Robust curvature extrema detection based on new numerical derivation

Cédric Join, Salvatore Tabbone

▶ To cite this version:

HAL Id: inria-00300799

https://hal.inria.fr/inria-00300799

Submitted on 20 Jul 2008

Robust curvature extrema detection based on new numerical derivation

Cédric Join[1,3] and Salvatore Tabbone[2,3]

[1] INRIA-ALIEN CRAN (UMR-CNRS 7039)-Université Henri Poincaré
[2] INRIA-QGAR LORIA (UMR 7503)-Université Nancy 2
[3] Campus scientifique BP 239, 54506 Vandœuvre-lès-Nancy, France
`Cedric.Join@cran.uhp-nancy.fr, tabbone@loria.fr`

Abstract. Extrema of curvature are useful key points for different image analysis tasks. Indeed, polygonal approximation or arc decomposition methods used often these points to initialize or to improve their algorithms. Several shape-based image retrieval methods focus also their descriptors on key points. This paper is focused on the detection of extrema of curvature points for a raster-to-vector-conversion framework. We propose an original adaptation of an approach used into nonlinear control for fault-diagnosis and fault-tolerant control based on algebraic derivation and which is robust to noise. The experimental results are promising and show the robustness of the approach when the contours are bathed into a high level speckled noise.

1 Introduction

Vectorization consists in analyzing a raster image to convert its pixel representation into vector representation. Often it is the last step before a recognition process. Then, it is a central part as it deals with converting a image into vectors suitable for further analysis/recognition steps. The basic assumption is that such representation is more suitable for further interpretation of the image; this typically holds for a lot of problems in scanned graphical document, in image processing or in pattern analysis. Therefore, the qualities of these subsequent treatments are related to the precision of the provided vectors. Many vectorization methods have been designed throughout these years. The more common vectorization is designed by polygonal approximation methods[4]. Some of the polygonal methods are based on heuristic algorithms[10, 19] minimizing a local cost and other on optimal solutions[8, 11] minimizing a global cost. A lot of different technics have been used like sequential tracing approach[15], split-and-merge methods[12], dominant point detection[9, 18], genetic algorithms [17] and dynamic programing[11].

Usually these approaches are fed with points provided by the skeleton, contours or key points of a shape. Even if a great number of approaches[14, 16] have been proposed this last 30 years, we cannot say that all the problems are completely solved. There is still a major problem of precision at the junction points, robustness and stability in the vectorization process. After approximation, it is often necessary to perform some post-processing, to find better positions for the junction points[13], to merge some vectors and remove some others.

This paper is focused on this problematic for which it is important to detect on a set of points those which have a high curvature helpful for further subsequent treatments. By definition, for a plane curve given parametrically as $(x(u), y(u))$, the curvature is:

$$\kappa(u) = \frac{\left| \left(\frac{dx(u)}{du}\right) \left(\frac{d^2 y(u)}{du^2}\right) - \left(\frac{d^2 x(u)}{du^2}\right) \left(\frac{dy(u)}{du}\right) \right|}{((x(u)^2 + y(u)^2)^{3/2}} \tag{1}$$

We can remark that the estimation of the derivatives play an important role for the precision of the curvature. In this perspective, we propose a method to define the derivatives of the contours which is robust to noise. The derivatives are defined into an algebraic framework and based on work defined in nonlinear control theory for the detection of abrupt changes, fault-diagnosis and fault-tolerant control[2, 3, 6, 7].

In the next section 2 we recall the definitions of algebraic derivatives. We give an example for computing these derivatives. Then, experimental results are given in section 3 for different levels of noise. Finally, we conclude and give perspectives to our work (section 4).

