Electronic Letters on Computer Vision and Image Analysis 5(2):98-110, 2005

Multiorder polygonal approximation of digital curves

I. Debled-Rennesson, S. Tabbone and L. Wendling
LORIA-INRIA
Campus Scientifique, BP 239
54506 Vandœuvre-les-Nancy Cedex
France
email: {debled,tabbone,wendling}@loria.fr

Received 16 July 2004; accepted 6 June 2005

Abstract

In this paper, we propose a quick threshold-free algorithm, which computes the angular shape of a 2D object from the points of its contour. For that, we have extended the method defined in [4, 5] to a multiorder analysis. It is based on the arithmetical definition of discrete lines [11] with variable thickness. We provide a framework to analyse a digital curve at different levels of thickness. The extremities of a segment provided at a high resolution are tracked at lower resolution in order to refine their location. The method is threshold-free and automatically provides a partitioning of a digital curve into its meaningful parts.
Key Words: Polygonal Approximation, Scale Space, Discrete Lines

1 Introduction

Generally, the goal of a polygonal approximation method is to provide a cut-out of a sequence of points into segments in order to minimise a global error criterion or to not locally overtake a predefined error. The interest is to give a description which is more compact and more adapted for further interpretation or shape recognition processing. The sequences of points are mainly obtained by an outline detection process in gray-scale images or by a skeleton in graphical documents.

Many polygonal approximation methods have been designed throughout the years [2, 13, 19]. The aim is to approximate a given digital curve by another polygonal curve with a number of line segments considering a local or global approximation error. Some approaches [9, 15, 16] use a linear scan of the digital curve. The aim is to find the longest segments such that the error with the digital curves is lower than a predefined tolerance error. Perez and Vidal [8] have proposed an interesting algorithm, based on a dynamic programming, to minimise a global error of segmentation. However the complexity of the approach is high because the whole space of solutions is checked. Recent works [7, 14] have been proposed to improve it.

Split-and-merge methods [10, 12] start from an initial segmentation and then iteratively split a line if the error is too big and otherwise merge two lines if the error is too small. In some methods [1, 12, 18] anchor points which correspond to high curvature points, are injected in the segmentation process.

Despite the important number of approximation methods, there are still major problems of robustness, stability and complexity when applying geometrical transformations. The algorithms rely on error tolerance thresholds which are manually defined without any knowledge on the most pertinent value of the threshold. Moreover they can be different from a sequence of points to another.

Correspondence to: debled@loria.fr

Recommended for acceptance by J.M. Ogier, T. Paquet, G. Sanchez
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

In this paper, we present a quick threshold-free algorithm, which computes the angular shape of a 2D object from the points of its contour. This rough representation shall be used, for example, in shape recognition system for a quick first classification of a database of objects. In this perspective, we use the linear method defined in [4, 5] which consists in the segmentation of a given curve into blurred segments. This algorithm depends on a parameter, called order, which allows to control the amplitude of the authorised noise by setting the thickness of the discrete line bounding the blurred segment.

The study of the behaviour of this algorithm lead us to use it on a multiorder analysis framework and we obtain a threshold-free method which permits to split a given curve into meaningful segments.

In the next section, we recall the theoretical concepts related to the blurred segment definition. Then, in section 3, we present the algorithm used to split a given curve into several blurred segments at a given order. At last, we present in sections 4 and 5 our approach and the threshold-free algorithm to analyse a sequence of points in a multiorder space as well as the obtained results.

2 Blurred segment

In the following, we refer to the first octant (8^{th}) of the plane such that $x \geq 0$ and $0 \leq y \leq x$. The notion of blurred (or fuzzy) segments relies on the ***arithmetical definition of discrete lines*** [11] where a line, whose slope is $\frac{a}{b}$, lower bound μ and thickness ω (with a, b, μ and ω being integer) is the set of integer points (x, y) verifying $\mu \leq ax - by < \mu + \omega$. Such a line is noted $\mathcal{D}(a, b, \mu, \omega)$. From now on, we shall name these segments blurred segments rather than fuzzy segments in order to prevent any confusion with fuzzy logic concepts.

The real lines $ax - by = \mu + \omega - 1$ and $ax - by = \mu$ are named the leaning lines of $\mathcal{D}(a, b, \mu, \omega)$. Moreover, the points (x_L, y_L) (resp. (x_U, y_U)) of a discrete line $\mathcal{D}(a, b, \mu, \omega)$ which verify $ax_L - by_L = \mu + \omega - 1$ (resp. $ax_U - by_U = \mu$) are called ***lower (resp. upper) leaning points*** and located under (resp. above) all other points of $\mathcal{D}$ (see dashed pixels in figure 1). We have the following definitions [4]:

Definition 1: A set $\mathcal{S}f$ of consecutive points ($|\mathcal{S}f| \geq 2$) of an 8-connected curve is a ***blurred segment with order d*** if there is a discrete line $\mathcal{D}(a, b, \mu, \omega)$ such that all points of $\mathcal{S}f$ belong to $\mathcal{D}$ and $\frac{\omega}{max(|a|,|b|)} \leq d$. The line $\mathcal{D}$ is said ***bounding*** for $\mathcal{S}f$.

The order of a blurred segment allows to limit the thickness of the discrete line bounding the 8-connected sequence of points of the blurred segment and, so doing, to control the length of vertical steps of the bounding line. In order to be reasonably close to the points of the blurred segment, we introduce more restrictive conditions to the discrete line with the notion of strictly bounding line as defined hereafter.

Definition 2: Let $\mathcal{S}f$ be a blurred segment with order d whose abscissa interval is $[0, l-1]$, and let $\mathcal{D}(a, b, \mu, \omega)$ be a bounding line of $\mathcal{S}f$. $\mathcal{D}$ is named ***strictly bounding for*** $\mathcal{S}f$ if, $\mathcal{D}$ possesses at least three leaning points in the interval $[0, l - 1]$ and, $\mathcal{S}f$ contains at least one lower leaning point and one upper leaning point of $\mathcal{D}$.

