Evaluation of Curve Length Measurements

Reinhard Klette and Ben Yip
CITR Tamaki, Auckland University
Tamaki Campus, Building 731
Auckland, New Zealand
{r.klette@, yyip003@cs.}auckland.ac.nz

Abstract

This paper compares two techniques for measuring the length of a digital curve. Both techniques (digital straight segment approximation, minimum length polygon) are known to be convergent estimators. Theoretical convergence results are cited. The main focus is on experimental evaluation: several measures are defined, applied and discussed. Test sets are digitized for consecutive resolutions.

1. Two methods for length estimation

This paper presents an evaluation and comparison of two methods for estimating the length of a digital curve. The *DSS method* is based on a partition into digital straight segments [3, 6], and the *MLP method* is characterized by the calculation of the minimum-length polygon [7], assuming a given digital curve as input in both cases.

Let $r \geq 1$ be the *grid resolution* defined as being the number of grid points per unit. We consider *r-grid points* $g^r_{i,j} = (i/r, j/r)$ in the euclidean plane, for integers i, j. Any r-grid point is assumed to be the center point of an *r-square* with *r-edges* of length $1/r$ parallel to the coordinate axes, and *r-vertices*. *Cells* are either r-squares, r-edges or r-vertices. The intersection of two cells is either empty or a joint *side* of both cells. We consider a non-empty finite set K of cells such that for any cell in K it holds that any side of this cell is also in K. Such a set K is a special finite *euclidean complex* [5], and thus an *abstract cellular complex* [1, 5] including straightforward definitions of a bounding relation for pairs of cells, and dimensions of cells.

Digital curves g are defined for $n \geq 1$: **(i)** An *edge-curve* $g = (v_0, e_0, v_1, e_1, ..., v_n, e_n)$ is a sequence of vertices v_i and edges e_i, for $0 \leq i \leq n$, such that vertices v_i and v_{i+1} are sides of edge e_i, for $0 \leq i \leq n$ and $v_{n+1} = v_0$. It is *simple* iff each edge of g has exactly two bounding vertices in g. **(ii)** A *square-curve* is a sequence $g = (e_0, s_0, e_1, s_1, ..., e_n, s_n)$ of r-edges e_i and r-squares s_i, for $0 \leq i \leq n$, such that edges e_i and e_{i+1} are sides of square s_i, for $0 \leq i \leq n$ and $e_{n+1} = e_0$. It is *simple* iff each square of g has exactly two bounding edges in g. The union of all squares contained in a simple square-curve is called its *tube*.

We have to specify digitization models for our evaluation of the DSS and MLP method. Let S be a set in the euclidean plane, called *real preimage*. The set $I_r(S)$ is defined to be the union of all those r-squares completely contained in the interior of S. The set $O_r(S)$ is the union of all those r-squares having a non-empty intersection with set S. Finally, let $C_r(S)$ be the union of all those r-squares whose center point $g^r_{i,j}$ is in S (see Fig. 1).

The boundary $\partial C_r(S)$ is the *r-frontier* of S, and this notion corresponds to the definition of simple edge-curves. The set $(O_r(S) \setminus I_r(S)) \cup \partial I_r(S)$ is the *r-boundary* of S, and this notion corresponds to the definition of simple square-curves. The Hausdorff distance d_∞ between $\partial O_r(S)$ and $\partial I_r(S)$ is not always $1/r$.

The *DSS method* calculates a sequence of consecutive maximum-length digital straight segments for a simple edge-curve, and its result depends upon starting

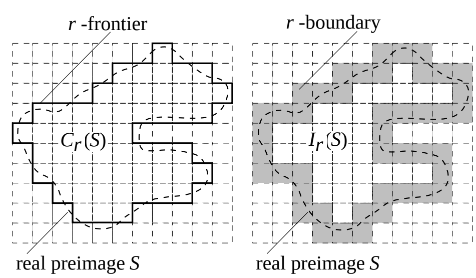

Figure 1. Digital curves and real preimage.

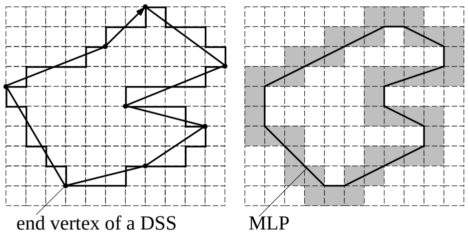

Figure 2. DSS approximation and MLP.

point and orientation of the traversal [1, 3]. The *MLP method* calculates a minimum-length polygon within the tube of a simple square-curve, where this polygon is required to have a non-empty intersection with any r-square contained in this simple digital curve [7]. The MLP method leads to a uniquely specified result, *the* minimum-length polygon [7]. See Fig. 2 for examples of such approximating polygons. The DSS or the MLP *length estimator* of the length of a digital curve is defined to be the total length of the calculated polygonal chain. Algorithms for calculating a DSS approximation or the MLP are detailed in [1], and the experimental evaluation reported in this paper is based on these algorithms (implemented in C++ on a PC).

