the curve $\mathcal{C}$. The reader can refer to Feschet and Tougne [9] for the full details and proofs of the algorithm. From now on in this paper, we will use the term polygonalization instead of locally maximal polygonalization.

The first links between the tangential cover and the polygonalizations of the curve are introduced in this section.

2.1 Definitions

Let us denote by T^P the discrete tangent at P to the curve $\mathcal{C}$. It is defined as the maximal segment around P.

Definition 2.1 *The tangent T^P at $P = M_i$ is a segment $\{M_{i-k-q}, \ldots, M_i, \ldots, M_{i+k+p}\}$, with $p = 0$ or $q = 0$, such that neither $T^P \cup \{M_{i-k-q-1}\}$ nor $T^P \cup \{M_{i+k+p+1}\}$ are discrete segments.*

As for any segment, $\pi_1(T^P)$ and $\pi_2(T^P)$ denote the extremities of the tangent. The tangential cover is the set of all distinct tangents of the curve $\mathcal{C}$.

Definition 2.2 *The tangential cover $\mathcal{T}$ of $\mathcal{C}$ is a graded set of subsets of $\mathcal{C}$, $\mathcal{T} = \{T_j,\ j \in [1, m]\}$, such that,*

1. $\forall j \in [1, m]$, $\quad T_j$ *is a discrete tangent,*
2. $\forall P \in \mathcal{C}$, $\exists j \in [1, m]$, $T^P = T_j$,

By construction, $\pi_1(T_j)$ and $\pi_2(T_j)$ are strictly increasing sequences relative to the ordering of the indices in $\mathcal{C}$. The sequences $\pi_1()$ and $\pi_2()$ are overlapping sequences: $\pi_1(T_j) < \pi_1(T_{j+1}) \leq \pi_2(T_j) < \pi_2(T_{j+1})$. We call T_k^M the tangents of $\mathcal{T}$ that contain the point M (see Fig. 2).

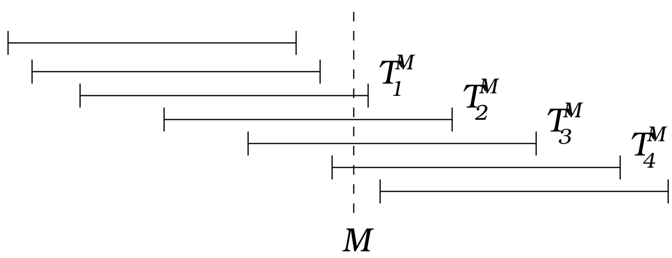

Figure 2: Examples of tangents T_k^M.

The computation of $\mathcal{T}$ can be done in $\mathcal{O}(n)$ [9].

2.2 First properties

In this part, we prove that all the polygonalizations can be deduced from the tangential cover in linear time. Since the tangential cover construction is also in linear time, this permits to construct a novel algorithm for the computation of discrete polygonalizations in linear time. The main advantage of this algorithm is that after the preprocessing of the tangential cover computation, all the polygonalizations are computed with a linear complexity with respect to the cardinal of the cover. In practice, this cardinal is several time smaller than the number of points in the curve. We use the simplification broaden by the tangential cover to approach the min DSS problem.

Property 2.1 *Let M a point of $\mathcal{C}$, N its successor in the polygonalization P_M and T_1^M,...,T_k^M the tangents that contain M. Let us denote by S the point following N in $\mathcal{C}$. Then,*

$$\forall i \in [1,k],\ S \notin T_i^M \quad and \quad N = \pi_2(T_k^M) \tag{2}$$

Proof
Let us take $i \in [1,k]$. We have $M \in T_i^M$. If $N \notin T_i^M$ then $S \notin T_i^M$ since S is after N in $\mathcal{C}$. If, on the contrary, $N \in T_i^M$ then $S \notin T_j^M$ because the maximal discrete segment starting at M in $\mathcal{P}_M$ ends at N and thus M, N and S does not belong to a common discrete segment.

Let us now consider the tangent of the tangential cover which is computed at the middle of M and N. This tangent is a maximal discrete segment which contains both M and N (see [9]). Thus it is one of the T_j^M's. Since $S \notin T_j^M$ for all $j \in [1,k]$, we deduce that this tangent is T_k^M and so that $N = \pi_2(T_k^M)$.

□

Let us denote by f the function which associates to a point M the tangent T_k^M, that is the last tangent of the tangential cover, which contains M.

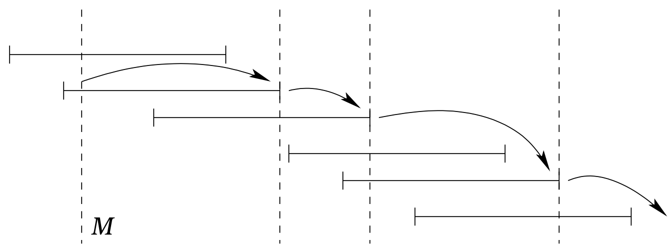

Figure 3: The iterative construction of a polygonalization

Property 1 says that for any M in $\mathcal{C}$, the maximal discrete segment starting at M ends at $\pi_2(f(M))$. And this can be recursively applied (Fig. 3) to get,

Theorem 2.1 *The polygonalization P_M starting at M is equal to the graded set $\{S_j\}_{1 \leq j \leq m}$ with,*

$$\forall j \in [1,m-1],\ \pi_2(S_j) = \pi_2(f^j(M)) \tag{3}$$

where f^j denotes the j-th iterate of f. Of course, $\pi_2(S_m)$ equals $\pi_1(S_1)$ equals M by definition.

