Tangential Cover for Thick Digital Curves

Alexandre Faure and Fabien Feschet*

Univ. Clermont 1, LAIC, F-63172 Aubière, France,
{afaure,feschet}@laic.u-clermont1.fr

Abstract. The recognition of digital shapes is a deeply studied problem. The arithmetical framework, initiated by J.P. Reveillès in [1], provides a great theoretical basis, as well as many algorithms to deal with discrete objects. Among the many available tools, the tangential cover is a powerful one. First presented in [2], it computes the set of all maximal segments of a digital curve and allows either to obtain minimal length polygonalization, or asymptotic convergence of tangent estimations. Nevertheless, the arithmetical approach does not tolerate the introduction of irregularities, which are however inherent to the acquisition of digital shapes. In this paper, we propose a new definition for a class of so-called "thick digital curves" that applies well to a large class of discrete objects boundaries. We then propose an extension of the tangential cover to thick digital curves and provide an algorithm with a $O(n \log n)$ complexity, where n is the number of points of specific subparts of the thick digital curve.

1 Introduction

Digital straight segments are fundamental building blocks in the analysis of shapes or objects. Thus they have been thoroughly studied [3]. Usually digital straightness can be handled with efficient algorithms. Several very deep connections with the continuous case have been obtained. For instance, length [4] or tangent estimation [5] are asymptotically convergent. However, as noted in [6], digital straightness is a very rigid concept and often in practice, the recognized digital segments are short. This might be difficult then to extract significant properties of shapes without a preprocessing step which would suppress all undesirable irregularities in the shape. Those irregularities are not intrinsic, but must be suppressed because of the rigidity of the definition of digital straightness.

Several extensions of digital straightness can be found in the literature, but we here rely on the notion of blurred segment [7] (or α -thick [8] since it is the same notion) due to its geometric properties and its relations with convex hulls.

The primary goal of the present paper is to extend the notion of Tangential Cover introduced in [2,9] which has proven itself an interesting tool in the study of digital shapes. To do this, we proceed in two steps. First, we provide an algorithm for its computation using α -thick segments. The critical part of this

* This work was supported by the French National Agency of Research (ANR) under contract GEODIB.

study is to manage two independent algorithms of Buzer [8,10], such that a low complexity is preserved, since naive connections between both algorithms lead to a quadratic algorithm, at least. Our solution has a complexity of only $O(n \log n)$. Then, in a second part, we precisely define what we call a thick digital curve. The main problem with thick curves is that the classical method must consider all points in the same fashion. Thus, interior and boundary points are treated as if they were similar. Our definition and solution allow us to avoid any confusion between interior and exterior points of thick digital curves. This leads to a solution that cannot have arbitrary cuts in the interior of the thick digital curves. Moreover, this avoids the use of any ordering of those interior points that are in fact useless for the computation. We provide an extension of our algorithm with an unchanged complexity. There, the n factor is no longer the number of points of the thick curves, but the length of the boundary of the digital curve.

This paper is organized as follows. In the first section, we recall all necessary definitions to make the paper self-content. Then, we provide a description of the algorithms of Buzer and precisely describe the necessity of a correct choice of data structures. We conclude this study by providing a good way of merging those two algorithms. Next, we provide the definition of thick digital curves and explain how to extend our algorithm to deal with such curves. We precisely describe one important problem related to the management of interior and exterior boundaries of thick digital curves. This is followed by a description of a simple to compute solution that solves the mentioned drawback. The paper ends by a conclusion and some perspectives.

2 Context and Tools

2.1 α -Thick Segments

In order to handle irregularities for discrete objects recognition and segmentation, the notion of blurred (or fuzzy) segments was introduced by I. Debled-Renesson et al [11]. It relies on the arithmetical definition of discrete lines, first proposed in [1], where a discrete line is the set S of integer points (x, y) which verify the following diophantine inequalities: $\mu \leq ax - by < \mu + \omega$, where $\frac{a}{b}$ is the line slope (usually $\gcd(a, b) = 1$), μ is its lower bound and ω its thickness. Such a line is referred to as $D(a, b, \mu, \omega)$. In this scope, a line is bounded by its two real leaning lines, lower and upper, respectively determined by the equations $ax - by = \mu$ and $ax - by = \mu + \omega - 1$. Leaning points are integer points of S on those lines.

