A comparative Evaluation of Length Estimators of digital Curves

David Coeurjolly, Reinhard Klette

To cite this version:


HAL Id: hal-00185088
https://hal.archives-ouvertes.fr/hal-00185088
Submitted on 6 Nov 2007
A Comparative Evaluation of Length Estimators of Digital Curves

David Coeurjolly and Reinhard Klette

Abstract—This paper compares previously published length estimators in image analysis having digitized curves as input. The evaluation uses multigrid convergence (theoretical results and measured speed of convergence) and further measures as criteria. This paper also suggests a new gradient-based method for length estimation, and combines a previously proposed length estimator for straight segments with a polygonalization method.

Index Terms—Length estimator, digital geometry, curve length, multigrid convergence.

1 INTRODUCTION

The digitization of curves or boundaries has been studied in image analysis for about 40 years [1]. Since these first studies, many algorithms have been proposed to estimate the length of a digitized curve. Some approaches are based on local metrics, such as the weighted metrics, other approaches are based on polygonalizations of digital curves, e.g., directed on subsequent calculations of maximum-length digital straight segments (DSSs) or of minimum length polygons (MLPs). This paper compares previously published length estimators by defining and applying different performance measures for a sample set of 2D curves. We also propose and include a new length estimator based on calculated gradients.

The computational problem of estimating the length of a digital curve based on maximum-length DSS segmentation is as follows: the input is a sequence of chain codes \( i(0), i(1), \ldots \) with \( i(k) \in A = \{0, \ldots, 7\}, k \geq 0 \). An offline algorithm takes finite words \( u \in A^* \) as input, and performs the required calculations, e.g., to decide whether \( u \) is a DSS or not. An online algorithm reads successive chain codes \( i(0), i(1), \ldots \) and always provides a result up to the most recent input value \( i(n) \), e.g., it decides whether \( i(0), i(1), \ldots, i(n) \) is still a DSS, and if not, then it initializes a new DSS with \( i(n-1) \). After completing one maximum-length DSS, its length estimate (typically, the Euclidean distance between its end vertices) is added to the length estimator of the curve. An offline algorithm is linear if it runs in \( O(n) \) time, i.e., it performs at most \( O(n) \) basic computation steps for any input word \( u \in A^* \). An online algorithm is linear iff it uses on the average a constant number of operations for any incoming chain symbol code.

The classification into online and offline algorithms may also be applied to other procedures for length estimation. An obvious benefit of local metrics based approaches is that they support linear online implementations. This method has frequently been suggested in the image analysis literature for length estimation, combined with proposals of local weights to improve these estimations. However, it is known that these methods are only of limited use if multigrid convergence is applied as a selection criteria.

DSS-based polygonalization is a popular method in image analysis, allowing us to transform digital boundaries into polygonal objects (see Fig. 1a). Linear offline algorithms for DSS recognition were published in 1981 in [5] and in 1982 in [8]. A linear offline algorithm for cellular straight segment recognition, based on convex hull construction, is briefly sketched in [7]. Two linear online algorithms for DSS recognition were published in 1982 in [6]; one of them is an online version of the offline algorithm published in [5].

The general problem of decomposing a digital curve into a sequence of DSSs, which includes DSS recognition as a subproblem, is discussed in, e.g., [15], [16], [20], [27]. Obviously, linear online DSS recognition algorithms will support linear decomposition algorithms, but the application of a linear offline algorithm for input-sequences of increasing length, i.e., first for \( i(0) \), then for \( i(0)(1) \), then for \( i(0)(1)(2) \), etc., leads to quadratic runtime behavior.

MLP-based polygonalization provides a third method, which is not yet of widespread use in image analysis. An MLP-approximation [22], [28] calculates a minimum-length polygon circumscribing a given (closed) inner boundary (given by a sequence of chain codes) and being in the interior of an outer boundary (typically, in Hausdorff-Chebyshev distance 1 to the inner boundary). This polygon is also known as relative convex hull [3] and calculations of relative convex hulls have a history in computational geometry and robotics (see Fig. 1b).

As a fourth method, we also introduce gradient-based length estimation in this paper that may be seen as an extension of the DSS-based approach. The notions online, offline, or linear time apply for algorithms following any of these four design strategies.

In this paper, we present a comparative evaluation of length estimators covering these four types of strategies. We are especially interested in evaluating these algorithms (and underlying methodologies) with respect to the accuracy of length estimation. Multigrid convergence is one option of characterizing this accuracy, and experimental studies provide another way for performance testing. Our experiments are directed on illustrating accuracy and stability of the chosen algorithms on convex and nonconvex curves (see Fig. 2 for the used test data which had been proposed in [27]). These given Jordan curves of known length are digitized for increases in grid resolution, allowing to study and illustrate experimental multigrid convergence.