2. Convergence Theorems

For both techniques it is known that the measured length of a digital curve converges towards the true value if a bounded, convex, polygonal or smooth (ie. boundary is C^1) subset of the euclidean plane is digitized with increasing grid resolution [2, 4, 7].

In the following theorem, value $\varepsilon_{DSS}(r) \geq 0$ is an algorithm-dependent approximation threshold specifying the maximum Hausdorff distance d_2 between the r-frontier $\partial C_r(S)$ and a DSS polygon. Its 'classical' value is $1/r$, see [6].

Theorem 1 (Kovalevsky/Fuchs 1992) *S be a bounded, convex, smooth or polygonal set. Then there exists a grid resolution r_0 such that for all $r \geq r_0$ it holds that any DSS approximation of the r-frontier $\partial C_r(S)$ is a connected polygon with perimeter l_r and*

$$|Perimeter(S) - l_r| \leq \tfrac{2\pi}{r}\left(\varepsilon_{DSS}(r) + \frac{1}{\sqrt{2}}\right) .$$

This theorem and its proof may be found in [2] but the proof was actually fully based on material given in [4]. The following theorem is basically a citation from [7] and specifies the asymptotic constant for MLP perimeter estimates. In [7] only powers of two have been considered as grid resolution r. However, it may be concluded from the material given in [7] (eg. Theorem 4.15 and Lemma 4.3) that the following theorem may actually be stated for arbitrary grid resolutions $r \geq 1$.

Theorem 2 (Sloboda et al. 1998) *Let S be a bounded convex set such that its boundary is contained in the r-boundary of S, for $r \geq 1$. Then it holds that the MLP approximation of the r-boundary is a connected polygonal curve with length l_r and*

$$l_r \leq Perimeter(S) < l_r + \frac{8}{r} .$$

Let $\varepsilon_{DSS}(r) = 1/r$. It follows that the upper error bound for DSS approximations is characterized by

$$\frac{2\pi}{r^2} + \frac{2\pi}{r \cdot \sqrt{2}} \approx \frac{4.5}{r}$$

and the upper error bound for MLP approximation is characterized by $8/r$. This theoretical ratio of about 1:2 coincides with our experimental studies, see next section.

3. Experiments

We report about experiments using six sets S as shown in Fig. 3: the function graph of the *sinc function* $y = sin(16\pi \cdot x)/(64\pi \cdot x)$ within a bounded interval symmetric to the y-axis, a square rotated by 45°, a square rotated by 22.5°, a halfmoon generated by two overlapping circles of identical size, a circle, and the yin-part in the Chinese yinyang symbol. All sets have been digitized for (at least) all grid resolutions from $r = 32$ to $r = 1024$. In this section we abstain from parameter r to simplify the notations.

A grid square is contained in $C(S)$ iff its midpoint is in S. The frontier $\partial C(S)$ is used as input for the DSS algorithm. We use an approximative scheme for $I(S)$: a square is contained in $I(S)$ iff all of its four vertices are in S ("four-vertex scheme"). The MLP algorithm [1] only requires that $\partial I(S)$ is available as input.

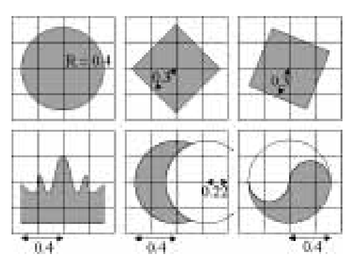

Figure 3. Sets within an area of 1 unit × 1 unit.

Size	256x256	512x512	1024x1024	2048x2048
$N_{C(S)}$	474	954	1922	3838
N_{DSS}	46	74	121	187
$N_{I(S)}$	474	946	1906	3834
N_{MLP}	80	124	199	330
E_{DSS}	3.061	2.466	1.605	1.432
E_{MLP}	6.006	4.390	3.037	2.018
θ_{DSS}	140.80	182.49	194.18	267.76
θ_{MLP}	480.45	544.42	604.27	666.00
T_{MLP}	2.048	4.199	8.398	17.206
T_{DSS}	4.710	9.421	19.254	38.099

Table 1. Yin curve of the Yinyang symbol.