Definition 1. [11] A set S_b of consecutive points ($|S_b| \geq 2$) of an 8-connected curve is a blurred segment with order d if there exists a discrete line $D(a, b, \mu, \omega)$, called bounding, such that all points of S_b belong to D and $\frac{\omega}{\max(|a|, |b|)} \leq d$. If the abscissa interval of S_b is $[0, l - 1]$, D is named strictly bounding for S_b if D possesses at least three leaning points in $[0, l - 1]$, and S_b contains at least one lower leaning point and one upper leaning point.

The parameter d enables the comparison between several blurred lines containing a set of points S , based on their bounding lines' thickness. The least thick of those lines (i.e. the one with the smallest order d) leads to a better-fitting approximation.

An incremental linear time algorithm of discrete curves segmentation into order d blurred segments was given in [11]. It ensured no guarantee of minimality for the built lines, as mentioned in [7]. Thus the obtained segmentation may not be optimal, in the sense that it could lead to oversegmented shapes. The study conducted in [7] suggested a way to overcome this drawback. It introduces the notion of width ν of a blurred segment. In the sequel, let us call "isothetic thickness" of a set of points the minimum value between its vertical and horizontal thicknesses. This notion is illustrated on figure 1.

Fig. 1. The isothetic thickness of this convex hull is the minimum between its vertical (VT) and horizontal (HT) thicknesses

Definition 2. [7] A bounding line of a digital set S_b is said optimal if its isothetic thickness is equal to the isothetic thickness of the convex hull $\text{conv}(S_b)$. S_b is a blurred segment of width ν if and only if its optimal bounding line has an isothetic distance less or equal to ν .

This class of blurred segments with width ν has later been called by Buzer " α -thick segments". To avoid any confusion between blurred segments, fuzzy segments and fuzzy geometry, we will also refer to these segments as α -thick. Recognizing α -thick segments is the problem of computing the isothetic thickness of the convex hull of S_b . The classical method uses Melkman's algorithm [12] in order to incrementally build the successive convex hulls in linear time. The thickness parameter, α , is fixed in advance, so different values can be used for the same digital shape and so α is a user dependent parameter.

2.2 Tangential Cover for Closed Discrete Curves

The tangential cover described in this section deals with naive and standard segments, that is respectively 8- and 4-connected.

Definition 3. A discrete curve C is an ordered set of integer points $(p_1, \dots, p_n)$ such that the real polygonal line passing through them, in order, is a simple polygonal line.

The curve is closed when the next point of p_n is p_1 .

Definition 4. A maximal segment T of a discrete curve C , is a subset $(p_l, \dots, p_r)$ of C such that $\forall l \leq k \leq r, p_k$ belongs to T and such that neither $T \cup \{p_{r+1}\}$ nor $T \cup \{p_{l-1}\}$ is a digital segment, for example according on the definition provided in [1].

Fig. 2. Left : a chromosome digital shape. Middle : the set of its maximal segments. Right : its tangential cover where to set apart overlapping arcs, the radius is increased or decreased accordingly.

For a given discrete curve, consecutive maximal segments overlap, and none is strictly included into another. In [2,9], this property was used to build the set $T(C)$ of all maximal segments, which is called the *tangential cover*. This construction is incremental and has a linear complexity. The incremental linear construction is obtained by using the following process: starting from any point of the 8- or 4-connected curve, points are added in $O(1)$ using the classical algorithm from [13] until this first set is no more a discrete (respectively naive or standard) segment. Then points are substracted, this time using the reverse algorithm, once again in $O(1)$ [2]. Deletions stop when a new point can be added to the right extremity of the current discrete segment, and so on. The process stops when all maximal segments have been constructed.

Each segment is represented by both its left and right ending points, p_l, p_r . Due to the properties of maximal segments, this algorithm leads to a canonical representation of closed discrete curves. In order to graphically represent this construction, $T(C)$ is mapped into the class of circular arc graphs. There, each segment $[p_l, \dots, p_r]$ is associated to the arc between the angles $2l\frac{\pi}{n}$ and $2r\frac{\pi}{n}$, n being the size of the segment. Each radial line starting from the center of the graph represents one point of C , and each arc corresponds to a maximal segment containing the point.