In our experiments, we digitize these curves in grids varying between \( 30 \times 30 \) and \( 1,000 \times 1,000 \). For the digitization of a planar Jordan curve (up to a given grid resolution), we may adopt any of the digitization models known in the image analysis literature.

Fig. 1. (a) Segmentation of a 4-path into a sequence of maximum-length DSSs. (b) MLP-approximation between two polygonal boundaries.
We assume an orthogonal grid with grid constant $0 < \theta \leq 1$ in the Euclidean plane $\mathbb{R}^2$, i.e., $\theta$ is the uniform spacing between grid points parallel to one of the coordinate axes. Let $r = 1/\theta$ be the grid resolution and the $r$-grid $\mathbb{Z}_r^2$ has resolution $r$, defined by $r$-points whose coordinates are $(i \cdot \theta - j \cdot \theta)$, with $i, j \in \mathbb{Z}$. Now, we consider a Jordan curve $\gamma : [0, 1] \rightarrow \mathbb{R}^2$, being the topological frontier of a set $S$ in the Euclidean plane. Let $D_r(\gamma)$ be an $r$-digitization of $\gamma$ in $\mathbb{Z}_r^2$ (see Fig. 1):

**Definition 1.** In this paper, an $r$-digitization of curve $\gamma$ is one of the following:

1. the cyclic 4-path $\sigma_{\text{c}}(\gamma)$ or 8-path $\sigma_{\text{s}}(\gamma)$ of $r$-grid points following an $r$-grid-intersection digitization of $\gamma$, see [2] for the original definition of grid-intersection digitization with $r = 1$;
2. the cyclic 4- or 8-path following vertices of $r$-grid squares in the frontier of the Gauss digitization $G_r(S)$ of set $S$ in the $r$-grid, where $G_r(S)$ is the union of all $r$-grid squares having their centroids in the given set $S$, or
3. the closed difference set between outer and inner Jordan digitization $J_2^+(S)$ and $J_2^-(S)$, i.e., $\text{cl}(J_2^+(S) \setminus J_2^-(S))$, where $J_2^+(S)$ is the union of all $r$-grid squares having a nonempty intersection with the given set $S$ and $J_2^-(S)$ is the union of all $r$-grid squares contained in the topological interior of the given set $S$.

See, e.g., [30] for more details and historic citations for these digitization methods. In the following, let $L(\gamma)$ be the length of the curve $\gamma$. We denote by $E$ an estimated property. Assume that $E$ is defined for digitizations $D_r(\gamma)$, for $r > 0$ and all curves $\gamma$ in a class $\Gamma$ of curves. In this paper, we only consider the class $\Gamma$ of all Jordan curves in the Euclidean plane.

**Definition 2.** The estimated property $E$ is said to be multigrid convergent toward $L$ with respect to digitization model $D_r$ and class $\Gamma$ iff for any curve $\gamma \in \Gamma$ we have that $E(D_r(\gamma))$ converges to $L(\gamma)$, for $r \to \infty$. More formally: $|E(D_r(\gamma)) - L(\gamma)| \leq \kappa(r)$ with $\lim_{r \to \infty} \kappa(r) = 0$. The order $O(1/\kappa(r))$ denotes the speed of this convergence.

Multigrid convergency of estimated properties is a standard method in numerical mathematics for discrete versions of Euclidean plane. Let $\alpha$ be the speed of convergence. It follows that the perimeter of $P_m$ equals $n_m \cdot p_m$ and $\alpha(n_m) \approx \frac{2n_m}{p_m^2 n_m}$ for $m \geq 1$. The function $\alpha(n)$ defines the speed of convergence, which is linear in this case. Altogether, the estimated length converges toward the true length with respect to regular $n$-gon approximation and the class of all circular curves.

### 2 Local Metrics

Local metrics were historically the first attempts toward a solution of the length estimation problem in image analysis. These algorithms apply to digital curves defined by options 1 or 2 in Definition 1 and can be viewed as shortest path calculations in weighted adjacency graphs of pixel locations. Weights have been designed with the intention of approximating the Euclidean distance. For example, horizontal and vertical moves in the orthogonal $r$-grid may be weighted by $1/r$ and diagonal moves may be weighted by $\sqrt{2}/r$. More generally, a chamfer metrics definition first lists elementary moves and then associates weights to each move, see [12], [13], [18].