In the theorem, m is the highest natural number such that $\pi_2(f^m(M)) < M$. We call $f^*(M)$ this last tangent used in the polygonalization P_M. Using the theorem, we can now compare two polygonalizations, P_M and $P_{\pi_1(f(M))}$. It is clear that these polygonalizations merge at $\pi_2(f(M))$. Thus, they are the same except possibly at the end if they differ in length. Thus, only the last segments of the decompositions are worth interesting.

3 Solving the min DSS problem

Before giving the whole proof, let us study its major steps. First of all, we prove in theorem 3.1 that we can restrict our attention to the subset of polygonalizations starting at $\pi_1(T)$ for any T in the tangential cover. This permits to reduce the set of possible polygonalizations and to fully use the structure of the tangential cover. Then, we remark that only the end of a polygonalization is important, that is the way we return to the starting point of the polygonalization. We thus introduce what is called the residue of a tangent and prove in theorem 3.2 that if for any T in C, its residue is not empty, then with this T we solve the min DSS problem. However, some discrete curves have a tangential cover for which all residues are empty. We next shows that merging is the crucial point. If no merging exists then the min DSS can be solved locally (proposition 3.2). However, this is not the case in general and a specific algorithm is designed to manage the merging for a particular subset of the tangential cover. The complexity analysis will be given in the next section.

3.1 Local study and blocking

Theorem 3.1 *Let M be any point of $\mathcal{C}$ and P_M the polygonalization of $\mathcal{C}$ issued from M. Then,*

$$|P_M| = \left|P_{\pi_1(f(M))}\right| \quad \text{if and only if} \quad \pi_2(f^*(M)) < \pi_1(f(M)) \tag{4}$$

Proof
By definition, $f^*(M)$ is the last tangent, until cycling, which does not contain M and which is used in P_M. But $\pi_1(f(M)) \leq M$ by definition. So there are two cases: either $\pi_2(f^*(M)) \in f(M)$ or $\pi_2(f^*(M)) \notin f(M)$.

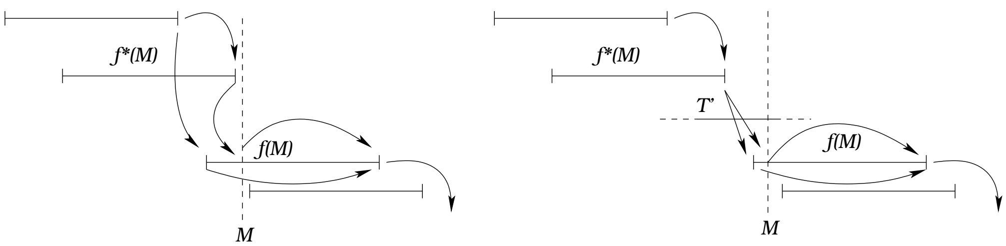

Figure 4: The two cases of theorem 2

In the first case (see Fig. 4 left), the step that permits to reach $f^*(M)$ is done from a tangent before it. But, since $f^*(M)$ also covers $f(M)$, this means that in one step, we can reach $\pi_1(f(M))$. Thus, we only need $|P_M|-1$ segments for $P_{\pi_1(f(M))}$. Hence, the min DSS is solved.

In the second case (see Fig. 4 right), $\pi_2(f^*(M)) \notin f(M)$ but since M is reachable by a discrete segment from $\pi_2(f^*(M))$, there exists a tangent T' in the tangential cover which is after $f^*(M)$ and which contains M. Let us simply remark that T' also contains $\pi_1(f(M))$ because $M \geq \pi_1(f(M))$. Hence, the polygonalizations P_M and $P_{\pi_1(f(M))}$ have equal lengths.

□

In previous theorem, we have shown that if M is not a point $\pi_1(.)$ of the tangential cover then using $\pi_1(f(M))$, we always construct a polygonalization which is shorter, in a wide sense, than the polygonalization P_M. Hence, the tangential cover always contains a polygonalization at most of the same length than any polygonalization constructed from a point not in the cover. Thus, we can restrict the min DSS problem to the polygonalizations constructed from $\pi_1(.)$ points of the tangential cover. In the following definition, indices are intended modulo the cardinal of the tangential cover.

Definition 3.1 *Let T be any tangent in C.*
We denote by $B(T)$ and we call it the backward set of T, the set of points M in T such that there exists T' in C with $T' < T$ and $M \in T'$.
We denote by $F(T)$ and we call it the forward set of T, the set of points M in T such that there exists T' in C with $T < T'$ and $M \in T'$.
We denote by $R(T)$ and we call it the residue of T, the set of points defined by $R(T) = T \setminus (B(T) \cup F(T))$.

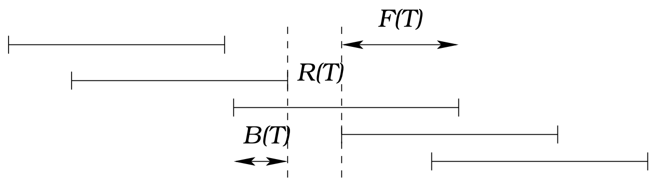

Figure 5: $B(T)$, $F(T)$ and $R(T)$ of a tangent T.