2   Derivatives of noisy signals

Consider a real-valued time function $x(t)$ which is assumed to be analytic on some interval $t_1 \leq t \leq t_2$. Assume for the sake of simplicity that $x(t)$ is analytic around $t = 0$ and let its truncated Taylor expansion be:

$$x(t) = \sum_{\nu=0}^{N} x^{(\nu)}(0) \frac{t^\nu}{\nu!} + o(t^N)$$

Approximate $x(t)$ in the interval $(0, \varepsilon)$, $\varepsilon 0$, by a polynomial $x_N(t) = \sum_{\nu=0}^{N} x^{(\nu)}(0) \frac{t^\nu}{\nu!}$ of degree N. The usual rules of symbolic calculus in Schwartz's distributions theory yield:

$$x_N^{(N+1)}(t) = x(0)\delta^{(N)} + \dot{x}(0)\delta^{(N-1)} + \cdots + x^{(N)}(0)\delta$$

where δ is the Dirac measure at 0. From $t\delta = 0$, $t\delta^{(\alpha)} = -\alpha\delta^{(\alpha-1)}$, $\alpha \geq 1$, we obtain the following triangular system of linear equations to determine the estimated values $[x^{(\nu)}(0)]_e$ of the derivatives[4] $x^{(\nu)}(0)$:

$$\begin{aligned} t^\alpha x^{(N+1)}(t) = t^\alpha &\left([x(0)]_e \delta^{(N)} \right. \\ &+ [\dot{x}(0)]_e \delta^{(N-1)} + \cdots + [x^{(N)}(0)]_e \delta \left. \right) \\ \alpha = 0, &\ldots, N \end{aligned} \tag{2}$$

The time derivatives of $x(t)$, the Dirac measures and its derivatives are removed by integrating (with respect to time) both sides of equation (2) at least N times:

$$\begin{aligned} \int^{(\nu)} \tau_1^\alpha x^{(N+1)}(\tau_1) = \int^{(\nu)} \tau_1^\alpha &\left([x(0)]_e \delta^{(N)} \right. \\ &+ [\dot{x}(0)]_e \delta^{(N-1)} + \cdots + [x^{(N)}(0)]_e \delta \left. \right) \\ \nu \geq N, &\ \alpha = 0, \ldots, N \end{aligned} \tag{3}$$

---

[4] The derivatives are *linearly identifiable* [3].

where $\int^{(\nu)} = \int_0^T \int_0^{\tau_{\nu-1}} \dots \int_0^{\tau_1}$ is an iterated integral. A quite accurate value of the estimates may be obtained with a small time window $[0, T]$.

For more details, the reader is invited to see, *e.g.*, [2] for various applications to nonlinear control (state and parametric estimations, fault-diagnosis and fault-tolerant control) and references therein for applications to signal processing.

2.1 Noise attenuation

These iterated integrals are moreover low pass filters[5]. They smooth highly fluctuating noises, which are usually dealt with statistical settings. We therefore do not need any knowledge on the statistical properties of the noises (see [6] for details on the numerical implementation).

2.2 Example

Let us consider the signal locally represented by the polynomial function:

$$p_2(t) = a_0 + a_1 t + \frac{a_2}{2} t^2$$

calculations are more easy in operational domain, thus the previous equation is written in this domain as:

$$P_2(s) = \frac{a_0}{s} + \frac{a_1}{s^2} + \frac{a_2}{s^3}$$

It is clear that to estimate a_1, it is to estimate the first derivative of the signal at $t = 0$. Multiplying both sides of the equality by s^3 and taking the derivative with respect to s, i.e. $\frac{d}{ds}$:

$$3s^2 P_2(s) + s^3 \frac{d}{ds} P_2(s) = 2sa_0 + a_1$$

divide by s:

$$3s P_2(s) + s^2 \frac{d}{ds} P_2(s) = 2a_0 + \frac{a_1}{s}$$

take one more times the derivative with respect to s:

$$3P_2(s) + 5s \frac{d}{ds} P_2(s) + s^2 \frac{d^2}{ds^2} P_2(s) = -\frac{a_1}{s^2}$$