In order to make length estimation as accurate as possible, the use of statistical analysis has been suggested to find those weights that minimize the mean square error between estimated and true length of a straight segment. For example, [13] presents a best linear unbiased estimator (BLUE for short) for straight lines and defines the following length estimator (CM from “chessboard metric” since that is the name of the local metric it is based on):

$$E_{\text{CM}}(\alpha_{\text{CM}}(\gamma)) = \frac{1}{r} \cdot (0.948 \cdot n_i + 1.345 \cdot n_d),$$

where $n_i$ is the number of isothetic steps and $n_d$ of diagonal steps in the $r$-grid. We include this estimator into our comparative study. It fails to be multigrid convergent for length estimations of digitized arcs or curves (see Section 6).

We also implemented the cornercount length estimator [9] (CC for short) and include this estimator into our comparisons:

$$E_{\text{CC}}(\alpha_{\text{CC}}(\gamma)) = \frac{1}{r} \cdot (0.980 \cdot n_i + 1.406 \cdot n_d - 0.091 \cdot n_{\text{num}}),$$

where also another type of elementary steps is counted: $n_{\text{num}}$ is the number of odd-even chaincode transitions in a digital arc or curve and the weights have been optimized (again as a best linear unbiased estimator).

### 3 Polygonal DSS-Approaches

Many algorithms have been discussed for the DSS recognition problem. Digital curves are again defined by options 1 or 2 in Definition 1. DSS-approaches are based on characterizations of digital lines, such as syntactic chain code properties [5], [8], arithmetical properties defining tangential lines [20], properties of...
featural regions in the (dual) parameter space [14], [16], or use linear programming tools such as the Fourier-Motzkin algorithm [21]. All these algorithms present a solution for deciding whether a given sequence of r-grid points is a DSS or even for segmenting a digital curve into a sequence of maximum-length DSSs. The length estimator EDSS is then defined by the length of the obtained polyline. Note that DSS-approximations are not uniquely defined; they vary from method to method and depend, in general, upon the chosen start point and the orientation of curve tracing.

In our comparative study, we include two representative implementations of DSS-based length estimators: If the digital curve is defined as an 8-curve, we use the Debled-Reveilles algorithm [20] and call it the EDR-estimator. In our implementation, we strictly follow the algorithm as described in [20]. If the digital curve is defined as a 4-curve, we consider a length estimator based on Kovalevsky’s algorithm [15] and call it the EYK-estimator. We use the algorithm as implemented for, and detailed, in [27]. These two DSS-based length estimators are known to be multigrid-convergent for convex Jordan curves γ [19], [30], [32].

Given a simply-connected compact set S in the Euclidean plane and a grid resolution r, the r-frontier ∂G_r(S) of S is uniquely determined (i.e., the frontier of the Gauss digitization G_r(S) with respect to the topology of the Euclidean plane). Note that an r-frontier may consist of several nonconnected curves even in the case of a bounded convex set. A set S is r-compact if and only if there is a number r_0 > 0 such that ∂G_r(S) is just one (connected) curve, for any r ≥ r_0.

**Theorem 1** [30]. Let S be a convex, r-compact polygonal set in IR^2. Then, there exists a grid resolution r_0 such that for all r ≥ r_0, any DSS approximation of the r-frontier ∂G_r(S) is a connected polygon with perimeter p_r satisfying the inequality

\[ |\mathcal{L}(\partial(S)) - p_r| \leq \frac{2\pi}{r} \left( \varepsilon_{DSS}(r) + \frac{1}{\sqrt{2}} \right). \]  

(4)

This theorem and its proof can be found in [30]. The proof is based to a large extent on material given in [19]. The value of r_0 depends on the given set and ε_{DSS}(r) ≥ 0 is an algorithm-dependent approximation threshold specifying the maximum Hausdorff-Chebyshev distance (generalizing the Euclidean distance between points to a distance between sets of points) between the r-frontier ∂G_r(S) and the constructed (not uniquely specified—see comments above) DSS approximation polygon. Assuming ε_{DSS}(r) = 1/r, it follows from (4) that the upper error bound for DSS approximations is characterized by

\[ \frac{2\pi}{r^2} + \frac{2\pi}{r \sqrt{2}} \approx \frac{4.5}{r} \quad \text{if} \quad r \gg 1 \quad (\text{i.e., } r \text{ is large}). \]  

(5)

1. Let κ(r) = 2π/r^2 + 2π/r √2. Then, it follows that κ(r) → π√2 as r → ∞.

Grid resolution 1/r is assumed in the chord property defined in [2], where a DSS is assumed to be a finite 8-path. In the case of using cell complexes, it is appropriate to consider a finite 4-path as a DSS iff its main diagonal width is less than √2, see [11], [17], [19].