The mapping is reversible if the slopes of the maximal segments are stored. This tool allows us to determine the polygonalization of C with the least number of vertices. This problem is known as min-DSS[3]. It is solved using the circular arcs graph rather than the original curve [14]. Solving the min-DSS problem is equivalent to finding a shortest path in the resulting circular arcs graph, using a linear algorithm described in [14].

3 Adaptation to Thin Digital Curves

3.1 Introduction

A thin digital curve is a discrete curve in the sense of definition 3. As long as we are able to compute a circular arcs graph from the maximal segments of a discrete curve, solving min-DSS remains linear in complexity. Hence it appears to be very interesting to compute the tangential cover of thin digital curves using α -thick segments. In order to adapt the tangential cover to the use of such α -thick discrete segments, we need a fast algorithm to compute the isothetic thickness of a given convex hull. We also need to maintain the convex hull that will go through the curve, searching for every maximal segment. In the next two subsections, we describe the methods used to solve both problems and then show how they must be implemented together to obtain a low complexity.

3.2 Isothetic Thickness Computation

We wish to determine whether a subset of an input polygonal chain is a maximal segment or not, in the sense of α -thick segments. The previous section shows that it is equivalent to the computation of the isothetic thickness of a convex hull. This could be performed $O(\log n)$, with n the number of vertices [8]. Let us present Buzer's approach.

It is obvious that the maximal vertical thickness in a convex hull is always located on the abscissa of a vertex. Computing the height of a vertex is done by subtracting the vertex' ordinate and the ordinate of its projection onto the facing edge or vertex. In a convex hull of size n , the overall computation would cost $O(n \log n)$. Let us have a look at figure 3. Let us denote U the upper hull of the convex hull, L its lower hull, and u' (respectively l') the successive values of the slopes between vertices. Using simple mathematical and convexity properties, it is proven in [8] that the maximum height is located at the unique abscissa where u' and l' intersect. The bottom left figure represents both staircase-like graphs for u' and l' . Let MU (resp. ML) the middle point of U (resp. L). If nU (resp. nL) is the size of the upper (resp. lower) hull, then MU (resp. ML) is the point indexed by $\lfloor nU/2 \rfloor$ (resp. $\lfloor nL/2 \rfloor$). It is sufficient to know the abscissae of MU and ML , and the values of their preceding and following slopes to determine the location of the maximum with respect to MU and ML .

Here is a simple example. On figure 3, $MU.x < ML.x$ and $(MU + 1).x > ML.x$. Then we have $Slope(ML - 1, ML) > Slope(MU, MU + 1)$. In this case, due to the non-increasing (resp. non-decreasing) nature of u' (resp. l'), the two graphs cannot intersect after ML and MU . It is now possible to delete all vertices in U (resp. L) with an abscissa greater than MU 's and ML 's. As those were middle points, half of the convex hull's vertices have been deleted, and another iteration may be performed. All the other configurations for MU , ML and the associated slopes may be handled similarly. The algorithm stops when the intersection is reached (meaning that the maximum has been located), or when the number of remaining vertices is inferior to a desired constant value.

Fig. 3. An example of iteration for the logarithmic computation of the thickness of a convex hull. The upper right box represents the upper and lower hulls of a convex set.

Then, height may be computed in logarithmic time, as this operation will only be called a constant number of times. The computation of the slopes and the comparisons are performed in $O(1)$. We eliminate half of the vertices per iteration. This leads to an overall complexity of $O(\log n)$. The computation of the horizontal thickness is similar.

3.3 Convex Hull Maintenance

This adaptation of the tangential cover requires that we compute the successive convex hulls of the input curve. In other words, we need to add *and* subtract points from convex hulls (i.e. managing dynamic convex hulls) in linear time. This seems difficult and known algorithms such as, Overmars and van Leeuwen [15], are at least of complexity $O(\log^2 n)$ per update. However, we do not need such a complex method, since, as shown in section 2.2, the computing of a tangential cover only requires deletions on the rear end and insertions on the front end. Thus, we are more confronted with incremental and decremental methods than to dynamic one since there is no need for insertions, deletions or updates in the middle of convex hulls.