Fig. 5 illustrates such notations. Of course, due to the structure of the cover, we can formulate precise characterizations of the backward and the forward sets of any tangent T. For this, let us denote by T_{prev} and T_{next} be the tangents just before and just after T in the cover. Then, it is clear that,

$$\begin{aligned} B(T) &= \{\pi_1(T), \ldots, \pi_2(T_{\text{prev}})\} \\ F(T) &= \{\pi_1(T_{\text{next}}), \ldots, \pi_2(T)\} \end{aligned} \tag{5}$$

Hence the following local characterization of the residue is true,

$$R(T) \neq \emptyset \Longleftrightarrow \pi_2(T_{\text{prev}}) < \pi_1(T_{\text{next}}) \tag{6}$$

Theorem 3.2 *Let T be any tangent in C. Then,*

$$R(T) \neq \emptyset \Longrightarrow \forall M \in R(T), \ \ |P_M| = \left|P_{\pi_1(T)}\right| + 1 \tag{7}$$

Proof
Suppose we have T in C with $R(T) \neq \emptyset$. Choose M in $R(T)$. Then since $M \notin F(f(M))$, the first segment of the polygonalization P_M begins at the point M and ends at $\pi_2(T)$. That is from $\pi_2(T)$, the polygonalizations P_M and $P_{\pi_1(T)}$ are similar. Let us study now $T' = f^*(\pi_1(T))$. By definition, there exists T'' in C which covers $\pi_2(T')$ and $\pi_1(T)$. This tangent T'' is used to end the polygonalization $P_{\pi_1(T)}$. However, since $M \notin B(T)$, we necessarily have $M \notin T''$ because $\pi_2(T'') < M$. Hence, the polygonalization P_M is not finished and we must add the segment $[T'', T]$ to end it. It is thus longer than $P_{\pi_1(T)}$.

□

The min DSS is not solved at this point if and only if we have $R(T)$ to be the empty set for any T in C. This is what we suppose in the sequel.

Theorem 3.3 *For any T in C, if $\exists T' \in C$ such that*

$$\pi_1(T') \leq \pi_2(f^*(\pi_1(T))) < \pi_1(f(\pi_2(T)) \leq \pi_2(T')$$

then,

$$\left|P_{\pi_1(f(\pi_2(T)))}\right| = \left|P_{\pi_1(T)}\right| - 1$$

Proof
Suppose $\pi_1(T') \leq \pi_2(f^*(\pi_1(T))) < \pi_1(f(\pi_2(T)) \leq \pi_2(T')$ for some tangent T'. Then, we have $T' < T$. Indeed, $\pi_2(f^*(\pi_1(T))) < \pi_1(T)$ implies that T must be after T' in the cover since the $\pi_1(.)$ sequence is strictly increasing. If we compare the two polygonalizations $P_{\pi_1(T)}$ and $P_{\pi_1(f(\pi_2(T)))}$, then it is clear these polygonalizations merge at $\pi_2(T)$ after two segments for the first one and only one segment for the second one. Fig. 6 illustrates this fact.

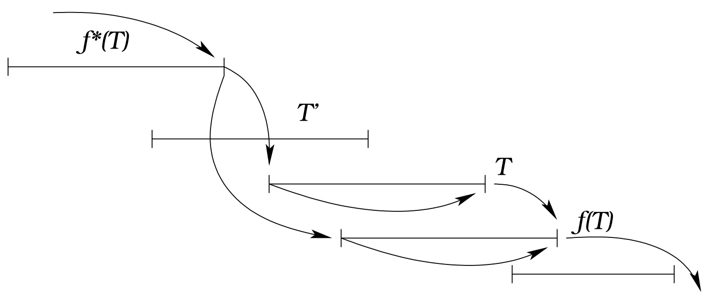

Figure 6: Solving the min DSS with T'

The last points used in $P_{\pi_1(T)}$, are respectively $\pi_2(f^*(\pi_1(T)))$ followed by $\pi_1(T)$. Moreover using T', it is clear that the last points used in $P_{\pi_1(f(\pi_2(T)))}$ are respectively $\pi_2(f^*(\pi_1(T)))$ followed by $\pi_1(f(\pi_2(T)))$. Hence, the conclusion is consequently that the length of $P_{\pi_1(f(\pi_2(T)))}$ is strictly lower than that of $P_{\pi_1(T)}$, which solves the min DSS problem.

□

The previous theorem permits to partially solved the min DSS problem. The cases that are not solved at this time corresponds to the condition: $\forall T' \in C$, neither $\pi_1(T') \leq \pi_2(f^*(\pi_1(T)))$ nor $\pi_1(f(\pi_2(T)) \leq \pi_2(T')$ since of course the others inequalities are always true. Those two conditions can be easily interpreted. If $\pi_1(f(\pi_2(T)))$ belongs to $B(T)$ then there exists tangents T' such that $\pi_1(f(\pi_2(T)) \leq \pi_2(T')$. Hence for these tangents, we necessarily have $\pi_2(f^*(\pi_1(T))) < \pi_1(T')$. On the contrary, if $\pi_1(f(\pi_2(T)))$ does not belong to $B(T)$ then the condition is always true. But for this case, we can determine the lengths of the polygonalizations $P_{\pi_1(T)}$ and $P_{\pi_1(f(\pi_2(T)))}$.

