



HAL
open science

Circular arc reconstruction of digital contours with chosen Hausdorff error

Bertrand Kerautret, Jacques-Olivier Lachaud, Thanh Phuong Nguyen

► **To cite this version:**

Bertrand Kerautret, Jacques-Olivier Lachaud, Thanh Phuong Nguyen. Circular arc reconstruction of digital contours with chosen Hausdorff error. *Discrete Geometry for Computer Imagery*, Apr 2011, Nancy, France. pp.250–262. hal-00579467

HAL Id: hal-00579467

<https://hal.science/hal-00579467>

Submitted on 23 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Circular arc reconstruction of digital contours with chosen Hausdorff error

Bertrand Kerautret^{1,2}, Jacques-Olivier Lachaud², and Thanh Phuong Nguyen¹

¹ LORIA (UMR CNRS 7503) Nancy University, France
{kerautret,nguyentp}@loria.fr

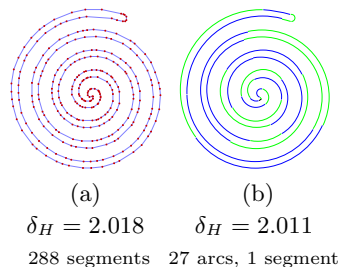
² LAMA (UMR CNRS 5127), University of Savoie, France
jacques-olivier.lachaud@univ-savoie.fr

Abstract. We address the problem of constructing an approximate continuous representation of a digital contour with guarantees on the Hausdorff error between the digital shape and its reconstruction. Instead of polygonalizing the contour, we propose to reconstruct the shape with circular arcs. To do so, we exploit the recent curvature estimators. From their curvature field, we introduce a new simple and efficient algorithm to approximate a digital shape with as few arcs as possible at a given scale, specified by a maximal admissible Hausdorff distance. We show the potential of our reconstruction method with numerous experiments and we also compare our results with some recent promising approaches. Last, all these algorithms are available online for comparisons on arbitrary shapes.

1 Introduction

Digital curve representation or approximation by simple primitives is useful for further shape analysis (recognition, matching, etc). Many methods approximate digital curves by a polygonal contour, using corner points or multi-scale analysis [6]. We propose here to work with higher order primitives like arcs of circle. This representation is more efficient for curved shapes like the one illustrated on the following floating figure and captures better its geometry.

In a previous work [11] we have explored the potential of different curvature estimators for corner detection. Together with a scale parameter, the obtained polygonal reconstructions were able to represent the contour by adapting the number of points according to the local value of curvature (see for example the polygon (a) of the upper floating figure). However as shown in figure (b), the circle arc primitive is much more efficient to represent numerous shapes. The previous example shows how arcs based representation gives a more compact description for the same accuracy (same Hausdorff error δ_H , more faithful normals). In this paper we investigate curvature based arc reconstruction by



introducing a simple reconstruction algorithm which exploits different recent curvature estimators.

Among previous works about reconstruction of digital contours with higher order primitives, we can mention the work of Rosin *et al.* [16], who constructed firstly a polygonal description and detected fitting arcs by grouping connected lines. Horng *et al.* [8] and Tortorella [18] introduced curve-fitting methods with an approach based on dynamic programming. Bodansky [4] presented a method for the approximation of a polyline with straight segments, circular arcs and free curves. It contains two steps. The first step is the segmentation of polygonal lines into fragments (short polygonal lines) and the second step is the approximation of the fragments by geometric primitives. If some fragments can not be approximated by geometric primitives with acceptable precision, they are recognized as free curves.

It is obvious that curvature information is meaningful for curve reconstruction by circle arcs and segments. A circle arc corresponds to a constant part in the curvature profile of the studied curve. Some methods have exploited this measure for curve reconstruction. Chen *et al.* [2] proposed a method for segmenting a digital curve into lines and arcs from curvature profile in which the number of primitives is given. This procedure contains two stages. The first stage computes a starting set of break points and determines an initial approximation by arcs and lines based on this set. This stage relies on the detection of significant changes along the curvature profile. The second stage is an optimization phase which adjusts the break points until the fitting error is locally minimized. Afterwards, Horng [7] proposed an adaptive smoothing approach for decomposition of a digital curve into arcs and lines. The input curve is segmented into arcs and lines according to the smoothed curvature representation. The curvature profile is determined by Gaussian filtering. Then, it is smoothed with an adaptive smoothing technique. Similarly, Salmon *et al.* [17] presented a method for decomposing a curve into arcs and segments according to the curvature profile too. They used a notion of discrete curvature which is related to the circumscribed circle induced by blurred segments. Their main idea is to extract key points on the curvature profile, which are then used for reconstruction. However the instability of this curvature estimator induces some complex curvature processing which adds new parameters and reduce the applicability of the approach.

We propose here to examine the potentiality of three recent curvature estimators in the context of reconstruction with circle arcs. Their main properties (stability wrt noise especially) are briefly described in Section 2. The reconstruction algorithm is presented in Section 3, and uses indifferently two of these estimators. We finally validate our reconstruction technique with several experimentations in Section 4. The visual curvature reconstruction [13] and the method of [15] act as reference reconstruction methods.