Dorst and Smeulders [16] specified a most probable original (MPO, for short) estimation method with superlinear convergence O(r^{-1.5}) of asymptotic length estimation for the case of digitized straight lines. Let n be the length of a digital straight segment and p/q the best possible rational estimate of its slope. Then,

\[ E_{MPO}(\sigma_s(\gamma)) = \frac{1}{r} \cdot (\alpha \sqrt{1 + (p/q)^2}). \]  

(6)

Of course, it can also be applied, in addition, to a maximum-length DSS segmentation method, e.g., defining a DSS-based length estimator DR-DSS-MPO: apply Debled-Reveilles algorithm for DSS segmentation followed by summing all MPO-length-estimates of all DSSs.

### 4 POLYGONAL MLP-APPROACHES

MLP-based length estimators consider a situation where a given simple digital curve C is described by two discrete curves γ_1 and γ_2, bounding sets S_1 and S_2, respectively, such that S_1 is contained in the interior S'_1 of set S_1 and C is contained in B = S_1 \ S_2. Digital curves are defined, in this case, by option 3 in Definition 1. S_1 and S_2 are given as J_r(S_1) and J_r(S_2). The task consists of calculating the MLP which is contained in B and circumscribes γ_2. The length estimator E_{MLP} is then defined by the length of this (uniquely defined [22], [25]) MLP.

In our comparative study, we include two representative implementations of MLP-based length estimators: the grid-continua MLP approach of [22], [25] has been derived for the model of using inner and outer Jordan digitization, and defining B to be the difference set between outer and inner Jordan digitization. We use the MLP algorithm as reported in [27]. We call it the E_{SZ-MLP} estimator.

As another MLP-method, we include the approximation-sausage MLP approach of [28], [31] defining the E_{AS-MLP} estimator which actually involves a parameter δ, with 0 < δ ≤ 5/r. This parameter specifies a polygon A_{δ}(S) which is referred to as the approximating sausage of the r-frontier (see Fig. 4a for δ = 1/2). The width of such an approximating sausage depends on the value of δ. An AS-MLP curve for approximating the boundary of S is defined as being a shortest closed curve γ_{AS}(S) lying entirely in the interior of the approximating sausage A_{δ}(S) and encircling the internal boundary of A_{δ}(S) (see Fig. 4b). It follows that such an AS-MLP curve γ_{AS}(S) is uniquely defined and that it is a polygonal curve defined by finitely many straight segments. We have taken δ = 5/r for the experimental evaluation.

Both cited MLP-based length estimators are known to be multigrid-convergent for convex Jordan curves γ [22], [28], [31].
The basic idea of this optimal algorithm is illustrated in Fig. 5: Discrete tangent parameters at a point \( p \) of a discrete curve will not change for some “neighboring” pixels. More precisely, we consider the discrete tangent at point \( p \) and the first rejected point is denoted by \( q \) (i.e., the first point that does not belong to the DSS, see Fig. 5, top). Then, we compute a DSS from \( q \) to \( p \) and define point \( l \) to be the first rejected point during this process (Fig. 5, middle). Finally, Feschet and Touigne [26] prove that for all pixels between \( p \) and the middle \( m \) of segment \([pq]\) (Fig. 5, bottom), the discrete tangent parameters do not change. They finally propose an efficient algorithm whose time complexity is linear in the number of pixels that computes the discrete tangent at each point of the discrete curve.

This Feschet-Touigne algorithm computes a discrete tangent at each 0-cell of a given curve (an alternating sequence of 0 and 1-cells). Thus, we can use this for computing the discrete normal vector at each 0-cell as a unit vector perpendicular to the discrete tangent.

We define the normal vector \( \bar{n} \) associated to a 1-cell as the mean vector of both vectors calculated at its two neighboring 0-cells. We also define an elementary normal vector \( n_e \) to a 1-cell as the unit vector perpendicular to this 1-cell (see Fig. 6). Hence, the discrete version of (9) is:

\[
\epsilon_{TAN}(D_1(\gamma)) = \sum_{s \in S} \bar{n}(s) \cdot n_e(s).
\]
3D extension: May the approach be extended to length estimation of digital curves in 3D space?

Table 1 explains the situation. The convergence speed is known to be linear for DR-DSS, VK-DSS, SZ-MLP, and AS-MLP; see the cited Theorems 1, 2, 3, and 4.