The following theorem, proven in [4], studies the different possible cases of the growth of a blurred segment.

Theorem 1: Let us consider a blurred segment $\mathcal{S}f$ in the first octant whose abscissa interval is $[0, l - 1]$ and $\mathcal{D}(a, b, \mu, \omega)$, a strictly bounding line. In this case, the order of $\mathcal{S}f$ is $\frac{\omega}{b}$. Let $M(x_M, y_M)$ be an integer point connected to $\mathcal{S}f$ whose abscissa is equal to l or $l - 1$. We call ***remainder*** at M, as a function of $\mathcal{D}$, noted $r(M)$ and defined by $\mathbf{r(M) = ax_M - by_M}$.

(i) If $\mu \leq r(M) < \mu + \omega$, then $M \in \mathcal{D}$;
$\mathcal{S}f \cup M$ is a blurred segment whose order is $\frac{\omega}{b}$ with $\mathcal{D}$ as strictly bounding line.

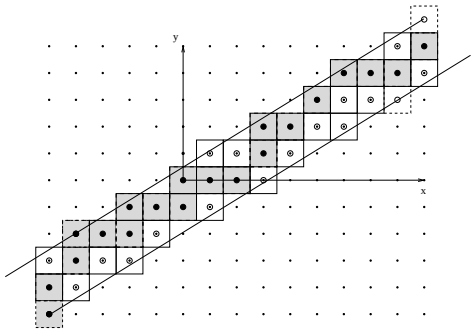

Figure 1: Points of $\mathcal{D}(8, 13, -6, 32)$ in the abscissa interval $[-5, 9]$, in grey a blurred segment with order 3 and $\mathcal{D}$ as strictly bounding line.

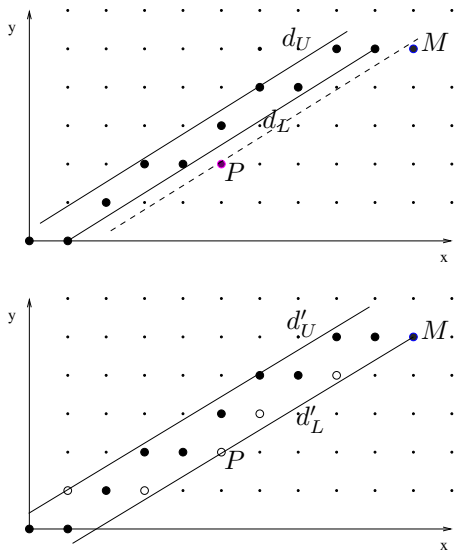

Figure 2: An example of blurred segment growth relying on the theorem 1.

(ii) If $r(M) \leq \mu - 1$, then M is exterior to $\mathcal{D}$;
$\mathcal{S}f \cup M$ is a blurred segment whose order is $\frac{\omega'}{b'}$ and the line $\mathcal{D}'(a', b', \mu', \omega')$ is strictly bounding, with

- b' and a' coordinates of the vector $\overrightarrow{P_{r(M)+1}M}$, $P_{r(M)+1}$ being the point whose remainder is $r(M)+1$ with regard to $\mathcal{D}$ and $x_{P_{r(M)+1}} \in [0, b-1]$,
- $\mu' = a'x_M - b'y_M$
- $\omega' = a'x_{L_L} - b'y_{L_L} - \mu' + 1$, with $L_L(x_{L_L}, y_{L_L})$ last lower leaning point of the line $\mathcal{D}$ present in $\mathcal{S}f$.

(iii) If $r(M) \geq \mu + \omega$, then M is exterior to $\mathcal{D}$;
$\mathcal{S}f \cup \{M\}$ is a blurred segment whose order is $\frac{\omega'}{b'}$ and the line $\mathcal{D}'(a', b', \mu', \omega')$ is strictly bounding with

- b' and a' coordinates of the vector $\overrightarrow{P_{r(M)-1}M}$, $P_{r(M)-1}$ being the point whose remainder is $r(M)-1$ with regard to $\mathcal{D}$ and $x_{P_{r(M)-1}} \in [0, b-1]$,
- $\mu' = a'x_{U_L} - b'y_{U_L}$ with $U_L(x_{U_L}, y_{U_L})$ last upper leaning point of the line $\mathcal{D}$ present in $\mathcal{S}f$,
- $\omega' = a'x_M - b'y_M - \mu' + 1$.

An example of application of this theorem is given in the figure 2. On the top of it, a blurred segment $\mathcal{S}f$ of order 1, $\mathcal{D}(5, 8, -2, 8)$ is strictly bounding for $\mathcal{S}f$, d_U and d_L are the leaning lines of $\mathcal{D}$. The point $M(10, 5)$ is added to $\mathcal{S}f$, as $r_{\mathcal{D}}(M) = 10$, adding M to $\mathcal{S}f$ corresponds to the case (iii) of the theorem : P is the point in $[0, 7]$ such that $r_{\mathcal{D}}(P) = 9$, therefore, $\mathcal{S}f \cup \{M\}$ has $\mathcal{D}'(3, 5, -2, 7)$ as strictly bounding line whose slope is calculated with the vector PM. In the bottom of figure 2, a representation of $\mathcal{D}'$ and $\mathcal{S}f \cup \{M\}$ (black points) is given. The points of $\mathcal{D}'$ which do not belong to $\mathcal{S}f \cup \{M\}$ are in white, d'_U and d'_L are the leaning lines of $\mathcal{D}'$.

The algorithm presented in the following section is directly deduced from this theorem.