2 Recent Curvature Estimators

An overview of different curvature estimators are given in the following and more details can be found in their respective reference (see also the comparative evaluation in [11]).

Global Minimization Curvature (GMC). Curvature estimation is very problematic since infinitely many shapes have the same digitization. The idea of the GMC estimator [9] is to determine, among all possible shapes that have the same digitization as the digital object under study, the shape which minimizes its squared curvature. This shape is the most probable or expected shape if only a digital object is given, since it is the smoothest possible. Then the curvature field estimation is simply the curvature field of the minimal shape.

It is not a trivial task to determine this minimal shape. A phase-field approach is proposed in [1]. In [9], the proposed approximate optimization method provides a piecewise constant curvature field, so as the reconstructed shape boundary is made of circular arcs with tangent continuity. Furthermore it takes into account possible noise in the input data, with the use of blurred segments as a preprocess [3]. This estimator determines a curvature field with the smallest possible number of inflexion points. It is also almost rotation invariant due to processing with maximal digital straight segments. Its stability makes it particularly suitable to our reconstruction algorithm.

Binomial Convolution Curvature (BCC). This estimator was proposed by Malgouyres *et al.* [5, 14] as a discrete alternative to the Gaussian smoothing technique for estimating the curvature of a digital contour. Differential operators of order n are obtained as successive convolutions of m (say) binomial kernels and n difference kernels. The authors have shown that this differential estimator is multigrid convergent for well-chosen m (m depends on the sampling rate and other parameters like maximal curvature), even in the presence of noise. The main problem with this method is how to choose this m for a given input shape, since there is an *ad hoc* balance to be made between accuracy and smoothness, and some parameters are tricky to estimate. This method is also computationally costly for large m . We will nevertheless use it to show that our reconstruction method is relatively independent of the curvature field estimation.

Visual Curvature (VC). The visual curvature has been introduced by Liu, Latecki and Liu [13]. Its principle is to measure the number of extreme points of the height function along several directions, within a given window around each point. It is thus clear that points on the shape boundary that are also vertex of its convex hull are always extreme points. Furthermore, they introduced a scale parameter which keeps only extreme points surrounded by big enough concave or convex parts. This process filters non-significant features at a given scale. Keeping only the vertices with a non-zero multiscale visual curvature defines a simplified polygon. The sequence of polygons obtained by increasing the scale parameter from 0 to 1 creates a natural filtration of polygons, the simplest one being the convex hull. A drawback of the method is that the visual curvature

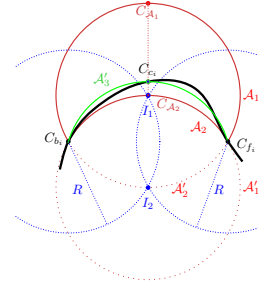
is only a qualitative estimation of the curvature in the general case. Furthermore, the simplified polygons are not controlled by an error measure. Lastly, this method requires four parameters. The visual curvature technique is recognized as a good feature detector and multiscale contour polygonalizer. Since the estimator is not suitable for our circle arc reconstruction (no distinction between concave/convex area), we will use it only for comparison purpose to assess the quality of our reconstruction method.

3 Contour Reconstruction with Circle Arcs

We propose a simple strategy to reconstruct a digital contour with the circle arc primitive. The main idea is to decompose the curvature estimation profile so as to find the significant curved areas which can be approximated by a circle arc. For this purpose a *split/merge* strategy is proposed. Split/merge is governed by a given maximal Hausdorff error E_{\max} with respect to the input digital contour. This parameter acts as a scale for the reconstruction. It also induces specific parametrizations of curvature estimators, which will be described in the multiscale reconstruction paragraph.

Algorithm 1 gives an overview of the main reconstruction process. First the curvature profile is decomposed into constant curvature parts. Then the set of local maxima/minima is extracted to define the initial regions as circle arcs. Since the circle arc estimation does not guarantee an error smaller than E_{\max} , a split process is first proposed to reduce the error between the circle arc and the curve until it becomes less than the maximal allowed value. The merge phase extends them with their neighborhood regions while the associated circle arc gives an error lower than E_{\max} . Before describing the error measure we focus on the problem of arc reconstruction from a contour region.

Arc reconstruction. Given a contour region R_i there are several ways to reconstruct a circle arc \mathcal{A}_i from two endpoints C_{b_i} and C_{f_i} . A first solution is to use the curvature information to determine the different possible solutions if there exist. Such a reconstruction is illustrated on the following figure by assuming a constant curvature value $\kappa_i = \frac{1}{R}$ estimated on the contour between C_{b_i} and C_{f_i} . The two possible centers of the osculating circles of radius R are represented with the intersections I_1 and



I_2 of the two dotted circles of center C_{b_i} and C_{f_i} . From these two points I_1 and I_2 , four circle arcs \mathcal{A}_1 , \mathcal{A}'_1 and \mathcal{A}_2 , \mathcal{A}'_2 are deduced as potential candidates for the reconstruction. Then the sign of the curvature retains only the circle arcs of same convexity. To select the final solution, we compute the distance between each middle arc point with the middle contour point C_{c_i} . In the previous illustrating example, the final arc will be \mathcal{A}_2 since the euclidean distance between C_{c_i} and $C_{\mathcal{A}_2}$ is less than the one between C_{c_i} and $C_{\mathcal{A}_1}$.