Theorem 3.4 *For any T in C,*

$$\pi_1(f(\pi_2(T))) \notin B(T) \Longrightarrow \left|P_{\pi_1(f(\pi_2(T)))}\right| = \left|P_{\pi_1(T)}\right| \tag{8}$$

Proof
Let us suppose that $\pi_1(f(\pi_2(T))) \notin B(T)$ (see Fig. 7).

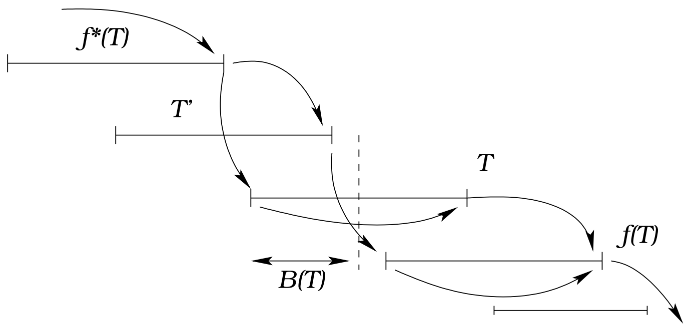

Figure 7: Case $\pi_1(f(\pi_2(T))) \notin B(T)$

By definition of $f^*(\pi_1(T))$ there must exist a tangent T' which covers both $f^*(\pi_1(T))$ and T thus allowing to end the polygonalization $P_{\pi_1(T)}$ in only one step. Suppose that we choose T' to be the closest tangent to T which allows to end the polygonalization $P_{\pi_1(T)}$ in one step exactly. As previously, the polygonalizations $P_{\pi_1(T)}$ and $P_{\pi_1(f(\pi_2(T)))}$ have merged after two steps for the first one and one step for the last one. So they are equal until $\pi_2(f^*(\pi_1(T)))$. We can use T' to reach $\pi_2(T')$ but not point at its right. Indeed, to reach a point at the right of $\pi_2(T')$, we must use a tangent after T' because the sequence $\pi_2(.)$ is strictly increasing. However, no tangent after T' covers $f^*(\pi_1(T))$ and thus can not be used from $\pi_2(f^*(\pi_1(T)))$. Hence, the polygonalization $P_{\pi_1(f(\pi_2(T)))}$ uses $\pi_2(T')$ and then using T goes back to $\pi_1(f(\pi_2(T)))$. Hence the two polygonalizations have equal lengths.

□

The only case which is still not solved is when $\pi_1(f(\pi_2(T)))$ belongs to $B(T)$ and that for each tangent T' covering $f(\pi_2(T))$, we have $\pi_2(f^*(\pi_1(T))) < \pi_1(T')$.

Property 3.1 *Let us choose T' to be the farthest from T, that is T' is the tangent before T, which covers $f(\pi_2(T))$ and such that $\pi_2(T')$ is minimal.*

$$|P_{\pi_1(T')}| = |P_{\pi_1(T)}| = |P_{\pi_1(f(\pi_2(T)))}| \tag{9}$$

Proof
First, it exists T'' before T such that $f^*(\pi_1(T))$ covers T'' and T'' covers T. This permits to end the polygonalization $P_{\pi_1(T)}$ in only one step. T'' is necessarily before T' because it is covered by $f^*(\pi_1(T))$ and T' is not. The residue of T'' is not empty. Hence, since T' and $f^*(\pi_1(T))$ are disjoint, there must exist a tangent L which is covered by $f^*(\pi_1(T))$ and which covers T'. This tangent is either before T'' or after it (see Fig. 8). Second, we have $f(\pi_2(T')) =$

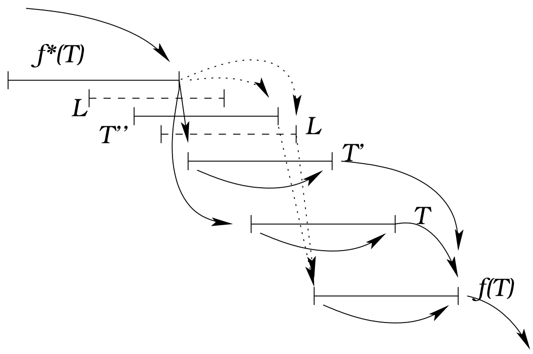

Figure 8: T'', L and the polygonalizations

$f(\pi_2(T))$. Indeed $f(\pi_2(T')) \leq f(\pi_2(T))$ since $T' < T$ but since T' covers $f(\pi_2(T))$ we have $f(\pi_2(T)) \leq f(\pi_2(T'))$ and so we get the equality. Thus the three polygonalizations $P_{\pi_1(T')}$, $P_{\pi_1(T)}$ and $P_{\pi_1(f(\pi_2(T)))}$ merge at $\pi_2(f(\pi_2(T)))$. Hence, they all arrived at $\pi_2(f^*(\pi_1(T)))$. At this point, using T'' it is possible to reach $\pi_1(T')$ in only one step and using T'', it is possible to reach $\pi_1(T)$ in only one step. Let us now remark that $\pi_1(f(\pi_2(T')))$ does not belong to $B(T')$ since no tangent before T' can covers $f(\pi_2(T))$ by definition. Hence using Theorem 3.4, we get $|P_{\pi_1(T')}| = |P_{\pi_1(f(\pi_2(T)))}|$. Since we also have $|P_{\pi_1(T)}| = |P_{\pi_1(f(\pi_2(T)))}|$ by construction, the proof is done.