Size	256x256	512x512	1024x1024	2048x2048
$N_{C(S)}$	860	1732	3468	6940
N_{DSS}	4	4	4	4
$N_{I(S)}$	852	1724	3460	6932
N_{MLP}	8	8	8	8
E_{DSS}	0.559	0.102	0.102	0.103
E_{MLP}	1.102	0.372	0.238	0.170
θ_{DSS}	2.24	0.41	0.41	0.41
θ_{MLP}	8.82	2.98	1.90	1.36
T_{MLP}	2.048	4.199	8.398	16.387
T_{DSS}	5.274	10.548	21.098	42.196

Table 2. Square rotated 45 degrees.

For the resulting isothetic (ie. edges parallel to coordinate axes) polygons the DSS partition and the MLP were calculated allowing to compute the DSS estimate and the MLP estimate of the perimeter of region S with respect to grid resolution r. Of course, $\partial C(S)$ may also be used as input for the MLP algorithm, and we call the result the *MLP_C perimeter estimate*.

Let $N_{C(S)}$ be the number of vertices of $\partial C(S)$ (the input sequence of the DSS algorithm), let $N_{I(S)}$ be the number of vertices of $\partial I(S)$ (the input sequence of the MLP algorithm), let N_{DSS} be the number of vertices of the calculated DSS polygon, and let N_{MLP} be the number of vertices of the calculated MLP. The *errors* E_{DSS} and E_{MLP} are the length errors in percent compared to the true perimeter of set S (length of the curve defining the boundary of S). The *effectiveness of the approximation* (as defined in [7]) is $\theta_{DSS} = E_{DSS} \cdot N_{DSS}$ or $\theta_{MLP} = E_{MLP} \cdot N_{MLP}$, which specifies an interesting trade-off measure: this product of error times number of segments informs about the efficiency of the convergence. If this product decreases faster or increases slower for algorithm A in comparison to a second algorithm B then algorithm A is more efficient in achieving reduced errors without generating too many new segments. [1] Finally, we also measure the computing time T_{DSS} and T_{MLP}. In our tables time is specified in multiples of 10^{-3} seconds. Tables 1 and 2 show selected results (for a few values of r) for two curves. In these tables, the values in row *Size* correspond to the resolution r, ie. size is equal to $r \times r$.

[1] Smaller values of this measure characterize a "more efficient" approximation, ie. the name "inverse effectiveness measure" might be more appropriate. However, we follow [7] because the given name is also reasonable.

Size	68x68	69x69	70x70	71x71	72x72
$N_{C(S)}$	80	80	88	92	88
N_{DSS}	20	18	17	18	18
$N_{I(S)}$	68	72	76	84	76
N_{MLP}	26	30	34	32	32
E_{DSS}	0.744	2.112	1.442	2.301	0.877
E_{MLP}	6.564	6.729	5.932	7.798	9.216
θ_{DSS}	14.89	38.02	24.52	41.41	15.79
θ_{MLP}	170.67	201.86	201.67	249.55	294.92
T_{DSS}	0.313	0.345	0.350	0.383	0.360
T_{MLP}	1.088	1.090	1.106	1.122	1.138

Table 3. Sinc function.

Table 3 illustrates by a few numerical values that measurements such as errors will not decrease monotonously with an increase in resolution. This is also illustrated in Fig. 4 for the resulting errors in case of the circle, and in Fig 5 for the resulting effectiveness values of the approximation, also in case of the circle. This shows that tables such as Tables 1 and 2 showing a few isolated measurements only may be absolutely misleading! They may only provide a 'first idea' about the behavior of the algorithms. Generalizations require statistical evaluations. Due to the inhomogeneous in-

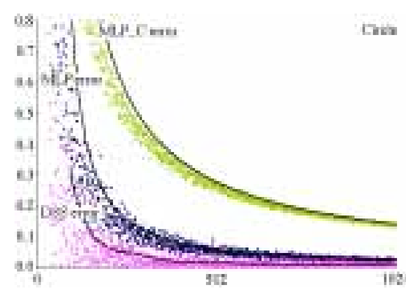

Figure 4. DSS and MLP errors (circle).

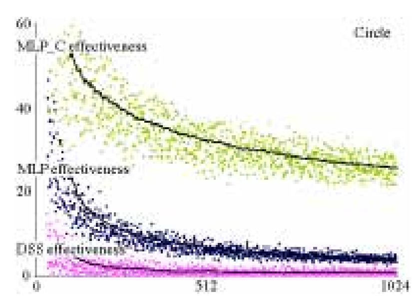

Figure 5. DSS and MLP trade-off (circle).