A second method of reconstruction simply assumes that the circle arc should interpolate the middle point C_{c_i} of the contour region. In this case the center of the circle arc can be determined if the center point is not collinear with the two other points C_{b_i} and C_{f_i} . By referring to the previous example the reconstructed circle arc is given by \mathcal{A}_3 . The two approaches have been experimented and the latter one gives the best results.

Error measure. Through the reconstruction process we need to evaluate the precision of the approximation by computing the error made between the reconstructed circle arcs (\mathcal{A}_i) of extremities (C_{b_i}, C_{f_i}) and the pieces of digital contour (C_i) defined between the two points. We propose to use the Hausdorff distance $\delta_H(\mathcal{A}_i, C_i)$ defined as:

$$\delta_H(\mathcal{A}_i, C_i) = \max\{\max_{b \in C_i} \{\min_{a \in \mathcal{A}_i} d(a, b)\}, \max_{a \in \mathcal{A}_i} \{\min_{b \in C_i} d(a, b)\}\}$$

The first term $\max_{b \in C_i} \{\min_{a \in \mathcal{A}_i} d(a, b)\}$ is computed in linear time by taking into account for each point C_j the angle $\theta_{\mathcal{A}_i}$ of the circle arc and the angle θ_{C_j} of the contour point C_j with the segment $O_i C_{f_i}$ where O_i is center of the circle arc \mathcal{A}_i . Two cases need to be considered (see Fig. 1(a)). If $\theta_{C_j} < \theta_{\mathcal{A}_i}$ then the minimal distance to the arc can be defined as the distance between C_j and the projection C'_j of C_j on the this circle. The minimal distance between C_j to any point of \mathcal{A}_i is thus given by $R - \|O_i C_j\|$ where R is the radius of the circle of center O_i . If θ_{C_j} is greater than the circle arc angle $\theta_{\mathcal{A}_i}$ then the projection C'_j is not located on the circle arc and the error is then defined by the minimal distance between $\|C_j C_{f_i}\|$ and $\|C_j C_{b_i}\|$.

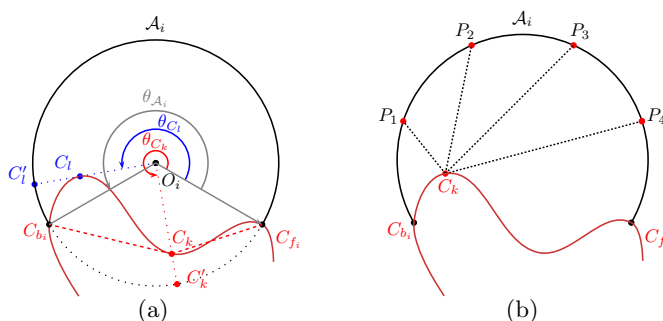


Fig. 1. Illustration of the evaluation of the Hausdorff distance between the circle arc and the digital contour.

The second term of the Hausdorff error measure implies a $O(N^2)$ complexity since for any point of the arc, the minimum distance with the N points of the contour part $C_{b_i} C_{f_i}$ is required. We therefore only approximate this measure, by computing the minimal distance only for some sampling points on the circle arc. They are defined so as to obtain at least one sampling point on each quadrant

(worst case for a circle arc with angle close to 2π). Thus four points are used to define the error (illustrated on Fig. 1(b)).

Merging process. The first merging process is initiated from the constant plot areas which are a local minima/maxima. The advantage of such strategy is to be independent of the choice of the initial contour point. Note that a precision parameter ϵ can be used to consider that two curvature values are equal. Then in order to avoid the presence of inconsistent circle arc, merging is allowed only between two contour regions with the same sign of curvature (definition included in the function `isExtendable`). It is also necessary to have a strategy for selecting in which order neighboring regions are merged with the considered central region. Here, the region whose mean curvature is closest is selected as first candidate to be merged. Such strategy is associated to the functions `selectFirstNearest`, `selectSecondNearest` and `extendFrontFirst` from Algorithm 3.

As described in Algorithm 1 the merging process is applied in a second phase in order to merge the potential arc regions located between two local maxima/minima. Finally, after this process, the function `selectMinErrPrimitive` is called to optionally verify if the straight segment could improve the reconstruction error. In order to favor circle arc reconstruction the straight segment is chosen only if it implies an error decrease at least equal to $E_{\max}/2$.