□

The min DSS is not solved if either theorem 3.4 or proposition 3.1 are true. But, we can interpret proposition 3.1 to be the existence of merging between polygonalizations. Indeed, two polygonalizations merge if and only if somewhere there exist T_1 in the first one and T_2 in the second one such that $T_1 < T_2$ and $f(T_1) = f(T_2)$. This corresponds to the behavior of T' in the proof of proposition 3.1. To be precise, if we never have $\pi_1(f(\pi_2(T))) \in B(T)$, then two polygonalizations from two different points can never merge. Indeed, applying the previous argument leads to the property that two polygonalizations can merge only when $\pi_1(f(\pi_2(T)))$ is in $B(T)$ for some tangent T, which is excluded. Hence, proposition 3.1 characterizes exactly the notion of merging. If there is no merging, it is clear that two polygonalizations can differ only at the end.

Property 3.2 *If $\pi_1(f(\pi_2(T))) \notin B(T)$ for all $T \in C$, then we can solve the min DSS locally.*

Proof
Consider any tangent T such that the set $\{T, \ldots, f(\pi_2(T))\}$ is of minimal cardinality n_T. Let us compute $P_{\pi_1(T)}$ and $P_{\pi_1(f(\pi_2(T)))}$. This permits to get

$f^*(\pi_1(T))$ and $f^*(f(\pi_2(T)))$. By definition $f^*(f(\pi_2(T))) < T$ because T covers $f(\pi_2(T))$. We also have $f^*(f(\pi_2(T))) \leq f(f^*(\pi_1(T)))$ because of the minimality of n_T and the fact that there is no merging. Hence, it is clear that $f^*(f(\pi_2(T)))$ is covered by $f^*(\pi_1(T))$. To solve the min DSS problem let us simply remark that for any tangent between T and $f(\pi_2(T))$, we know their $f^*(.)$ images and thus we simply have to check if polygonalizations end in only one step or not.

□

To conclude the local analysis, care must be taken that the min DSS can be solved partially but under the hypothesis that the f^* function has been previously calculated.

3.2 Global analysis

When merging is possible, the min DSS problem can not be solved locally. The whole curve must be check because merging can occur in any polygonalizations. As previously, the residue is supposed to be empty for any tangent of the tangential cover.

The problem has been far reduced by the previous analysis. Indeed, let us define the function $f^R(.)$ to be the one such that $f^R(T) = U$ with $f(\pi_2(U)) = T$ and $\pi_2(U)$ minimal. We have $f^R(f(\pi_2(T))) = T$ if $\pi_1(f(\pi_2(T))) \notin B(T)$. Proposition 3.1 simply says that if $f^R(f(\pi_2(T))) \neq T$ then we simply have to consider $T' = f^R(f(\pi_2(T)))$ to verify the hypothesis of theorem 3.4. In fact for any tangent T, the use of $f^R(.)$ permits to construct a separating set of the tangential cover such that any polygonalization must use a tangent in the set delimited by $f^R(f(\pi_2(T)))$ and $f(\pi_2(T))$. As a consequence of proposition 3.1, there always exists – at this step of the solution of the min DSS problem – a tangent T such that $f^R(f(\pi_2(T))) = T$. We choose such a tangent in the sequel. Moreover, in practical situation, we consider a tangent T such that the set $\{T, \ldots, f(\pi_2(T))\}$ is of minimal cardinality. This choice does not influence the complexity but can drastically reduce the effective time of computations. The key point for solving the Min DSS problem is the fact that the set $\{T, \ldots, f(\pi_2(T))\}$ is a separating set in the sense that every polygonalization of the curve must intersect the set of $\pi_2(.)$ points of all these tangents. Hence, computing the polygonalizations for all $\pi_1(.)$ points is sufficient. However, the size of this sets is $\mathcal{O}(n)$ in the worst case and thus the resulting strategy is quadratic. To obtain a linear time algorithm, it is necessary to find a strategy to compute all those polygonalizations in linear-time by exploiting their intrinsic structure. The first step of our strategy consists in detecting the merging of polygonalizations at the beginning, that is after only one step. This is important because it permits to suppress redundant polygonalizations and to detect merging with no change in length. We proceed as follows. The Length() value is initialized to 1 for all tangents of the tangential cover.

$m \leftarrow 0$
$T_g \leftarrow T$
do
  T_g is marked with m
  If $(f(\pi_2(T_g))$ is not marked) Then
    $f(\pi_2(T_g))$ is marked with m

```
    Length(f(π2(Tg))) ← Length(Tg) + 1
  Else
    The tangent corresponding to the mark of f(π2(Tg)) is a tangent
    with which Tg merge at the beginning, so note it.
  End If
  m ← m + 1
  Proceed with Tg being the next tangent of the tangential cover
while (Tg ≠ f(π2(T)))
```