By multiplying both sides of equations by $s^{-\nu}$, with ν large enough, here $\nu = 3$, only iterated integrals of P_2 appear

$$3 \frac{1}{s^3} P_2(s) + 5 \frac{1}{s^2} \frac{d}{ds} P_2(s) + \frac{1}{s} \frac{d^2}{ds^2} P_2(s) = -\frac{a_1}{s^5}$$

Using correspondence rules from operational domain to time domain and the Cauchy rule (*cf.* Appendix), the estimation of derivative is given by:

$$a_1 = \frac{4!}{T^4} \int_0^T \left(3\frac{(T-t)^2}{2} - 5(T-t)t + t^2 \right) P_2(t) dt$$

---

[5] The iterated integrals may be replaced by more general low pass filters, which are defined by strictly proper rational transfer functions.

where T is the length the estimation time window. A quite short time window is sufficient to obtain accurate values of a_1.

3 Experimental results

Several tests are now presented in order to highlight the influence of different parameters on our method. More precisely, we will study the behaviour of the approach according to the length of integration window (previously denoted by T) and the curvature threshold. The aim is to show the noise robustness of the proposed method. For each simulation, the curvature is defined on a sliding window with a starting point and contours are considered as closed. In this context, the position and the identification of extrema have to be considered relatively to this starting point and for this reason all the first curvature inside the starting window should not be considered in all the following figures.

3.1 Window length influence

The figure 1 presents the behaviour of the curvature related to the integration window size (T) (see (1)).

First, we consider the *Rectangle* picture (100×200 pixels). The important noise level (see figure 1.a) is an independent random choice among the set $\{-3, -2, -1, 0, 1, 2, 3\}$ of pixel positions which are added on each coordinate components.

The experimental results show that the size of integration window has an influence on the residual noise. More the size is large less the curvature estimation is corrupted. However, the window size does not really influence the extrema position compared to the impulse position. We can remark that in each case, the four rectangle corners correspond to the high curvatures in figures 1.b-i for several window sizes.

In figure 1.j, symbols (different kinds of symbols are reported and associated to different window sizes) indicate the high curvature localization for the different levels of noise defined in figures 1.b to i. We can remark that the position of the extrema are either on the rectangle corners or near.

3.2 Noise influence

On figure 2, we present results for several noise levels using the same threshold for a rectangle of 20×30 pixels (see figures 2.a to d). The small form size implies to use very small evaluation windows (here, $T = 20$ pixels). This case should be considered as extreme since the original rectangle is very difficult to recognize particularly in the figure 2.d and its small size adds more complexity to reduce the noise.

Positions of high curvature points are given in the figure 2.i where symbols (o, blue), (x, magenta), (+, red) and (*, black) are respectively associated to figures 2.a-d. Clearly the noise corrupts the results, i.e. more noise level decreases the precision of the extrema position. However, for moderate noise the points are on the corner or very closed. The lines joining the four extrema obtained with our algorithm, for each levels of noise, are presented in dashed lines and superimposed on the figures 2.a-d.

On figure 3, we show a real case on the fish shape. Similar experiments than previously are conducted. We can remark that, with the same threshold for important noise levels, results are degraded. This can be explained by the fact that due to the noise, high curvatures are more difficult to discriminate than for the rectangle case [6]. In this context for real images, it will be better to adjust the threshold according to the number of extrema wanted by a user.

3.3 Threshold adjustment

With the same simulation conditions than the figure 2 where symbols (o, blue), (x, magenta), (+, red) and (*, black) are now respectively associated to figures 3.a-d, we propose to automatically adjust the threshold. It is computed such that the first 20 main curvatures are detected. The threshold thus computed is respectively 0.057, 0.056, 0.065 and 0.073 and, obviously, increases with the noise level. The main result can be evaluated thanks to figure 3.j, where for each case and in spite of the different noise levels, the high curvatures found are approximately the same. It was not the case with one fixed threshold for all noise levels (see figure 3.i).