Algorithm 1: Reconstruction with arcs and segments

Data: $C = \{C_i\}_{i=0}^n$ digital curve, $\kappa = \{\kappa_i\}_{i=0}^n$ curvature estimation, float `maxArcError`;
Result: curve represented by a set of arcs and segments.
begin
 Decompose κ into a set of constant curvature interval S defined by:
 $\{(b_0, f_0), \dots, (b_i, f_i), \dots, (b_M, f_M)\}$.
 For each contour point C_i , store in `regionIndex[i]` the index $k \in \{0, \dots, M\}$ of its region $S[k]$.
 Extract from S the set S_m containing all the regions which are a local maxima/minima.
 $S_{tmp} = S$;
 while `nbElements(Stmp)! = 0` **do**
 | $S_{tmp} = \text{SPLIT_REGIONS}(S_{tmp}, S, \text{regionIndex}, \text{maxArcError})$;
 // First extension from mini/maxima regions:
 while `nbElements(Sm)! = 0` **do**
 | $S_m = \text{EXTEND_PLOT_REGIONS}(S_m, S, \text{regionIndex}, \kappa, \text{maxArcError})$;
 // Second extension from all others regions S_u :
 $S_u =$ set of index of valid non maxima/minima regions of the current regions.
 while `nbElements(Su)! = 0` **do**
 | $S_u = \text{EXTEND_PLOT_REGIONS}(S_u, S, \text{regionIndex}, \kappa, \text{maxArcError})$;
 // Verify or change primitive for region which are better approximate
 // by a straight segment:
 `checkBestPrimitive(S, tabRegionIndex, maxArcError)`;
end

Algorithm 2: SPLIT_REGIONS

Data: The set `setToCheck` of region index to be checked for splitting.
The set `S` of all curve `PlotRegions`.
The `rIndex` associating each point C_i to its region $S[k]$.
`maxArcError`: the maximal error allowed for splitting.
Result: A set containing all new index of splitted region.

```

for  $i = 0; i < \text{sizeOf}(\text{setToCheck}); i++$  do
    PlotRegion plotReg =  $S[i]$ ;
    float error = ComputeError(plotReg);
    if  $\text{error} > \text{maxArcError}$  then
        // Split the considered region
        PlotRegion newRegion1, newRegion2;
        newRegion1.b = plotReg.b;
        newRegion1.f =  $(\text{plotReg.b} + \text{size}(\text{plotReg})/2) \bmod (\text{rIndex.size}())$ ;
        newRegion2.b =  $(\text{plotReg.b} + \text{size}(\text{plotReg})/2 + 1) \bmod (\text{rIndex.size}())$ ;
        newRegion2.f = plotReg.f;
        add(S, newRegion1); add(S, newRegion2);
        add(splitSet, newRegion1); add(splitSet, newRegion2);
        InvalidAndUpdateReg( $S[i]$ , rIndex); updateRegs(newRegion1,
        newRegion2, rIndex);
return splitSet;

```

Multi-scale reconstruction. The reconstruction process is governed by a maximal error, which naturally induces a multi-scale reconstruction. To speed up the process and enhance the detection of significant points, the curvature estimator should also be tuned to take into account this error. This is easily done for the GMC curvature estimator, whose thickness parameter ν corresponds nicely with the Hausdorff error E_{\max} . Fig. 2(a) illustrates the blurred segment of width ν used in the GMC curvature estimator. When using the GMC estimator, the maximal error and the thickness ν are simply set to the chosen scale value. For the reconstruction with the BCC estimator its mask size has a linear dependence with the chosen scale value (although the initial mask size must be somehow determined by the user it was set as the contour size for the following experiment), while the maximal error is set equal to the chosen scale.

Time complexity. The global reconstruction complexity is first dependant of the curvature estimator. Since the error measure is computed in linear time, the complexity for the split/merging process is in the worst case equal to $O(n^2)$ when the contour (composed of n points) is reconstructed with a final arc obtained by adding one point at each step. On average the complexity is equal to $O(n \log(n))$.

Beside the difference in quality of the reconstructions, which is described in the next section, the choice of GMC or BCC estimator influences the efficiency of the merging process. Fig. 2(b) shows the number of region merges obtained for both estimators. The resulting plot shows that GMC is around 7 times more efficient than BCC for the merging process. The main reason of this difference