Let us assume that the input consists in a simple polygonal chain (i.e. not self-intersecting). Recent works of Buzer [10] enable the computing of all successive convex hulls of such chains in linear time. It does so by adapting Melkman's famous algorithm [12] that computes incremental convex hulls. Buzer's algorithm performs linear time incremental *and* decremental management of convex hulls, which is exactly the tool we need. Please refer to [10] for details and complexity proofs. Let us only notice that the fast computation of additions and deletions of points in the convex hull of a simple ordered set of points C is achieved by splitting C into two consecutive subsets. C^+ is used to perform insertions in

front of C , while C^- handles deletions from the back of the set. Both convex hulls of C^- and C^+ are maintained, and $C = C^- \cup C^+$. The algorithm grants a virtual access to C and keeps the computation of all successive convex hulls of a chain of size n within an $\mathcal{O}(n)$ complexity.

Fig. 4. Successive steps in computing an α -thick tangential cover

We now describe how to compute an α -thick tangential cover, using figure 4. C starts with any point of the input chain. At first we have $C = C^+$. Points are added to C^+ as in (1). Its isothetic thickness is computed thanks to the algorithm described in subsection 3.2. When this thickness exceeds α , the corresponding maximal segment is stored, and C^+ becomes C^- . C^+ is initialized with elements added after the last point of the convex hull, if any (2). Now $C = C^-$, as $C^+ \subset C^-$. Points are then deleted from C^- until its thickness is inferior to α (3). Then, points may be added again to C^+ (4). When C^+ is no more included in C^- , the merging $C = C^+ \cup C^-$ must be performed. It is obtained using a "smart" rotating caliper-based method that computes bridges between C^+ and C^- , without explicitly computing C (see Fig. 5). Insertions and deletions are performed in accordance with the variation of C 's thickness. Obviously C^- decreases and C^+ increases. When $C^- \subset C^+$ (5), C^- is deleted, $C = C^+$ and the whole process starts again from (1) until the last maximal α -thick segment has been computed.

3.4 Putting All Together

The two methods presented above are totally independent, and were not specifically designed to work together. For a given input chain S of size n , the computation of all convex hulls costs $\mathcal{O}(n)$, and for a given convex hull of size m , the computing of its isothetic thickness costs $\mathcal{O}(\log m)$. We have to merge those two methods into one algorithm which computes the tangential cover of S . In

Fig. 5. C^+ and C^- are not included into each other. $C = C^+ \cup C^-$. The bridges between C^- and C^+ are computed using a rotating caliper-based method. Right, the virtual C with access to MU and ML .

Fig. 6. The chromosome shape, with four different thickness values. $\alpha = 1, 2, 3, 4$ pixels. The dots represent a good polygonalization for each α , and on the bottom lie the associated tangential covers, represented by their circular arcs graph.

the worst case, the resulting algorithm should be in $O(n \log n)$ complexity. To achieve that, it is of the first importance that we take care of the data structures we use. The vertices of each convex hull are stored as usual in a deque, in counterclockwise order, and the first point is duplicated at the rear end of the deque. For instance, in the case (described above) where $C = C^+ \cup C^-$, C^+ and C^- not being included into each other, C is not explicitly computed but only maintained, in order to keep up with linear complexity. Meanwhile, the isothetic thickness computation algorithm requires a constant time access to the middle of C 's upper and lower hulls. This is clearly the critical point for the algorithm's complexity. So, we must compute in $O(1)$ the number of points contained in the upper and lower hulls and locate the position of the bridge(s) (see Fig. 5). These values should be maintained, also in constant time, during the convex hull maintenance routine (for insertions and deletions). These operations do not

generate any particular difficulties, but are crucial to keep the complexity within the $O(n \log n)$ bound.

The adaptation of the tangential cover to thin digital curves is now complete, with a total complexity of $O(n \log n)$ in the worst case. It allows us to work with a class of curves not reached by algorithms mentioned in the previous section. While some other recent works use similar approaches ([16], [17]), they either only manage the incremental part [17], or use heavy algorithms which seem not as appropriate to solve our particular problem, and are slightly more costly ($O(n \log n)$) [16]. Below, are presented some results of tangential covers and polygonalizations of a chromosome shape, with different thickness values, and their associated tangential covers. This particular shape is 8-connected, but our algorithm works fine with any thin digital curve.