We consider two measures: 1) the relative error in percent between estimated and true curve length and 2) for DSS- and MLP-approaches, also a trade off measure defined as the product of relative error times the number of generated segments (in [25], [27], it has been called the efficiency of convergence). Test data (as used in [27]) are shown in Fig. 2.

In Fig. 7, experimental convergence is evident for all methods. However, in case of methods CM and CC, we have convergence to a false value! Errors are calculated for all curves, transformed into a mean value for a given grid size, and curves are generated by sliding means (of 30 values) along different grid sizes. We show a decimal and a logarithmic scale of errors. DR-DSS-MOP slightly overestimates the length in comparison to DR-DSS.

The trade off measure is presented in Fig. 8 for polygonal approaches only. Again, values are calculated for all curves, transformed into a mean value for a given grid size, and curves are generated by sliding means (of 30 values) along different grid sizes.

As an additional test, we also rotated a square of fixed size in a grid of resolution 128. Fig. 9a shows the behavior of estimated perimeters for such a rotated square curve, for different estimation algorithms. The ideal situation would be a totally orientation-independent estimator and all methods besides CM and CC approximate this reasonably well.

Fig. 9b summarizes the runtimes of polygonalization-based estimators compared to a local-metric estimator. Obviously, both local metric algorithms are the fastest. Method SZ-MLP in the implementation of [27] provides the fastest global estimator, but VK-DSS and DR-DSS are close. The AS-MLP algorithm has not yet been optimized and faster implementations might be possible. Theoretically, it is known that the TAN method allows a linear asymptotic runtime implementation [26]. However, the used program showed quadratic runtime. Further algorithmic optimization is needed.

7 CONCLUSIONS

The experiments show that both local-metrics approaches are not multigrid convergent (already for the test data set), but all five global

<table>
<thead>
<tr>
<th>Method</th>
<th>multigrid</th>
<th>discrete</th>
<th>unique</th>
<th>3D extension</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR-DSS</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>[32], [20]</td>
</tr>
<tr>
<td>DR-DSS-MPO</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>-</td>
<td>[20]+[16]</td>
</tr>
<tr>
<td>VK-DSS</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>-</td>
<td>[15]</td>
</tr>
<tr>
<td>SZ-MLP</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
<td>[29], [3], [22]</td>
</tr>
<tr>
<td>AS-MLP</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>-</td>
<td>[28], [31]</td>
</tr>
<tr>
<td>TAN</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>-</td>
<td>[4], this article</td>
</tr>
<tr>
<td>Local metrics</td>
<td>no</td>
<td>depends</td>
<td>yes</td>
<td>[24]</td>
<td>[12], [13]</td>
</tr>
</tbody>
</table>

Fig. 7. Evident multigrid convergence of length estimators: (a) sliding means of relative errors, also shown in (b) a logarithmic scale.

Fig. 8. Trade off value diagrams for polygonal approaches.
approximation methods confirm known theoretical convergence results by experimental evidence. A mathematical analysis is (still) unpublished which describes local methods that cannot achieve multigrid convergence if input data are "sufficiently complex," e.g., not just isothetic rectangles (actually there was a seminar presentation at Dagstuhl/Wadern in April 2002 by M. Tajine from Strasbourg showing that local methods cannot ensure multigrid convergence in specified situations of measurements—however, this has not yet been published). The potential failure of local methods is well-known in mathematics for 3D surface area estimates since the end of the 19th century, see, e.g., cited work of Schwarz in [34].

Interestingly, the increase in runtimes of the studied polygonal methods is only minor compared to that of local-metric algorithms. Hence, the use of a (incorrect) local-metric algorithm is also not justified by a runtime argument. The choice of a global method may depend on preferences defined by the context of an image processing software package and the authors can recommend any of the five studied global methods. Studies on test data might be useful for selecting the most efficient implementation for a given application context.

Originally, the authors also intended to use a third measure for comparison, the minimum value \( r_0 \) of grid resolution \( r \) such that a method estimates \( \pi \) the first time within the error interval defined by (1) (i.e., where a circular region is digitized in a grid with edge length \( 1/r \)), what might be called the Archimedes-Hui constant of the algorithm implementing a method. However, due to oscillations of calculated estimations, results of an algorithm may be outside of this interval again for \( r > r_0 \) for an Archimedes-Hui constant \( r_0 \) of this algorithm and just using a sliding mean can also not be recommended because of unsecured knowledge on the general behavior of the measured error sequence. A methodically correct introduction of such an Archimedes-Hui constant remains an open problem.

**Acknowledgments**

At the time this research was done, D. Coeurjolly was a member of the ERIC Laboratory (Université Lumière Lyon 2).

**References**


For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.