Algorithm 3: EXTEND_REGIONS

Data: The set $setToExtend$ of region index to be checked for merging.
The set S of all PlotRegions.
The $rIndex$ associating each point C_i to its region $S[k]$.
The curvature estimation $\kappa = \{\kappa_i\}_{i=0}^n$,
 $maxArcError$: the maximal error allowed after a merge.
Result: The set of region indexes which were updated: $resultSet$

```

for  $i = 0; i < nbElements(setToExtend); i++$  do
  PlotRegion  $reg = SetToExtend[i]$ ;
  PlotRegion  $regBack = getBackRegion(reg, S, rIndex)$ ;
  PlotRegion  $regFront = getFrontRegion(reg, S, rIndex)$ ;
  PlotRegion  $regFirst = selectFirstNearest(regBack, reg, regFront, \kappa)$ ;
  PlotRegion  $regSec = selectSecondNearest(regBack, reg, regFront, \kappa)$ ;
  bool  $frontFirst = extendFrontFirst(regBack, reg, regFront, \kappa)$ ;
  if  $isExtendable(reg, regFirst, \kappa)$  then
    PlotRegion  $tmp = MERGE(reg, regFirst, frontFirst)$ ;
    float  $error = COMPUTE\_ERROR(tmp)$ ;
    if  $error < maxArcError$  then
      add( $resultSet$ ,  $tmp$ ); add( $S$ ,  $tmp$ );
      InvalidAndUpateReg( $reg, rIndex$ );
      InvalidAndUpateReg( $regFirst, rIndex$ );
  if  $isExtendable(reg, regSec, \kappa)$  then
    PlotRegion  $tmp = MERGE(reg, regSec, !frontFirst)$ ;
    float  $error = COMPUTE\_ERROR(tmp)$ ;
    if  $error < maxError$  then
      add( $resultSet$ ,  $tmp$ ); add( $S$ ,  $tmp$ );
      InvalidAndUpateReg( $reg, rIndex$ );
      InvalidAndUpateReg( $regSec, rIndex$ );
return  $resultSet$ ;

```

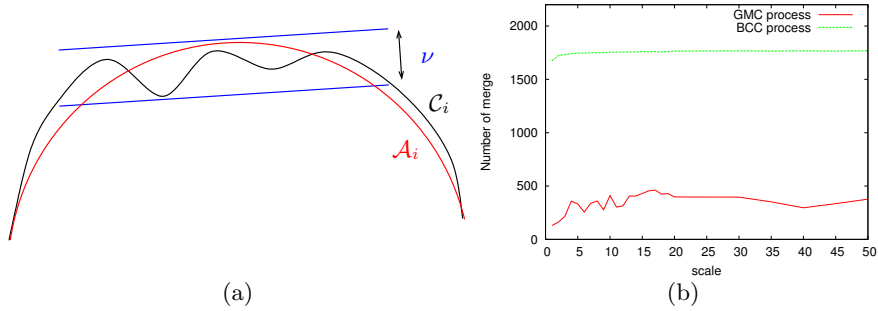


Fig. 2. Illustration of the blurred segment recognition of width ν illustrated in blue (a). A resulting arc detection at the scale $E_{max} = \nu$ is illustrated in red. Comparison of the number of merges of circle arcs when the reconstruction is defined with GMC estimator or with BCC estimator by using different scales (horizontal axis) (b).

comes from the stability of the GMC estimator (see [11] for detailed comparisons).

4 Experiments and comparisons

In this section, the experiments and comparisons were performed on a MacBook Pro running *Mac OS X 10.6.4*, with a processor 2.8 GHz *Intel Core 2 Duo* and 4GB of memory. Note that most of the experiments presented here are available online [12]: the reader may here tests and compares the different methods with its own shapes. First, we reconstruct the kangaroo shape of Fig. 4. Images (a-d) show the reconstructions obtained with the GMC estimator. For all experiments of the figure, resulting arcs are represented alternatively in blue and green color while straight segments are represented in red. The comparison between GMC and BCC (images (e-h)) curvature estimators shows a relative equivalence of the reconstructions: approximately the same accuracy with the same number of primitives (slight advantage of accuracy for GMC). However BCC is much slower, especially at large scale. Note that the initial constant curvature intervals were defined from the same precision parameter ϵ set to $10e - 6$ for both GMC and BCC estimator.

We also compare our proposed method with the NASR method of Nguyen, who proposed a linear time algorithm for approximate circle arc recognition (chapter 4, page 133) [15]. The method is based on the representation of the contour in the tangent space and shares with the GMC estimator the preprocess with blurred segments (thickness is used as a scale parameter for comparisons). NASR method is faster than GMC and BCC approach (Fig. 4, images (i-l)), but is much less accurate at a comparable scale (i.e. for the same number of primitives). The Fig. 3 displays the number of primitives (segments and arcs) according to the scale. GMC and BCC show comparable evolution while circle arc primitives disappear in NASR at large scales.

As mentioned in Section 2, we also perform comparisons with the classical visual curvature reconstruction ([13], see Fig. 5), later called VC. Depending on its parameters, VC is generally faster than GMC, especially at large scale. However, for a given number of primitives, the GMC reconstruction always achieves better accuracy than VC (Hausdorff distance about 6-7 times smaller). Note

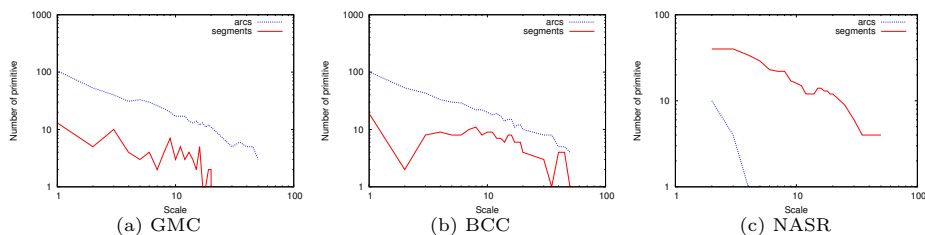


Fig. 3. Primitive evolution through the change of scale of the shape of Fig. 4.