4 Thick Digital Curves

Unformally, a thick digital closed curve is a set of integer points bounded by two distinct thin digital Jordan curves, one being strictly included in the other.

Definition 5. Let C_{int} and C_{ext} be two thin Jordan digital curves such that $C_{int}^\circ \subset C_{ext}^\circ$ where the notation C° denotes the interior of the curve C . The set of points between C_{int} and C_{ext} (both included) is called a *thick digital (closed) curve*.

Thin digital curves are particular thick digital curves where the two sets C_{int} and C_{ext} are equal. Our definition of thick digital curve is related to the notion of a simple polygon with one hole, as seen in Computational Geometry.

Fig. 7. Left : a thick digital closed curve. An interior Jordan thin digital curve, and an exterior Jordan thin digital curve strictly including the interior one. Right : the thickened chromosome shape.

This new definition is crucial in the sense that it avoids any questions about what lies between the two bounding thin curves. Whether it be compact blocks of points, a strongly noisy shape or totally disconnected sets of points, they still belong to the same thick digital curve. There is not even a need for any ordering of those interior points. All that is needed is an ordering of the two bounding thin curves. This can be done with a pre-processing step. Such pre-processings would frequently return two 4- or 8-connected bounding sets of points, but our

algorithm is able to handle two disconnected sets of points without any problem. An example of such a curve is given on figure 7.

The tangential cover of a thick digital curve is computed in the same way as precedently, using the same methods and parameter α , the only difference being that we simultaneously try to add the next point of the interior curve and the next point of the exterior curve to the current segment. If those simultaneous insertions fail (i.e. thickness greater than α after both insertions), then we try to only insert the next point of the exterior curve. If the obtained segment's thickness is still too large, points are deleted from the back of the segment. For each bounding curve, one point at least must remain in the segment, otherwise it would not represent a true part of the thick shape. In case the segment becomes empty, the current cover stops and another one starts with the next two points. This means the input chain is disconnected according to α , and does not constitute one α -thick closed curve. Why choosing to add an exterior point in priority ? For a given shape, its exterior bounding line generally contains more points than its interior one. Let nI (resp. nO) the number of points of the interior (resp. exterior) curve. Total complexity is obviously $O((nI + nO) \log(nI + nO))$. This algorithm gives us a lightning-fast, irregularity-robust method for computing the minimal polygonalization of a thick digital curve. Nevertheless, the approach we describe could lead to some major issues, as shown on figure 8.

Fig. 8. A problem occurring when trying to compute maximal segments on a thick digital curve (we use the thick chromosome shape from figure 7)

In grey are drawn two consecutive maximal segments with thickness value $\alpha = 2$. Light grey represents the parts of the segments which lie outside the curve: these portions are undesirable, as the tangential cover is supposed to stick to the curve. *Segments should not lie outside the curve*, in order to give the best approximation. Moreover, this problem will carry on because the first segment englobes more points from the interior bounding curve than points from the exterior one. This induces an asymmetry that will not be fixed immediately, as shown by the next segment (right figure).

In order to solve this issue, some ideas appeared. Here is one of them: we could use a ratio to force the insertion of exterior points to a segment if the computation of previous segments led to an asymmetry. At each step, we could compute the division between the number of interior and exterior points used, and compare the result to $\frac{nI}{nO}$. If the interior border is strongly favored in a segment, then, for the next one, the algorithm will force the insertion of new

Fig. 9. Maximal segments obtained using the ratio $\frac{nI}{nO}$

points on the exterior border, until the ratio reaches $\frac{nI}{nO}$ again. Same goes if it is the exterior border that is favored. Figure 9 shows that this is an acceptable solution to this issue.

This is the same situation as figure 8. The first maximal segment uses more interior points than exterior ones. So the ratio is greater than $\frac{nI}{nO}$. The computation of the next segment is then forced to only add points from the exterior bounding curve, and the result appears on the middle: not only does the second segment stick way better to the curve, but it also cancels the asymmetry created by the first one. The recognition may then continue as usual (on the right) and leads to a better approximation of the curve.