This code fragment permits to attach a mark to each tangent inside T and $f(\pi_2(T))$. In case of merging, as we note the tangent of merging, we can use its mark. Moreover, in case there is no merging, we update the length which is by definition the current length of the polygonalizations started at some $\pi_1(T')$ for T' between T included and $f(\pi_2(T))$ excluded. Once the initial marks have been defined, we test each tangent of the tangential cover to detect whether or not a mark has to be propagated. This is done by the following pseudo-code.

```
Tg = f(π2(T))
do
  If (Tg has been marked) Then
    If (f(π2(Tg) covers the tangent of the mark owned by Tg) Then
      If (f(π2(Tg)) is not marked) Then
        Mark f(π2(Tg)) with the mark of Tg
        Length(f(π2(Tg))) ← Length(Tg) + 1
      Else
        Note the merging with the tangent of the current mark of f(π2(Tg))
      End If
    Else
      The f*(.) of the tangent of the mark of Tg is Tg
      Length(Mark of Tg) = Length(Tg)+1
    End If
  End If
  Proceed with Tg being the next tangent of the tangential cover
while (Tg ≠ T)
```

At the end of the previous algorithm, some tangent does have an $f^*(.)$ value and others have a merging mark. The next step consists in using the $f^*(.)$ values to compute all the $f^*(.)$ values of all tangents of the set $\{T, \ldots, f(\pi_2(T))\}$. To do this, we use the merging mark. If a tangent T' taken in the previous set has an $f^*(.)$ value, nothing has to be done. Otherwise, we must use the tangent of merging to get an $f^*(.)$ value. However, the tangent of merging might also have a merging mark. So, we get a list of tangent corresponding to successive merging. This list is necessarily finite and the last tangent of the list owns an $f^*(.)$ value. So, it is sufficient to propagate it back to all elements of the list. Then, we can proceed with another tangent. What is extremely important from the point of view of complexity is that we do not use a tangent twice in a list because if it has already been used, then it has an $f^*(.)$ value and thus the list stops at the tangent. This assures that the whole set of elements of the lists is restricted to be a subset of the set $\{T, \ldots, f(\pi_2(T))\}$ with no repetitions. So, by the previous propagation process, every tangent of the set $\{T, \ldots, f(\pi_2(T))\}$ has an $f^*(.)$ corresponding in fact to the $f^*(.)$ of the tangent of merging. Thus, the reader must be careful that these $f^*(.)$ are only approximations of the true values. Indeed in case of some merging, they have to be corrected. On Fig. 9, there are two types of merging.

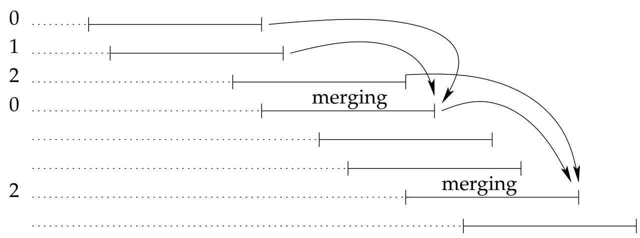

Figure 9: Two cases of merging without or with change in length

The first one, between 1 and 0, has no influence on the length whereas the second one between 0 and 2 does change the length in the sense that two steps has been done from $\pi_1(2)$ but three steps from $\pi_1(0)$ was necessary. We must note this difference to check with the use of $f^*(.)$ whether or not these two polygonalizations really have different lengths or not. The key point is that if the mark are in decreasing order then there is no change in length, otherwise, the one with the lowest mark has a length greater by one than the other one. Thus for each merging, we perfectly know whether or not the merging generates a difference in length. To conclude with the min DSS, it is now sufficient to check the $f^*(.)$ values. Remember that those values are computed only for the tangent of merging which has raised its mark first. Hence, for each tangent, we must check if its $f^*(.)$ value is correct or not. To do this, it is sufficient to test whether or not the current $f^*(.)$ covers the tangent or not. If not, then the value is correct otherwise the $f^*(.)$ is not correct and we conclude that we have used one unnecessary tangent and thus, we decrease the length by one to correct it. After this correction, the correct length is computed for each tangent of the set $\{T, \ldots, f(\pi_2(T))\}$. Hence, a min search suffices to detect the solution of the min DSS problem.

4 Complexity analysis

Several key points must be analyzed. First of all, the tangential cover can be computed in linear time from the curve [9]. One important fact is that the number of tangents in the tangential cover is bounded by the number of points of the original curve. Thus, with n points in the curve, we get a $\mathcal{O}(n)$ size of the tangential cover. Calculating $R(.)$ can be done locally and thus it is a linear time computation. Next, finding a T such that $f^R(f(\pi_2(T))) = T$ can be done in linear time and thus finding the one with minimal width is also done in linear time. The initial marking is done in linear time and the propagation is also done in linear time since the propagation of $f^*(.)$ permits to use the tangents one time only. Next is the propagation of $f^*(.)$. This is done simply by doing one turn of the tangential cover using local rules only. Hence, it is a linear time process. Correcting the $f^*(.)$ values is linear time since it is done locally. Finally, check the length is exactly the search for a min and thus is linear time.

Hence, we can give the final theorem.

Theorem 4.1 *The complexity of our algorithm for solving the min DSS problem is $\mathcal{O}(n)$ and thus reaches the lower bound. Consequently, it is an optimal algorithm.*

5 Example

In this section, we illustrate the algorithm on a concrete example. Fig. 10 gives the starting curve. This is the eight-connected border of a circle of radius 10. The point with index 0 is the rightest point on the lowest part of the circle. The curve is decomposed via Freeman coding using a counter-clockwise order.