3 A segmentation algorithm of 8-connected curves into blurred segments

The curve $\mathcal{C}$ is incrementally scanned, each point is watched. Let $\mathcal{S}f$ be the current order d blurred segment, a point M of $\mathcal{C}$ is added to $\mathcal{S}f$, the characteristics of a strictly bounding line of $\mathcal{S}f \cup M$ are calculated (according to the theorem 1). The current segment includes the point M if the value of the obtained ratio $\frac{\omega}{max(|a|,|b|)}$ is lesser than or equal to the order d. In the algorithm given hereafter, two procedures are used:

- Each point M of E is analysed, transformed in the first octant and added to the current segment by the procedure addPointSf which possibly changes the characteristics a, b, μ and ω of a strictly bounding line of this segment according to the theorem 1.
- The testOctant procedure tests the validity of the point M according to the octant of the current segment, and sets the boolean $isSameOctant$ to the right value, possibly updates the number of the octant of the current segment. The way the boolean value $isSameOctant$ is updated in the procedure testOctant depends on the authorised directions (freeman code elements) of a segment in a given octant.

Algorithm 1: Blurred Segmentation

Input: $\mathcal{C}$ an 8-connected sequence of points and d the authorised order
for the blurred segments.
Output: the list L of blurred segments, each of them being defined by its
number of points $nbPoint$ and the characteristics a, b, μ, ω
of a strictly bounding line.
Initialisation: $a = 0$, $b = 1$, $\mu = 0$, $\omega = b$, $nbPoint = 1$, $M_c = (0, 0)$,
$isSegment = true$, $end = false$, $isSameOctant = true$,
M= the first point of $\mathcal{C}$.

while !end **do**
  while $isSegment$ and $isSameOctant$ and !end **do**
    $M_{last} = M$;
    M = next point of $\mathcal{C}$;
    testOctant(M) ; M_c = image of M in the first octant ;
    $a_{last} = a$; $b_{last} = b$; $\mu_{last} = \mu$; $\omega_{last} = \omega$;
    if $isSameOctant$ **then**
      addPointSf(a,b,μ,ω, M_c) ;
      $isSegment$ = $\frac{\omega}{b} \leq d$;
      if $isSegment$ **then** $nbPoint$ ++ ; **endif**
    endif
    end = $\mathcal{C}$ is entirely scanned ;
  endwhile
  if $isSegment$ and $isSameOctant$ **then**

Add to L the blurred segment characterised by $nbPoint$ and, according to the current octant, the transformed characteristics of a, b, μ, ω ;
else
Add to L the blurred segment characterised by $nbPoint$ and, according to the current octant, the transformed characteristics of a_{last}, b_{last}, μ_{last}, ω_{last} ;
endif
$a = 0$; $b = 1$; $\mu = 0$; $\omega = b$; $nbPoint = 1$; $M = M_{last}$; $M_c = (0, 0)$;
$isSegment = true$; $isSameOctant = true$;
endwhile

This algorithm is very fast; each point M of $\mathcal{C}$ is analysed only once and added to the current segment by the procedure addPointSf. Furthermore the operations which are necessary to make the characteristics evolve are not costly (for details see [5]). The order of a blurred segment, allows to control the amplitude of the authorised noise by fixing the thickness of the discrete line bounding the segment.

Study of the behaviour of the Algorithm 1. At a low order, the extremities of the obtained segments are well localised. However, for noisy sets of points, too many small inconsistent segments are also provided. When the order increases, we can see shifts at the extremities points between the obtained segments and the expected ones (c.f. the segmentation at order 7 in the figure 3). Nevertheless the number of segments decreases and only significative segments remain *. For example, in the figure 3, we can see that the number of segments for an order 1 is 15 while it decreases to 3 for an order 7.

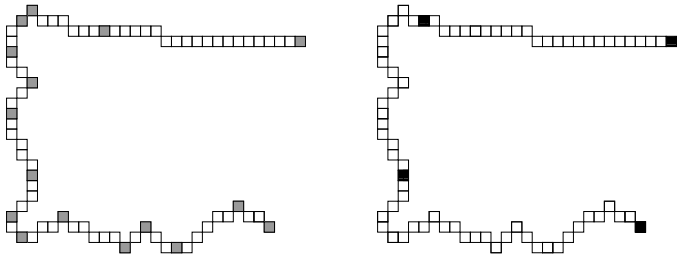

Figure 3: Examples of segmentation at two orders. Grey points (left): extremities points of obtained order 1 blurred segments. Black points (right): extremities points of obtained order 7 blurred segments.

From this statement, we present in the next section a multiorder approach which uses the algorithm 1 and the previous remarks.

4 Multiorder analysis

We propose to analyse a set of points in a multiorder space. Such a space looks like the scale space representation which was first introduced by A. Witkin [17] for the detection of edges at different scales.

Figure 4 shows a multiorder space determined from the segments obtained with algorithm 1 at different orders using the set of points of the figure 3. For each order d, represented on the Z-axis, we put the coordinates of the extremities points of the obtained blurred segments. We can see in this example that 4 extremities remain. They correspond to the description in three segments of the set of points (see figure 5). Moreover the position

*Significative qualifies segments which correspond to a rough polygonal contour defined from the main segments of the set of points.

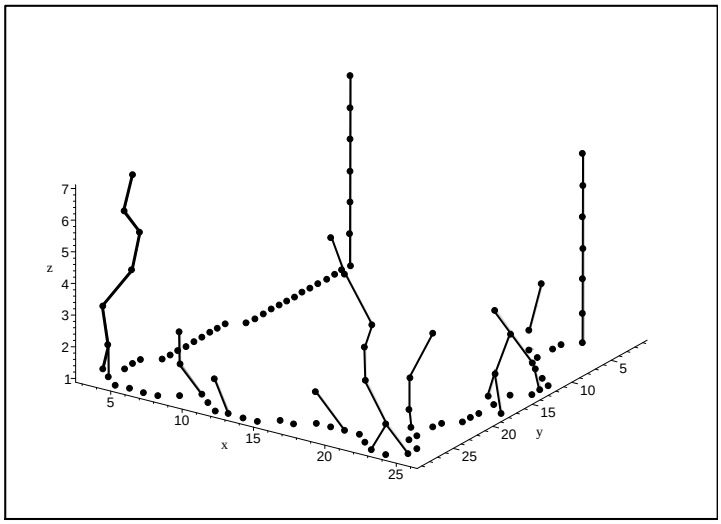

Figure 4: Example of a multiorder space.

of the points evolves except for the extremities of the set. When the number of segments from one order to another does no longer change, we decide to stop the description process. From that order, the extremities of the points are projected onto a lower order and the new position of each point is defined from the nearest segment extremity provided by the lower order. The size of the observation neighbourhood is equal to the thickness of the segment provided by the currently analysed level. This thickness is deduced from the order (cf. §3).