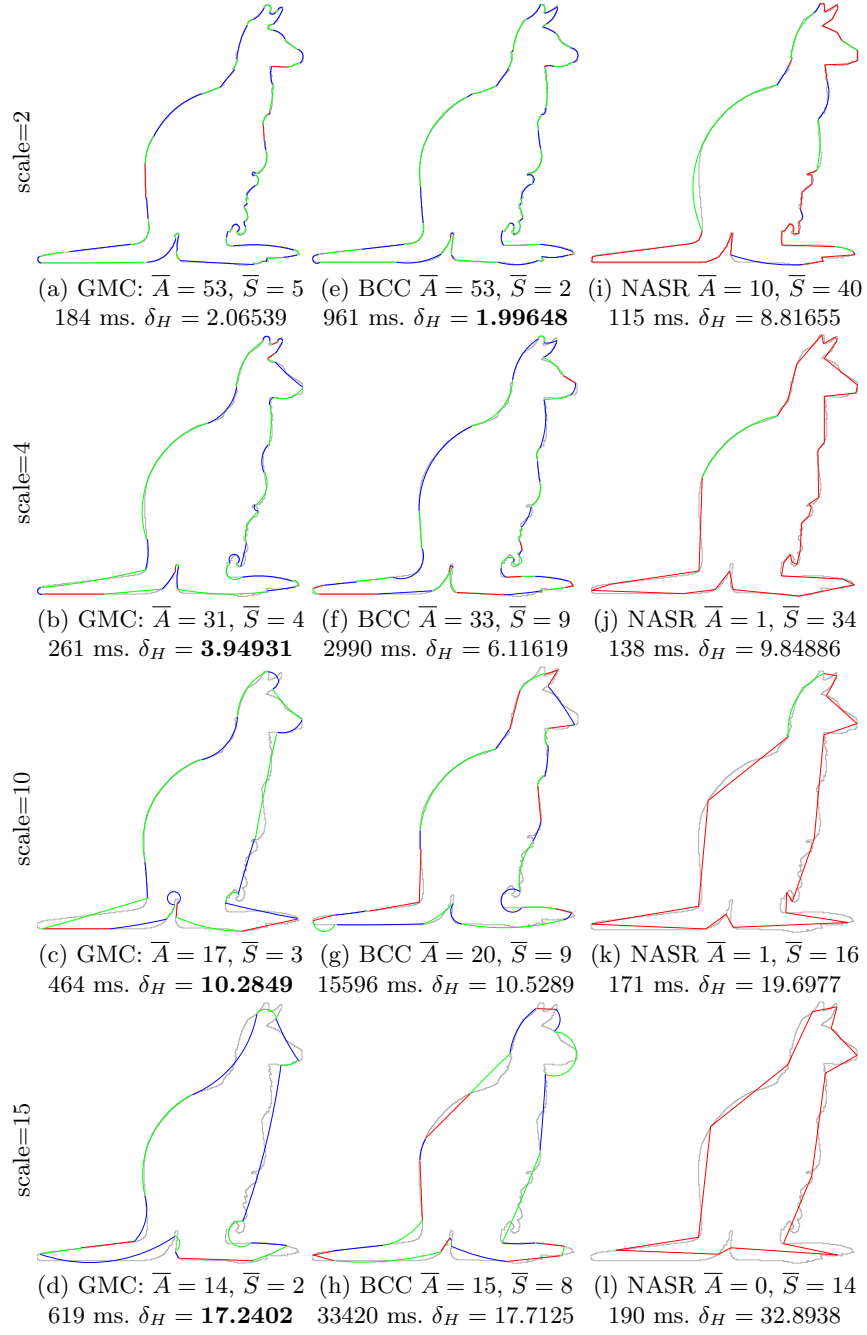


Fig. 4. Results and comparisons using the Hausdorff distance δ_H of the same reconstruction process by using GMC and BCC curvature estimators (a-h). Images (i-l) show for comparison, the result with NASR method by using the same scale parameter.

that arcs and segments represent the same cost since each reconstructed arc interpolates the middle region point C_{C_i} (second method of arc reconstruction described in Section 3). To conclude the experimentation, a last comparison was performed on a real photography of a drawing (Fig. 4) with the three different methods GMC, NASR and VC. The resulting representations confirm that GMC is always more precise for a given number of primitives.

5 Conclusion

This paper has proposed a simple method to reconstruct a digital contour with circular arcs, given a scale parameter that is simply the maximal Hausdorff error. Although the method is not specific to one curvature estimator, the GMC estimator has shown the best adequacy with our algorithm, since it gives precise results while keeping a reasonable execution time. The comparisons with the recent works demonstrate the quality of the proposed reconstruction. In future works, we plane to integrate more information in the reconstruction process, namely the automatic noise detection and the information on flat/curve contour parts [10], in order to obtain a parameter free contour reconstruction.