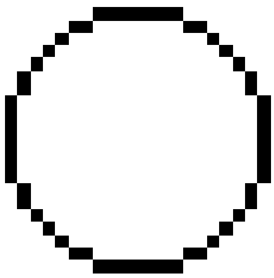

Figure 10: A test curve

The first step of the algorithm is obviously the computation of the tangential cover. It is given on Fig. 11. The hatched bars in the figure represent, for each tangent T, the intersection between $B(T)$ and $F(T)$. As we can remark, all the intersections are not empty. Consequently, all the $R(T)$ are empty and we cannot conclude here.

The following step consists in looking at the $\pi_1(f(\pi_2(T)))$. The figure 12 gives a graphical representation of the function f. We can remark that there exists at least one tangent T such that $\pi_1(f(\pi_2(T))) \in B(T)$ (in fact, all gray arrows represent such case). Consequently, we have to use the function f^* and so, we cannot solve locally the problem on this curve.

In the same way, it is easy to compute the f^R function and we do not represent it here for simplicity. From this, we can obtain the minimal separating set of the tangential cover of the curve. Remember that it is obtained considering a tangent T such that $f^R(f(\pi_2(T))) = T$ such that $T, ..., f(\pi_2(T))$ is minimal. In our example, the separating set is [55,4],...,[4,9], where $[t1, t2]$ indicates the extremities of the corresponding tangent.

Fig. 13 indicates the separation set and the marks (in black) obtained after using the algorithm. Let us recall that this algorithm attach a mark to each tangent inside $T, ..., f(\pi_2(T))$.

The second presented algorithm propagates such marks (in grey) and compute the merging. Considering the polygonalizations in the merging (in our example we only have one merging) we can conclude : the polygonalization with mark $m = 2$ has a length equals to 8 and the polygonalization with mark

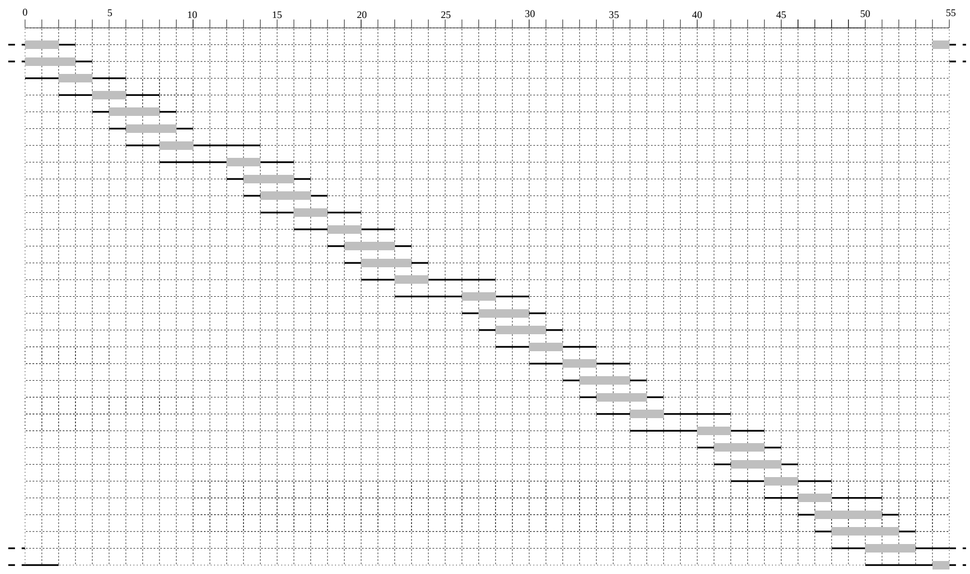

Figure 11: The tangential cover and the intersections between $B(T)$ and $F(T)$

$m = 0$ has a length equals to 9. So, a minimal length polygonalization is the one starting at point 2. It is interesting to remark that only the tangents reached by the f function are used. Indeed, it has been proved before that the iterates of f generates the elements of a polygonalization. However, this facts has also a practical impact because the complexity is linear in the number of tested tangents and this number is $\mathcal{O}(n)$ in the worst case. But the true execution time really depends on the number of used tangents and thus, as it can be seen on the example, this might represent only a small part of the tangential cover. This explain why, in practice, the running time of the algorithm is small due to a small constant in front of the dominant n term.

6 Conclusion

In this paper, we have proposed a solution to the min DSS problem. This problem concerns the computation of the smallest polygonalization of a discrete curve with respect to the number of discrete segments. A solution of this problem can be computed in quadratic time but it was an open problem to solve it in linear time. Our study is based on a curve decomposition called the tangential cover. This redundant decomposition is based on the notion of discrete tangents. As a consequence, all segments of any polygonalization are in or can be deduced from the tangential cover. By a careful study of connectivity links between discrete tangents, a linear time algorithm is deduced. Since computing the decomposition can be done in linear time, this gives the first linear time algorithm for the min DSS problem. The tangential cover can be extended to deal with imperfect curves where some points are missing or to thick curves. This is a work under progress.