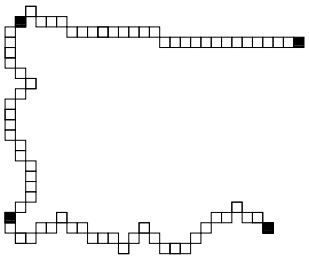

Figure 5: Multiorder Segmentation. Black points: multiorder points obtained with algorithm 2.

We present below the general algorithm which permits to extract the multiorder points. $Approx$ is a function, coming from the algorithm of §3, enabling to segment a sequence $\mathcal{C}$ of order d and returning the number of found points, denoted $nbpoint$, as well as a set of points named $L[d][nbpoint]$. The search of neighbourhood for a given order is determined by calculating a « chess board »-type distance [3], denoted $Dist$ with $Dist(a, b) = max(|x_a - x_b|, |y_a - y_b|)$. This metric assumes that you can make moves on the pixel grid as if you were a King making moves in chess, i.e. a diagonal move counts the same as a horizontal move. This distance is much faster to compute than the Euclidean metric.

Algorithm 2: Multiorder Segmentation

Input: $\mathcal{C}$ an 8-connected sequence
Output: the list L_{MO} of multiorder points
Initialisation : $d = 0$, $nbpoint = 0$

do $d = d + 1$;
$nbprec = nbpoint$;

$nbpoint =$ Approx($\mathcal{C}$, d, $L[d]$);
while $nbpoint \neq nbprec$
$L_{MO} \Leftarrow L[d]$
for $d_c = d - 1$ **downto** 1
$\forall u \in L_{MO}$
% One scan is enough to consider any point
if $\exists p \in L[d_c]/Dist(p,u) < Int((d_c+1)/2) + 1$ **then**
% shifting in multi-order space (p becomes u)
$L_{MO} \Leftarrow L_{MO} - \{u\}$; $L_{MO} \Leftarrow L_{MO} \cup \{p\}$;
else
$L_{MO} \Leftarrow L_{MO} - \{u\}$; % the multiorder path is not right
endif
endfor

5 Experimental results

Figures 6, 7, 8 and the appendix section show some examples of polygonal approximation obtained using our approach.

By comparison, we print on the figures 6 and 7 the results found by applying the Wall and Danielsson's method [16]. We adjust at best their cut threshold so that the number of segments is similar to the one of our approach. Moreover we show in the figure 7, the results of the polygonal approximation obtained for each image at the highest order. The stop orders are respectively 11, 4 and 8. We can remark that the number of iterations to obtain the convergence is linked to the regularity of the shape. Moreover, in our approach, the number of segments is automatically obtained while we have to test several different thresholds in the other method to vectorize the image with the same number of segments.

In some cases, some improvements of the algorithm shall have to be done regarding the tracing of points in the multiorder space. For example, on the figure 7.h some points provided by the last order are merged by moving to a lower order. Another problem concerns the tracing of points for which two choices are possible (points at an equal distance) at lower orders (cf. multiorder space of the figure 4). Currently a move by default is chosen. At last the first point of the sequence is considered at all orders even if it does not correspond to the extremity of an expected segment. We foresee to make this cut-out from the estimation of the tangent at some points [6].

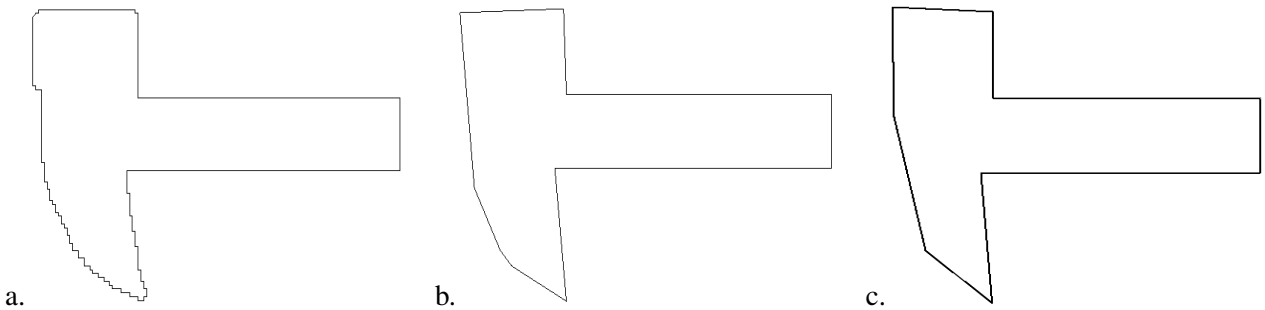

Figure 6: Experimental results.
a) Curve representing a hammer. b) Results automatically obtained with our approach. c) Results obtained with the Wall and Danielsson's method [16] with a cut threshold of 30.