References

1. Bretin, E., Lachaud, J.O., Oudet, E.: Regularization of discrete contour by willmore energy (2010), submitted.
2. Chen J.-M., Ventura J.A., W.C.: Segmentation of planar curves into circular and line segments. *Image Vision and Computing* 14, 71–83 (1996)
3. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear time. *Computers & Graphics* 30(1) (2006)
4. E. Bodansky, A.G.: Approximation of a polyline with a sequence of geometric primitives. In: *ICIAR. LNCS*, vol. 4142, pp. 468–478 (2006)
5. Esbelin, H.A., Malgouyres, R.: Convergence of binomial-based derivative estimation for c 2 noisy discretized curves. In: *Proc. of DGCI. No. 5810 in LNCS* (2009)
6. Feschet, F.: Multiscale analysis from 1d parametric geometric decomposition of shapes. In: *IEEE (ed.) Int. Conf. on Pattern Recognition*. pp. 2102–2105 (2010)
7. Horng, J.H.: An adaptive smoothing approach for fitting digital planar curves with line segments and circular arcs. *Pat. Rec. Letters* 24(1-3), 565–577 (2003)
8. Horng, J.H., Li, J.T.: A dynamic programming approach for fitting digital planar curves with line segments and circular arcs. *Pat. Rec. Letters* 22(2), 183–197 (2001)
9. Kerautret, B., Lachaud, J.O.: Curvature estimation along noisy digital contours by approximate global optimization. *Pattern Recognition* 42(10), 2265 – 2278 (2009)
10. Kerautret, B., Lachaud, J.O.: Multi-scale analysis of discrete contours for unsupervised noise detection. In: *IWCIA. LNCS*, vol. 5852, pp. 187–200 (2009)
11. Kerautret, B., Lachaud, J.O., Naegel, B.: Curvature based corner detector for discrete, noisy and multi-scale contours. *IJSM* 14(2), 127–145 (December 2008)
12. Kerautret, B., Lachaud, J.O., Nguyen, T.P.: Curvature based contour representation demo. <http://kerrecherche.iutsd.uhp-nancy.fr/CBCContours> (2010)
13. Liu, H., Latecki, L.J., Liu, W.: A unified curvature definition for regular, polygonal, and digital planar curves. *Int. J. Comput. Vision* 80(1), 104–124 (2008)