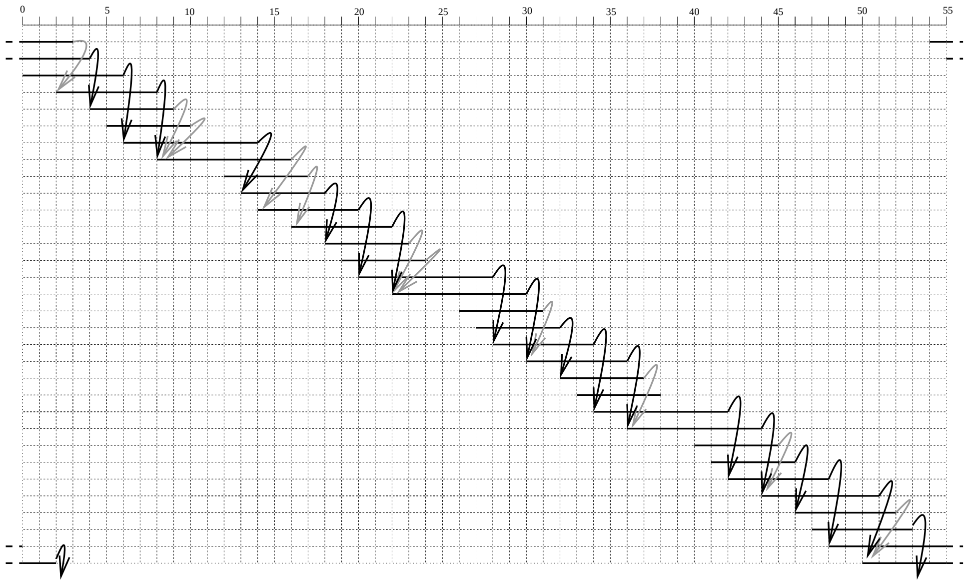

Figure 12: The function f

Acknowledgments.

Many thanks to the referees who gave us many hints to improve the paper.

References

[1] K. Voss, Discrete Images, Objects, and Functions in $\mathbb{Z}^n$, no. 11 in Algorithms and Combinatorics, Springer-Verlag, 1993.

[2] J. Bresenham, An incremental algorithm for digital plotting, in: Proceedings of ACM National Conference, 1963.

[3] A. Rosenfeld, Digital straight line segments, IEEE Transactions on Computers 23 (1974) 1264–1269.

[4] M. Lothaire, Algebraic Combinatorics on Words, Cambridge University Press, 2002, Ch. Sturmian Word (J. Berstel and P. Séébold), pp. 40–95.

[5] J.-P. Reveillès, Géométrie discrète, calcul en nombres entiers et algorithmique, Thèse d'etat, Université Louis Pasteur, Strasbourg (1991).

[6] A. Rosenfeld, R. Klette, Digital straightness, in: S. Fourey, G. Herman, T. Kong (Eds.), IWCIA 2001 Proceedings, Vol. 46 of ENTCS, Elsevier Science Publishers, 2001.

[7] A. Smeulders, L. Dorst, Decomposition of discrete curves into piecewise straight segments in linear time, Contemporary Mathematics 119 (1991) 169–195.

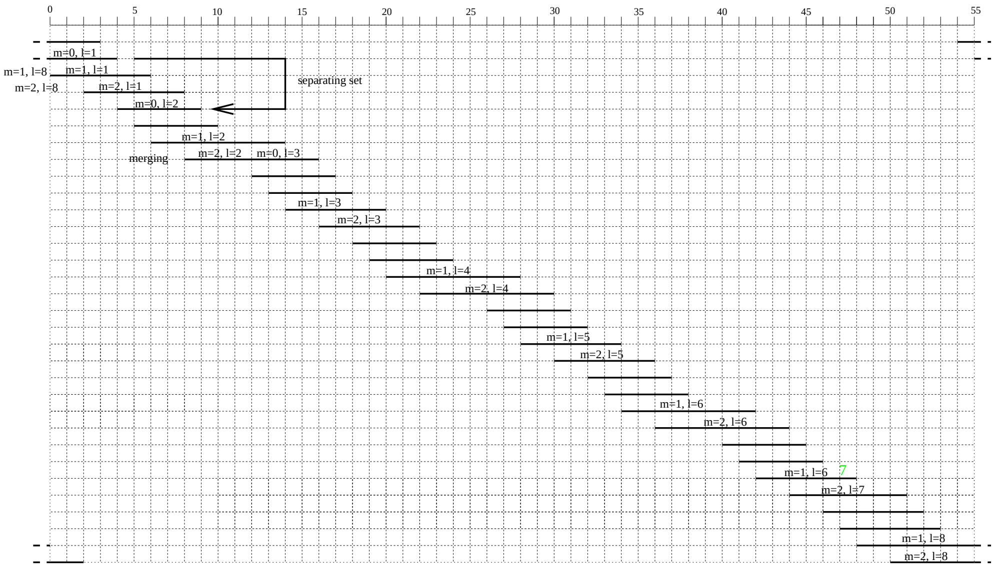

Figure 13: Separating set and marks associated to $T, ..., f(\pi_2(T))$

[8] M. Lindenbaum, A. Bruckstein, On Recursive, O(n) Partitioning of a Digitized Curve into Digital Straight Segments, IEEE Trans. Pattern Anal. and Machine Intelligence 15 (9) (1993) 949–953.

[9] F. Feschet, L. Tougne, Optimal time computation of the tangent of a discrete curve: application to the curvature, in: G. Bertrand, M. Couprie, L. Perroton (Eds.), 8th DGCI, LNCS 1568, Springer-Verlag, 1999, pp. 31–40.