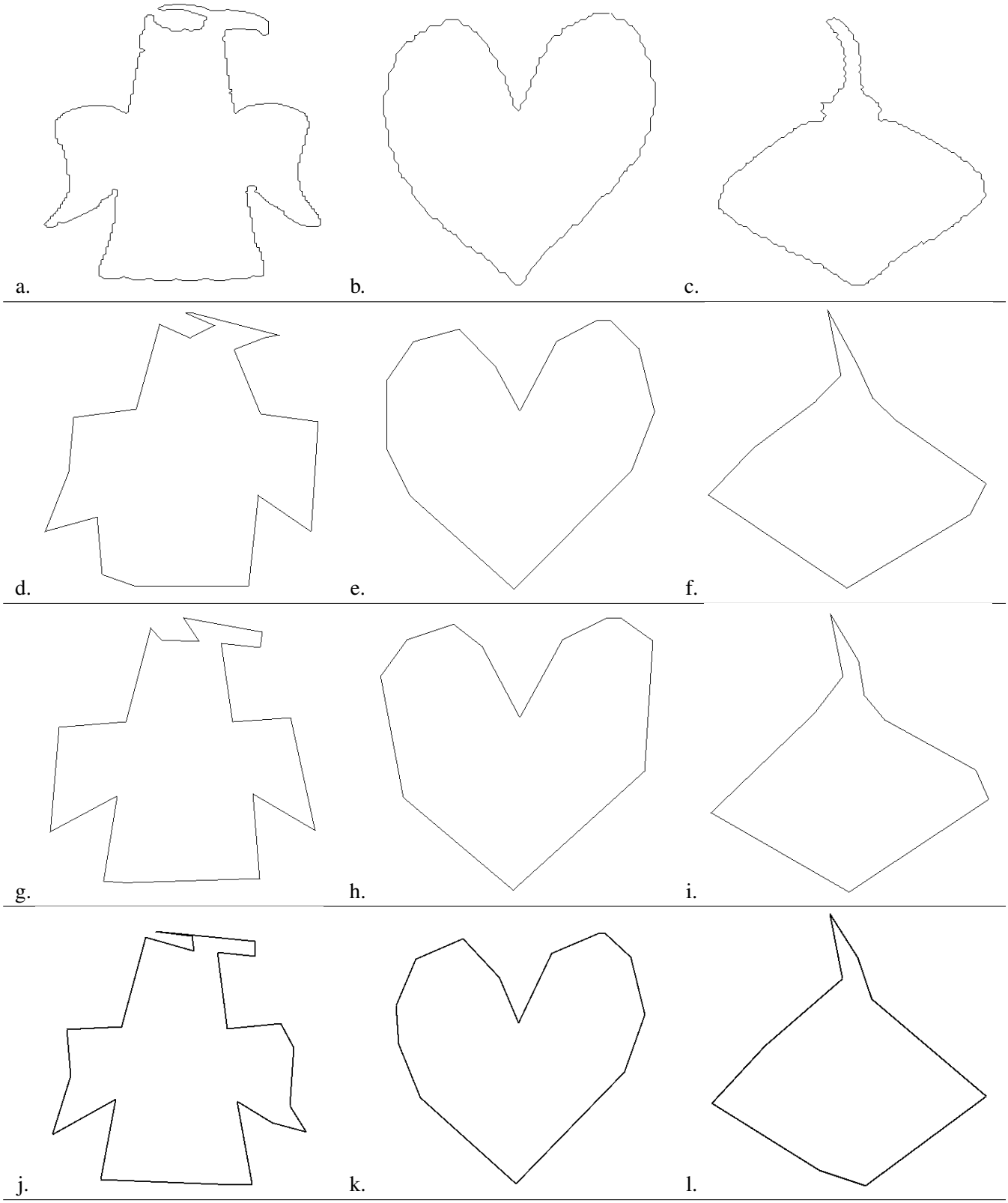

Figure 7: Other experimental results.
a)-c) Initial images. d)-f) Results of the polygonal approximation obtained at high order with algorithm 1. g)-i) Final results obtained with the multiorder analysis (algorithm 2). j)-l) Results obtained with the Wall and Danielsson's algorithm [16] from the QGAR platform (http://www.qgar.org/)

Moreover, an error criterion [8, 14] has been implemented to assess the gain of accuracy due to the multiorder approach. The error $e_{i,j}$ is defined from a curve in the range $[c_i; c_j]$ and its orthogonal projection onto the polar line $x\cos(\theta) + y\sin(\theta) = \rho$ as follows: Let $C = \{c_1, \ldots, c_N\} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a curve defined by N sorted points. Let $Q = \{q_1, \ldots, q_{M+1}\}$ be the polygon computed from C using the multi-order algorithm. Each point of Q belongs to C with $q_1 = c_1$ and $q_{M+1} = c_N$. Let us now consider a segment (q_m, q_{m+1}) of Q and the associated series of points $\{c_i, \ldots, c_j\}$ in C. The error between them is directly given by:

$$e^2(p_i, p_j) = \sum_{k=i+1}^{j-1} (\sin(\theta_{ij})y_k + \cos(\theta_{ij})x_k - \rho_{ij})^2$$

$\cos(\theta_{ij})$ and $\sin(\theta_{ij})$ are defined from the polar representation $x\cos(\theta) + y\sin(\theta) = \rho$ of the straight line (c_i, c_j). Considering all the segments of Q, the global error E is calculated as follows:

$$E = \sum_{m=1}^{M} e^2(q_m, q_{m+1})$$

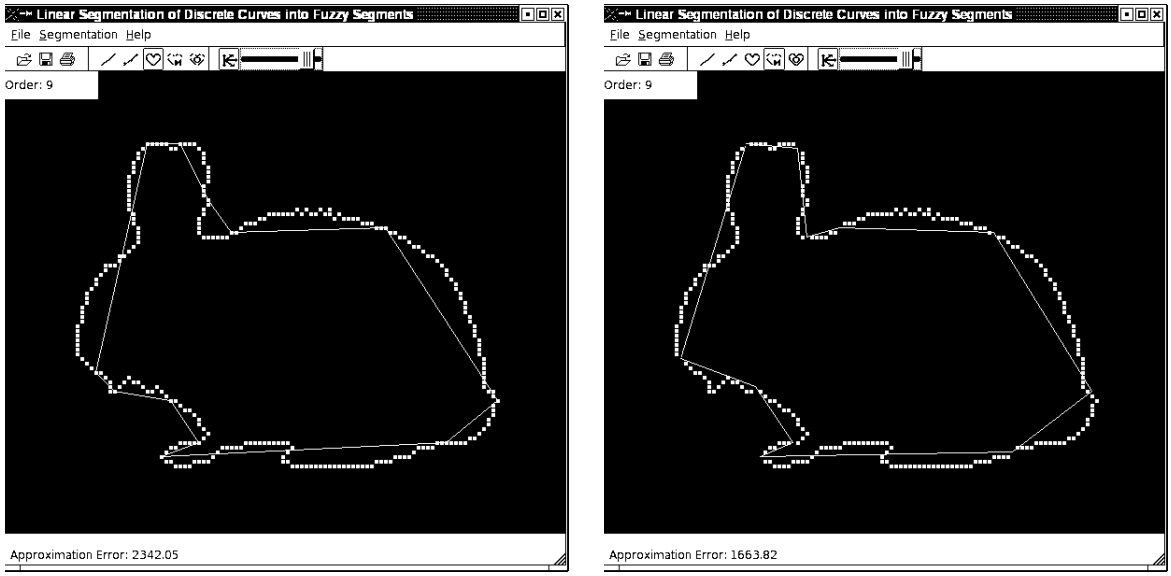

Figure 8: Errors (see at the bottom of each window) reached using blurred segmentation (algorithm 1, on the left hand side) and multiorder segmentation approaches (algorithm 2, on the right hand side).

In most cases, the error reached is lower by using the multiorder segmentation (algorithm 2) than by using the basic blurred segmentation (algorithm 1) for the same order (see figure 8). This refinement of the location of the multiorder points permits to have a better representation of the global shape of the object. Applications on several shapes are provided into the appendix section.

Figure	8	9	10	11	12
(a) Blurred segmentation	2342.05	1558.66	6696.26	1400.55	24840.6
(b) Multiorder segmentation	1663.82	795.007	5810.34	1343.04	10327.7
Ratio	0.71	0.51	0.87	0.96	0,42

Table 1: Error variations in blurred and multiorder segmentations.

Table 1 provides approximation errors reached using both methods applied on figures 8 to 12. We also give a ratio (b/a) to see the improvement due to the multiorder algorithm. A low ratio means that the multiorder segmentation improves the result obtained using the first method.

The more the shape has acute angular points the lower is the error ratio. In this case, the multiorder method is more suitable. When shapes are relatively smoothed, both approaches give similar results because the problem of wrong location of extremities points by the first method is not improved by the second method as shown in figure 10 and 11.

6 Conclusion

We have presented a multiorder analysis algorithm which provides a fast polygonal approximation of a digital curve from a scale space study. Unlike the other methods where an error of approximation is predefined and set through a trial and error process to achieve the better compromise between the number of segments and their location, our method is threshold-free and automatically provides a partitioning of a digital curve into its meaningful parts. We are aware that in some cases the stability criterion based on the number of segments is not optimal (local minima problem). Further work will be devoted to find better conditions to stop to analyse a digital curve at different levels of thickness. Moreover we wish to extend this approach to a rough classification of shapes. Nonetheless the approach is not invariant to rotation since the approximation method always begins with the same point. Such property should be checked before embedding the method in a pattern recognition process.

Acknowledgements

The authors thank T. Nguyen, internship student at Loria during summer 2004, he tested and implemented all algorithms in C++ with Qt library.

References

[1] Ansari N., Delp E. J., "On Detecting Dominant Points", *Pattern Recognition, vol. 24, 5, pp 441-451, 1991.*

[2] Davis T., "Fast Decomposition of Digital Curves into Polygons Using the Haar Transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, 8, pp 786-790, 1999.*

[3] P. E. Danielsson, "Euclidean distance mapping", *Computer Vision, Graphics, and Image Processing, 14, pp. 227- 248, 1980.*

[4] Debled-Rennesson I., Rémy J.-L., Rouyer-Degli J., "Segmentation of Discrete Curves into Fuzzy Segments", *9th International Workshop on Combinatorial Image Analysis, Electronic Notes in Discrete Mathematics, vol. 12, may 2003.*

[5] Debled-Rennesson I., Rémy J.-L., Rouyer-Degli J., "Segmentation of Discrete Curves into Fuzzy Segments, extented version", *INRIA Report RR-4989 (http://www.inria.fr/rrrt/rr-4989.html), november 2003.*

[6] Debled-Rennesson I., "Estimation of Tangents to a Noisy Discrete Curve", *Vision Geometry XII, Electronic Imaging, San José, Jan 2004.*

[7] Kolesnikov A., Fränti P., "Reduced-search dynamic programming for approximation of polygonal curves", *Pattern Recognition Letters, vol. 24, 14, pp 2243-2254, 2003.*

[8] Perez J., Vidal E., "Optimum polygonal approximation of digitized curves", *Pattern Recognition Letters, vol. 15,pp 743-750, 1994.*

[9] Ramer U., "An Iterative Procedure for the Polygonal Approximation of Plane Curves", *Computer Graphics and Image Processing, vol. 1, pp 244-256, 1972.*

[10] Ray B., Ray K., "A new split-and-merge technique for polygonal approximation of chain coded curves", *Pattern Recognition Letters, vol. 16, pp 161-169, 1995.*

[11] Reveillès J.-P., "Géométrie discrète, calculs en nombre entiers et algorithmique", *Thèse d'état. Université Louis Pasteur, Strasbourg, 1991.*

[12] Rosin P. L., West G. A., "Segmentation of Edges into Lines and Arcs", *Image and Vision Computing, vol. 7, 2, 1989, pp 109-114, 1989.*

[13] Rosin P. L., "Techniques for Assessing Polygonal Approximations of Curves", *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, 6, pp 659-666, 1997.*