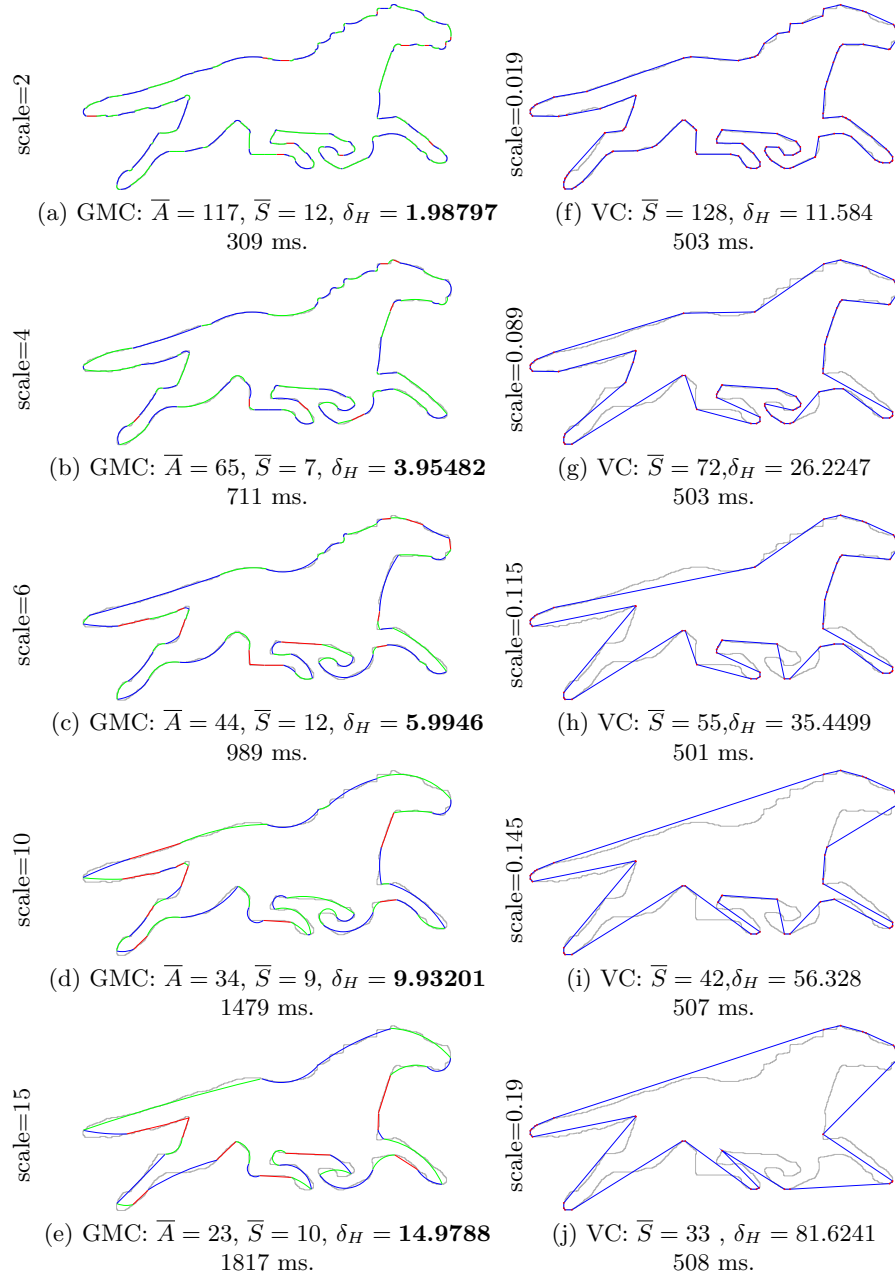


Fig. 5. Reconstruction results of the proposed method (a-e) at different scales. Comparisons with the Visual Curvature based approach.

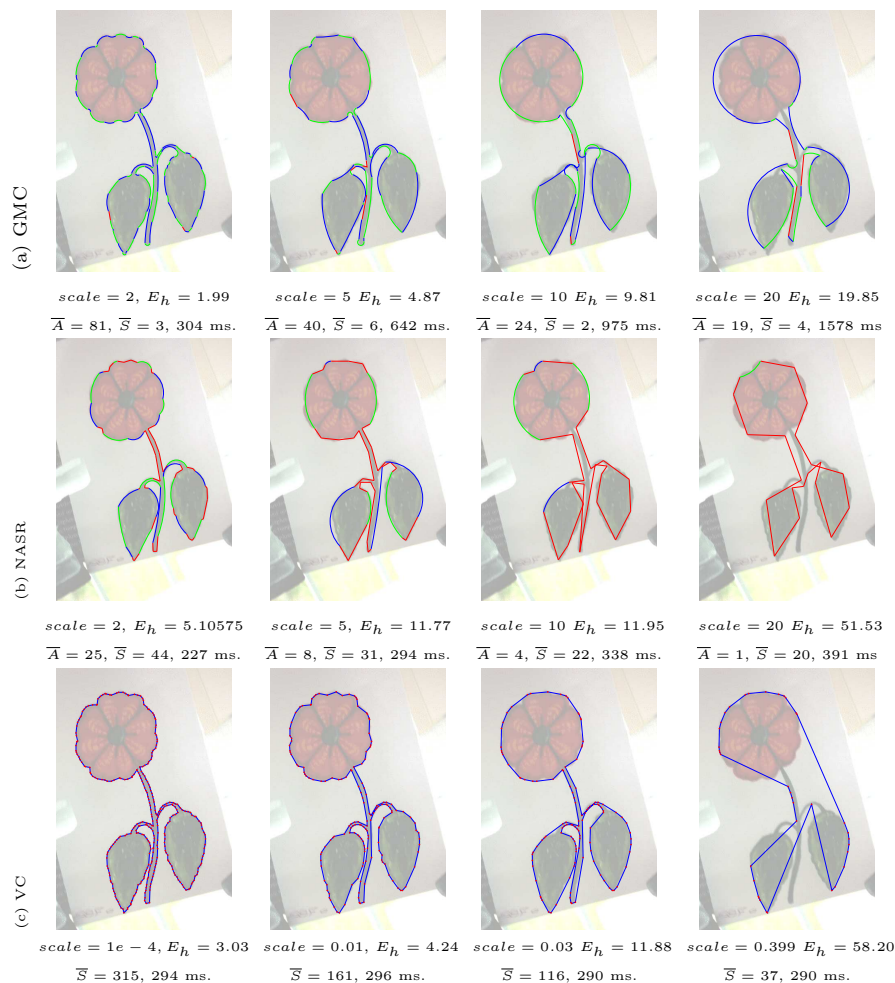


Fig. 6. Experiments and comparisons of three methods applied on a photograph of a flower drawing. For GMC and NASR, the resulting circle arcs are represented alternatively with blue and green color while straight segments are represented in red.

14. Malgouyres, R., Brunet, F., Fourey, S.: Binomial convolutions and derivatives estimation on noisy discretizations. In: Proc. DGCI. pp. 370–379 (2008)
15. Nguyen, T.P.: Etude des courbes discrètes: applications en analyse d’images. Ph.D. thesis, Nancy University - LORIA (2010), (in french)
16. Rosin, P.L., West, G.A.W.: Segmentation of edges into lines and arcs. Image Vision Comput. 7(2), 109–114 (1989)
17. Salmon, J.P., Debled-Rennesson, I., Wendling, L.: A new method to detect arcs and segments from curvature profiles. In: ICPR. pp. 387–390. IEEE (2006)
18. Tortorella, F., Patraccone, R., Molinaro, M.: A dynamic programming approach for segmenting digital planar curves into line segments and circular arcs. In: ICPR (2008)