4 Conclusion

In this paper, we have proposed an original adaptation of an approach used into nonlinear control based on algebraic derivation to the detection of extrema points. The experimental results are promising and show the robustness of the approach for high level speckled noise. Further work will be devoted to use these points to describe a shape into a polyline using a fitting framework.

5 Appendix

Let us recall some usual transformation lows

$$
\begin{array}{cc}
operational\ domain & time\ domain \\
\frac{1}{s^{n+1}} \longrightarrow & \frac{t^n}{n!} \\
\frac{1}{s^n}F(s) \longrightarrow & \int^{(n)} f(t)dt \\
\frac{d^n}{(ds)^n} \longrightarrow & (-t)^n
\end{array}
$$

where $\int^{(n)} f(t)dt$ is the iterated integral of order n, i.e $\int_0^T \cdots \int_0^{\tau_2} f(\tau_1)d\tau_1 d\tau_2 \cdots d\tau_n$. This last equation, thanks to Cauchy rule, is rewritten as $\int^{(n)} f(t)dt = \int_0^T \frac{(T-t)^{n-1}}{(n-1)!} dt$.

---

[6] It is well known that discontinuity detection on signal derivatives is an open problem, see [1, 5, 7] for more details.

References

1. M. Basseville, I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*, Prentice-Hall, 1993.
2. M. Fliess, C. Join, H. Sira-Ramírez, *Non-linear estimation is easy*, *Int. J. Modelling Identification Control*, T. 3, 2008. Available at http://hal.inria.fr/inria-00158855/fr/.
3. M. Fliess, H. Sira-Ramírez, *Closed-loop parametric identification for continuous-time linear systems via new algebraic techniques*, in H. Garnier & L. Wang (Eds): *Continuous-Time Model Identification from Sampled Data*, Springer, Berlin, 2008. Available at http://hal.inria.fr/inria-00114958/fr/.
4. A. Kolesnikov, *Efficient algorithms for vectorization and polygonal approximation*, PhD thesis, 2003, University of Joensu, Finland.
5. M. Lavielle, *Using penalized contrasts for change-point problem*, *Signal Processing*, vol. 85, pp. 1501–1510, 2005.
6. M. Mboup, C. Join, M. Fliess, *A revised look at numerical differentiation with an application to nonlinear feedback control*, *Proc. 15th Mediterrean Conf. Control Automation (MED'2007)*, Athènes, 2007. Available at http://hal.inria.fr/inria-00142588/fr/.
7. M. Mboup, C. Join, M. Fliess, *A delay estimation approach to change-point detection*, *Proc. 16th Mediterrean Conf. Control Automation (MED'2007)*, Ajaccio, 2008. Available at http://hal.inria.fr/inria-00179775/fr/.
8. J-H. Lee and J-W Chung and J-K Kim, *Optimal vertex adjustement method using Viterbi algorithm for vertex-based shape coding*, Electronics Letters, 35, 1999.