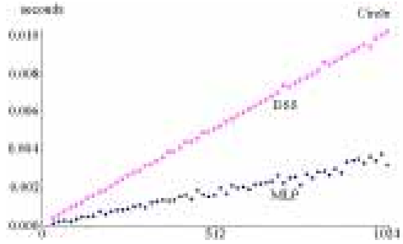

Figure 6. DSS and MLP time (circle).

crease in error and effectiveness values we use *sliding means* for graphical representations of results. These means are the arithmetic averages of 64 values, 32 values on the left of the recent position, the value at the recent position, and 31 values on the right of the recent position. In Fig. 4 and Fig. 5 these sliding means form curves.

Figures 4 and 5 also show results for the MLP_C estimates where the input polygons $I(S)$ have been replaced by the larger input polygons $C(S)$. This is just a minor increase of the input polygons but it illustrates the sensitivity of the MLP algorithm. Similar results have been obtained if the input $C(S)$ of the DSS algorithm is replaced by $I(S)$ or $O(S)$. The errors of the DSS_I estimate are slightly larger than the errors of the MLP_C estimate in case of the circle. However, for the halfmoon boundary the MLP_C estimates are better than the MLP estimates. The computing time (see Fig. 6) is more regular in its behavior. The computing time of the MLP_C algorithm is about the same as in the MLP case.

Figure 7 summarizes our experiments. It shows the averaged sliding means of error values E_{DSS} or E_{MLP}, and of sliding means of effectiveness values θ_{DSS} and θ_{MLP}, where each value at resolution r is generated by averaging the results for the six examples of sets at this resolution r. The tangential geometric convergence of the yin curve has a special impact on slowing down numerical convergence to the true value. For that reason curves only averaging five sets (without yin curve) are also included in Fig. 7.

4. Conclusions

Typically the DSS errors in estimating the true perimeter have been at 0.1% or lower for grid resolutions of 600 or higher, also for non-convex sets such as the halfmoon or the sinc-curve. The errors were slightly larger for the yin-part of the Yinyang symbol. The DSS algorithm shows in general faster convergence and a better efficiency with respect to our trade-off measure.

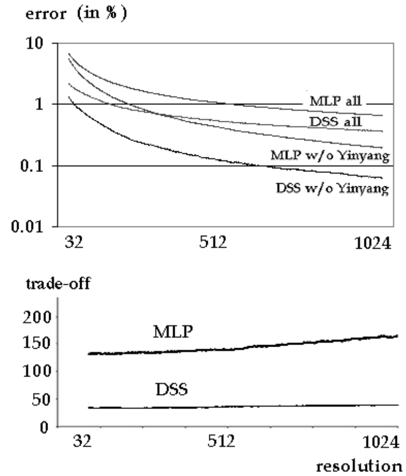

Figure 7. Errors and effectiveness.

In our implementation, the DSS algorithm needs about 2-3 times more computing time compared to the MLP algorithm, and this is about the same for the different test sets. The times for generating the input data, ie. frontiers of $C(S)$ or $I(S)$, are not counted in this time comparison. The four-vertex scheme might be a reason that MLP 'typically' approximates the curve length from below but DSS results 'typically' oscillate (i.e. there are both positive and negative differences) around the true value. – The authors acknowledge the collaboration with V. Kovalevsky (Berlin).

References

[1] R. Klette, V. Kovalevsky, and B. Yip. On the length estimation of digital curves. *Vision Geometry VIII* (Denver), SPIE Volume 3811, pages 52–63, 1999.

[2] R. Klette, J. Žunić. Multigrid convergence of calculated features in image analysis. CITR-TR 51, CITR Tamaki, Auckland University, October 1999.

[3] V. Kovalevsky. New definition and fast recognition of digital straight segments and arcs. Proc. *10th ICPR* (Atlantic City), IEEE Publ., pages 31–34, 1990.

[4] V. Kovalevsky and S. Fuchs. Theoretical and experimental analysis of the accuracy of perimeter estimates. *Robust Computer Vision* (W. Förstner, S. Ruwiedel, eds.), Wichmann, Karlsruhe, pages 218–242, 1992.

[5] W. Rinow. *Topologie.* Deutscher Verlag der Wissenschaften, Berlin, 1975.

[6] A. Rosenfeld. Digital straight line segments. *IEEE Trans. Comp.*, C-23:1264–1269, 1974.

[7] F. Sloboda, B. Zaťko, and J. Stoer. On approximation of planar one-dimensional continua. *Advances in Digital and Computational Geometry* (R. Klette, A. Rosenfeld and F. Sloboda, eds.), Springer, Singapore, pages 113–160. 1998.