[14] Salotti M., "An efficient algorithm for the optimal polygonal approximation of digitized curves", *Pattern Recognition Letters, vol. 22, pp 215-221, 2001.*

[15] Sklansky J., Gonzalez V., "Fast Polygonal Approximation of Digitized Curves", *Pattern Recognition, vol. 12, 1980, pp 327-331, 1980.*

[16] Wall K., Danielsson P., " A Fast Sequential Method for Polygonal Approximation of Digitized Curves", *Computer Vision, Graphics and Image Processing, vol. 28, 1984, pp 220-227, 1984.*

[17] Witkin A., "Scale space filtering", *Proc. International Joint Conference on Artificial Intelligence, 1983.*

[18] Wu W., Wang M., "Detecting the Dominant Points by the Curvature-Based Polygonal Approximation", *Graphical Models and Image Processing, vol. 55, 2, pp 79-88, 1993.*

[19] Yin P., "A Tabu Search Approach to Polygonal Approximation of Digital Curves", *International Journal of Pattern Recognition and Artificial Intelligence, vol. 14, 2, pp 243-255, 2000.*

Appendix

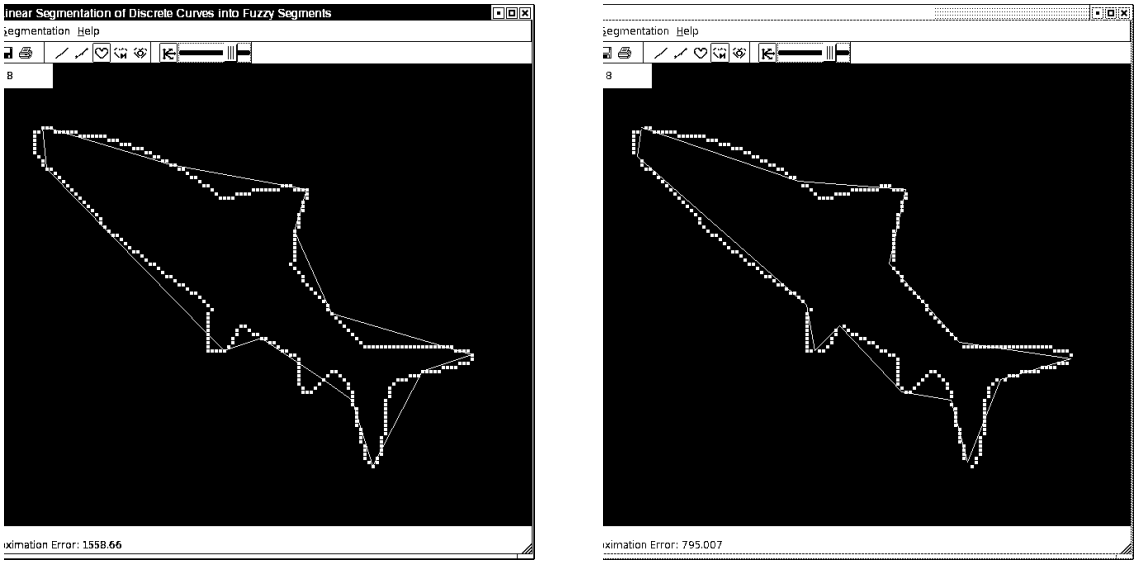

Figure 9: Errors (see at the bottom of windows) reached using blurred segmentation (algorithm 1) and multi-order segmentation approaches (algorithm 2).

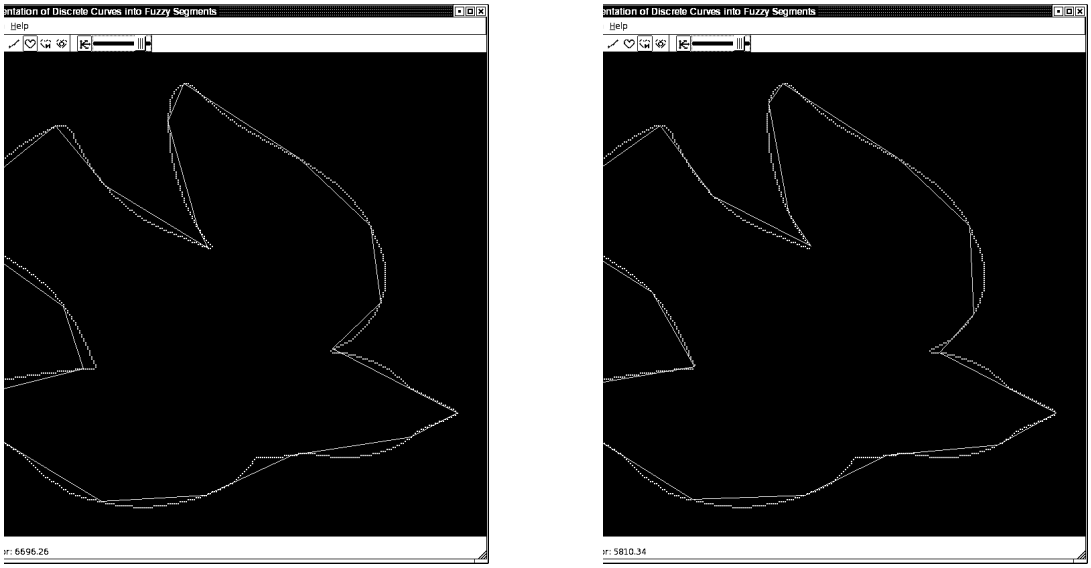

Figure 10: Errors (see at the bottom of windows) reached using blurred segmentation (algorithm 1) and multi-order segmentation approaches (algorithm 2).