9. M. Marji and P. Siy, *A new algorithm for dominant points detection and polygonization of digital curves*, Pattern recognition, 36, 2003.
10. A. Mikheev and L. Vincent and V. Faber, *High-Quality Polygonal Contour Approximation Based on Relaxation*, International conference on document analysis and recognition, 2001.
11. J.C Perez and E. Vidal, *Optimum Polygonal Approximation of Digitized Curves*, Pattern recognition letters, 15(8), 1994.
12. B. K. and Ray K. S., *A New Split-and-merge Technique for Polygonal Approximation of Chain Coded Curves*, Pattern recognition letters, 16, 1995.
13. M. Röösli, G. Monagan, *Adding Geometric Constraints to the Vectorization of Line Drawings*, Graphics Recognition—Methods and Applications, Lecture Notes and Computer Science, 1072, 1996.
14. P. L. Rosin, *Assessing the behaviour of polygonal approximation algorithms*, Pattern recognition, 36, 2004.
15. J. Song, F. Su, C.-L. Tai and S. Cai, *An Object-Oriented Progressive-Simplification Based Vectorization System for Engineering Drawings: Model, Algorithm, and Performance*, IEEE Transactions on Pattern Analysis, 24(8), 2002.
16. K. Tombre and S. Tabbone, *Vectorization in graphics recognition: to thin or not to thin*, Proceedings of the 15th International Conference on Pattern Recognition, Barcelona (Spain), vol 2, 2000.
17. V.J. Traver, G. Recatalá and J.M Inesta, *Exploring the performance of genetic algorithms as polygonal approximators*, International Conference on Pattern Recognition, 2000.
18. W-Y. Wu, *An adaptive method for detecting dominant points*, Pattern Recognition, 36(10),2003.
19. P-Y. Yin, *Algorithms for straight line fitting using K-means*, Pattern recognition letters, 19, 1998.
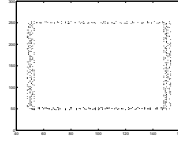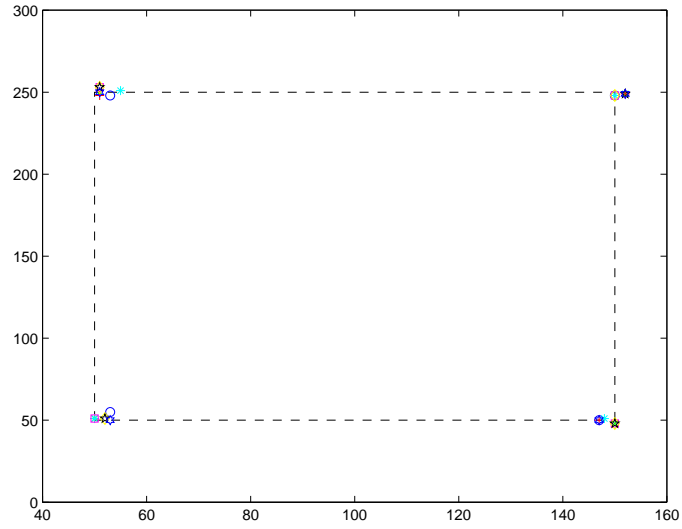
(a) Noisy rectangle
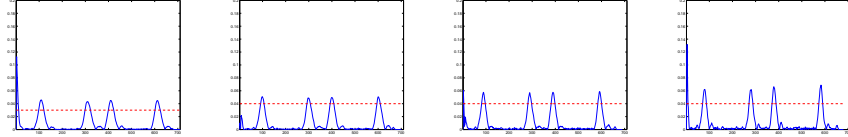
(b) Window of 100 pixels *(c)* Window of 90 pixels *(d)* Window of 80 pixels *(e)* Window of 70 pixels
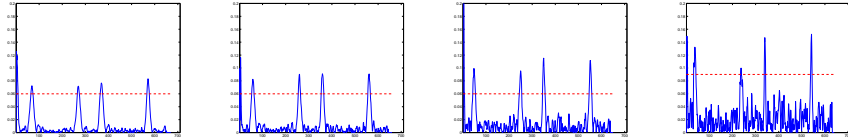
(f) Window of 60 pixels *(g)* Window of 50 pixels *(h)* Window of 40 pixels *(i)* Window of 30 pixels

(j) High curvature points for various noise levels

Fig. 1. Influence of window size for the curvature $\kappa(\bullet)$

(a) Without noise   *(b)* With noise= $\{-1, 0, 1\}$ pixel   *(c)* With noise= $\{-2, ..., 2\}$ pixel   *(d)* With noise= $\{-3, ..., 3\}$ pixel

(e) Without noise   *(f)* With noise= $\{-1, 0, 1\}$ pixel   *(g)* With noise= $\{-2, ..., 2\}$ pixel   *(h)* With noise= $\{-3, ..., 3\}$ pixel