5 Conclusion

We succeeded in extending the notion of tangential cover to thin and thick digital curves, of which we provide a definition. The latter represents a great improvement for handling irregularities and noise in digital shapes recognition. In fact, we avoid any problems generated by the lack of order within inner points of the digital shape and we do not cut arbitrarily the interior of the thick curve. By only considering the two bounding thin digital curves of a shape, we also provide a very fast algorithm. The $O((nI + nO) \log(nI + nO))$ time complexity we obtain is indeed very low regarding the size of the input shape. In further works, we plan to study the relation between triangulation of a thick digital curve and the tangential cover computation. This could lead to an even better balanced algorithm with respect to the use of both interior and exterior thin digital curves.

References

1. Réveillès, J.P.: Géométrie discrète, calcul en nombres entiers et algorithmique, Thèse d'Etat (1991)
2. Feschet, F., Tougne, L.: Optimal time computation of the tangent of a discrete curve: Application to the curvature. In: Bertrand, G., Couprie, M., Perroton, L. (eds.) DGCI 1999. LNCS, vol. 1568, pp. 31–40. Springer, Heidelberg (1999)
3. Klette, R., Rosenfeld, A.: Digital Geometry: Geometric Methods for Digital Picture Analysis. In: Computer Graphics and Geometric Modeling, Morgan Kaufman, San Francisco (2004)

4. Kovalevsky, V., Fuchs, S.: Theoretical and experimental analysis of the accuracy of perimeter estimates. In: Förster, Ruwiedel (eds.) *Proc. Robust Computer Vision*, pp. 218–242 (1992)
5. Lachaud, J.O., Vialard, A., de Vieilleville, F.: Fast, accurate and convergent tangent estimation on digital contours. *Image and Vision Computing* 25(10) (2006)
6. Bhowmick, P., Bhattacharya, B.: Fast polygonal approximation of digital curves using relaxed straightness properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(9), 1590–1602 (2007)
7. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition in linear time. In: Andrès, É., Damiani, G., Lienhardt, P. (eds.) *DGCI 2005*. LNCS, vol. 3429, pp. 371–382. Springer, Heidelberg (2005)
8. Buzer, L.: Digital line recognition, convex hull, thickness, a unified and logarithmic technique. In: Reulke, R., Eckardt, U., Flach, B., Knauer, U., Polthier, K. (eds.) *IWCIA 2006*. LNCS, vol. 4040, pp. 189–198. Springer, Heidelberg (2006)
9. Feschet, F.: Canonical representations of discrete curves. *Pattern Anal. Appl.* 8(1–2), 84–94 (2005)
10. Buzer, L.: Computing multiple convex hulls of a simple polygonal chain in linear time. In: *23rd European Workshop on Computational Geometry* (2007)
11. Debled-Rennesson, I., Rémy, J.L., Rouyer-Degli, J.: Segmentation of discrete curves into fuzzy segments. In: *9th Intl. Workshop on Combinatorial Image Analysis*. *Electronic Notes in Discrete Mathematics*, vol. 12, pp. 122–137. Elsevier, Amsterdam (2003)
12. Melkman, A.A.: On-line construction of the convex hull of a simple polyline. *Inf. Process. Lett.* 25(1), 11–12 (1987)
13. Debled-Rennesson, I., Reveillès, J.P.: A linear algorithm for segmentation of digital curves. *IJPRAI* 9(4), 635–662 (1995)
14. Feschet, F., Tougne, L.: On the min dss problem of closed discrete curves. *Discrete Applied Mathematics* 151(1–3), 138–153 (2005)
15. Overmars, M.H., van Leeuwen, J.: Maintenance of configurations in the plane. *J. Comput. Syst. Sci.* 23(2), 166–204 (1981)
16. Nguyen, T.P., Debled-Rennesson, I.: Curvature estimation in noisy curves. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) *CAIP 2007*. LNCS, vol. 4673, pp. 474–481. Springer, Heidelberg (2007)
17. Roussillon, T., Tougne, L., Sivignon, I.: Computation of binary objects sides number using discrete geometry, application to automatic pebbles shape analysis. In: Cucchiara, R. (ed.) *14th International Conference on Image Analysis and Processing*, pp. 763–768. IEEE Computer Society, Los Alamitos (2007)