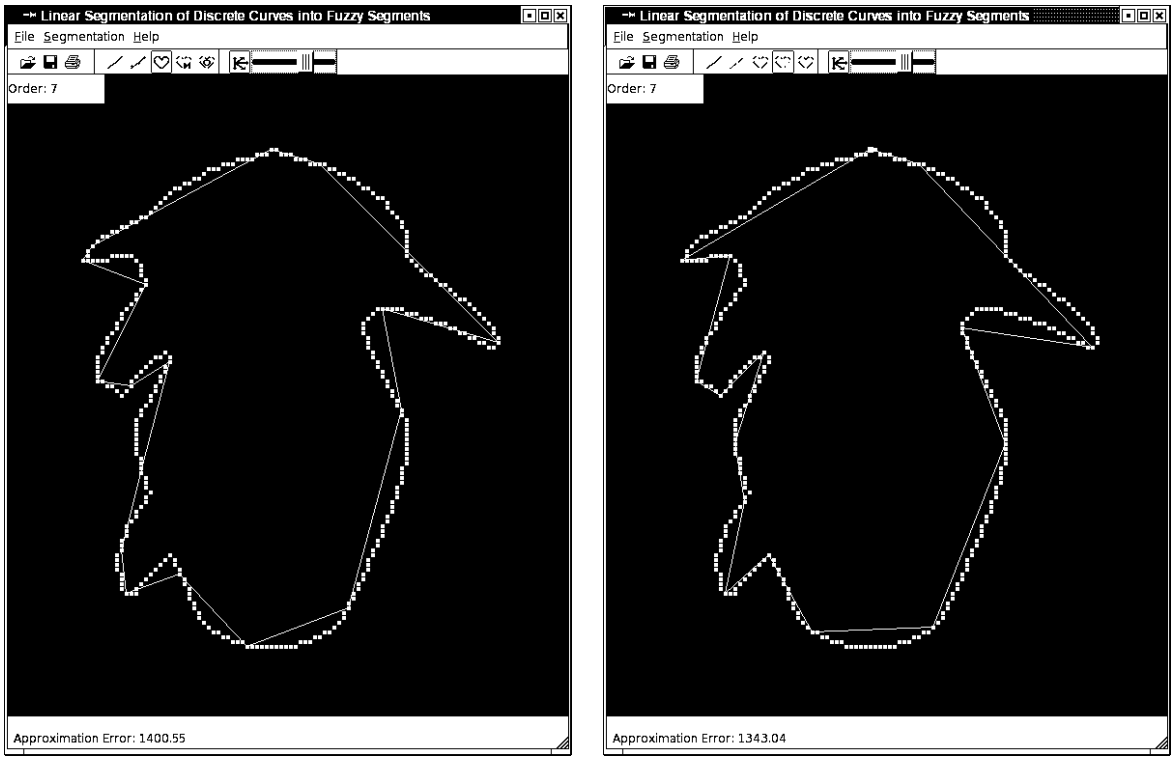

Figure 11: Errors (see at the bottom of windows) reached using blurred segmentation (algorithm 1) and multi-order segmentation approaches (algorithm 2).

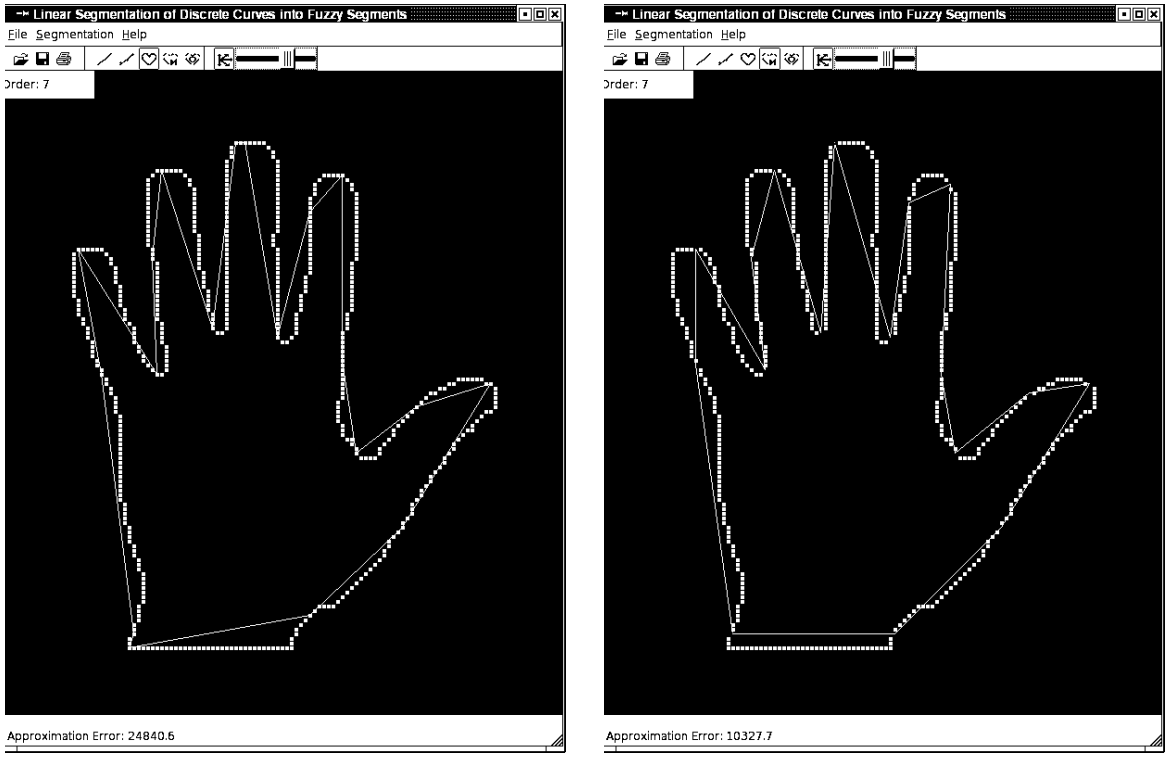

Figure 12: Errors (see at the bottom of windows) reached using blurred segmentation (algorithm 1) and multi-order segmentation approaches (algorithm 2).