(i) High curvature points following various noise levels
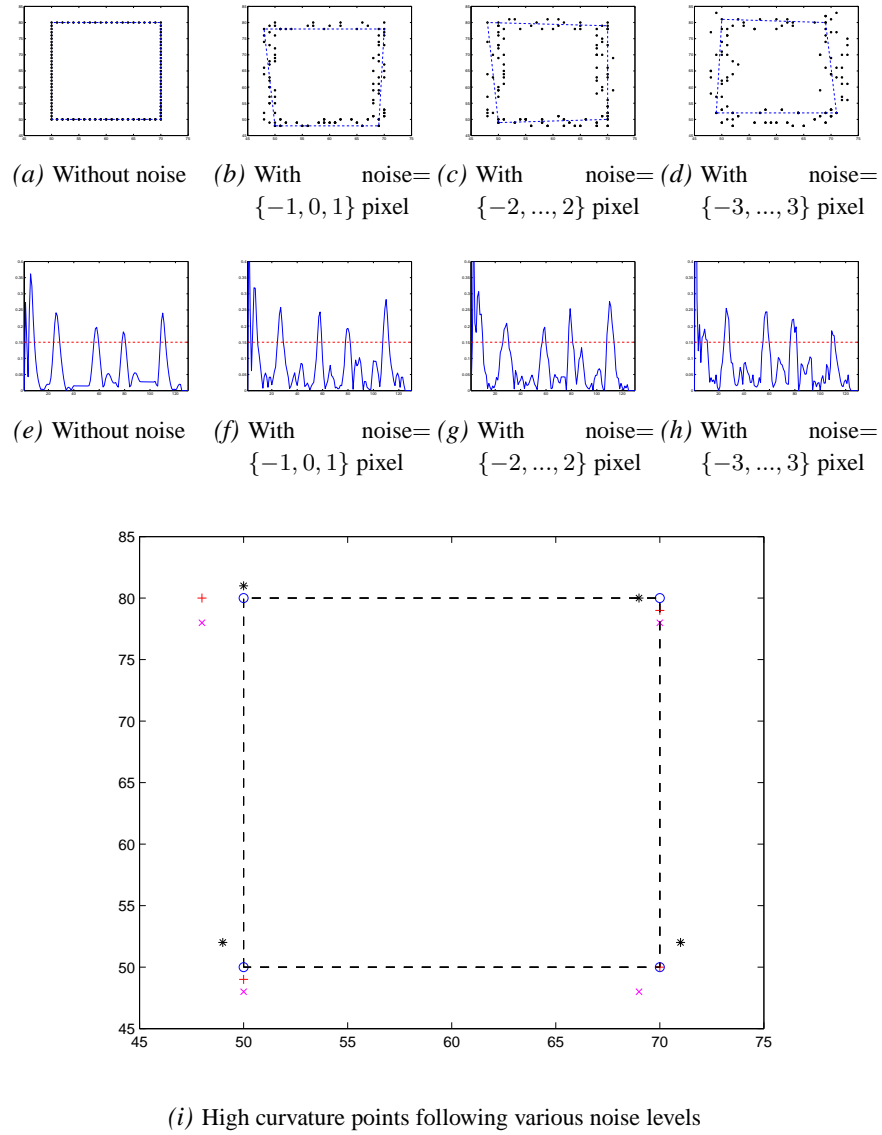
Fig. 2. *Rectangle* segmentation

(a) Without noise   *(b)* With noise= $\{-1,0,1\}$ pixel   *(c)* With noise= $\{-2,...,2\}$ pixel   *(d)* With noise= $\{-3,...,3\}$ pixel

(e) Without noise   *(f)* With noise= $\{-1,0,1\}$ pixel   *(g)* With noise= $\{-2,...,2\}$ pixel   *(h)* With noise= $\{-3,...,3\}$ pixel

(i) High curvature points following various noise levels

(j) Automatic threshold (0.057, 0.056, 0.065 and 0.073) adjustment following various noise levels for the 20 main curvatures

Fig. 3. *Fish* segmentation