

# Discrete Curvature Based on Osculating Circle Estimation

David Coeurjolly, Serge Miguet, and Laure Tougne

Laboratoire E.R.I.C.  
Université Lyon 2  
5 av. Pierre Mendès-France  
69676 Bron cedex, France  
{dcoeurjo,miguet,ltougne}@eric.univ-lyon2.fr

**Abstract.** In this paper, we make an overview of the existing algorithms concerning the discrete curvature estimation. We extend the Worring and Smeulders [WS93] classification to new algorithms and we present a new and purely discrete algorithm based on discrete osculating circle estimation.

**Keywords.** Discrete Geometry, Curvature Calculus, Discrete Circles.

## Introduction

Boundary analysis is an important step in many computer vision paradigms in which contours are used to extract conceptual information from objects. In such a contour analysis, the curvature is a commonly used geometrical invariant to extract characteristic points. In 3D medical imaging or in 3D snow sample analysis, the curvature measurement on surfaces is also used as a registration tool [MLD94, TG92] or as a physical or mechanical characteristics extraction tool [Pie99].

In classical mathematics, the curvature calculus is clearly defined and its properties are well known but when we want to apply this calculus on discrete data (2D or 3D discrete images), two different approaches are possible: we can first change the model of the data and put them into the classical continuous space by using interpolations or parameterizations of mathematical objects (B-splines, quadratic surfaces) on which the continuous curvature can be easily computed [Cha00, HHTS98]. Otherwise, we can try to express discrete curvature definitions and properties, and make sure that these new definitions are coherent with the continuous ones.

In the first approach, we have two main problems: the first one is that there exists a great number of parameterization algorithms in which some parameters have to be set according to the inputs. In order to provide a given accuracy, we have to reduce the input area and thus to limit our method. The second problem is that these algorithms have got a prohibitif computational time when we use large input data such as in medical imaging.

In a discrete approach, there are three classical ways to define this differential operator : we have the tangent orientation based algorithms, the second discrete derivative estimation based methods and finally the osculating circle based methods. In the continuous space, these definitions are equivalent but in the digital space, they lead to specific classes of algorithm.

In this paper, we propose an optimal algorithm to compute the curvature of a discrete curve in 2 or 3 dimension based on circle estimation. This algorithm holds two important properties: it's a purely low level algorithm that does not need neither preprocessing tool nor data linked parameters, and it only lies on the discrete model.

## 1 Framework and Related Methods

In this section, we present the different definitions of the continuous curvature we can find in the literature. We explain what they become on the discrete space and we extend the Worring and Smeulder 's [WS93] classification to new discrete algorithms.

### 1.1 Continuous Definitions

Given a continuous objet  $\mathcal{X}$  with boundary  $\partial\mathcal{X}$ , we consider a curvilinear abscissa parametrization  $x(s)$  of the boundary. We have three classical ways to define the curvature of a curve or a path  $x$ . The first one is based on the norm of the second derivative of the curve.

#### Definition 1 (Second derivative based curvature)

$$k(s) = \text{sign}\|x''(s)\|$$

where *sign* is either -1 or 1 according to the local convexity of the curve.

We can also define the curvature using the directional changes of the tangent. The curvature is obtained by computing the angle variations between the tangent  $t$  of the curve and a given axis.

#### Definition 2 (Tangent Orientation based curvature)

$$k(s) = \theta'(s) \quad \text{where} \quad \theta(s) = \angle(t(s), \text{axis})$$

Finally, we have a geometrical approach of the curvature definition given by the osculating circle of radius  $r(s)$ .

#### Definition 3 (Osculating circle based curvature)

$$k(s) = \text{sign}\left(\frac{1}{r(s)}\right)$$

In a continuous space, all these definitions are obviously equivalent whereas in discrete space, they lead to specific algorithms.

## 1.2 Discrete Curvature Computation

Now, we consider  $X$  as a digitization of the object  $\mathcal{X}$  on a discrete grid of resolution  $h$  using a digitization process  $D_h$  (i.e.  $X = D_h(\mathcal{X})$ ). The goal of this paper is not to study the best digitization process. Based on the discrete object  $X$ , we can define the boundary  $dX$  of  $X$  as the set of eight-connected points such that each of them have a four-connected neighbor in the complement of  $X$  [KR96].

We can now formulate the discrete curvature computation: *given a discrete boundary  $dX$  how can we compute the curvature at each point?*

First of all, we detail existing algorithms that can be classified according to the previous definitions. Just remark that due to the difficulty to provide an accurate first derivative calculus of a discrete curve, the second derivative based methods are not really tractable. Indeed, Worring and al. have presented a definition of the curvature using second derivative at a point  $p_i$  using derivative Gaussian kernels [WS93].

**Tangent-Orientation based methods.** A first definition of the tangent orientation based discrete curvature is computed using a local angle [BD75,CM91]: given a neighborhood  $m$ , the discrete curvature at a point  $p_i$  of  $dX$  is:

$$k_h(p_i) = \frac{\angle(p_{i-m}p_i, p_i p_{i+m})}{|p_{i-m}p_i| + |p_i p_{i+m}|}$$

In [RJ73], Rosenfeld and al. proposed a new curvature calculus that adapts the window size  $m$  to local characteristics of the discrete curve.

In such a way, Worring and Smeulders [WS93] proposed curvature definitions based on the variation of the angle between the best straight line fitting to the data and an axis in a given neighborhood of  $p_i$ :

$$k_h(p_i) = \frac{\theta(p_i) * \mathcal{G}'_\sigma}{1.107} \quad \text{where} \quad \theta(p_i) = \angle(\text{Line fitting}(p_i, m), x - \text{axis})$$

Worring and al. computed the line fitting using an optimization in a window of size  $m$ .  $\mathcal{G}'_\sigma$  corresponds to the derivative of a Gaussian kernel with parameter  $\sigma$  to estimate the variation of angle.

In [Via96], Vialard proposed a purely discrete definition of the line fitting process. Let us remember the definition of a discrete straight line given by Réveillès [Rev91]. A set of discrete points belong to the arithmetical discrete straight line  $D(a, b, \mu, \omega)$  (with  $a, b, c, \mu, \omega \in \mathbb{Z}$ ) if and only if each point  $p(x, y)$  of the set satisfies:

$$\mu \leq ax - by < \mu + \omega$$

$a/b$  denotes the slope of the line,  $\mu$  the lower bound on the grid and  $\omega$  the thickness (in this paper, we only consider *naive straight lines* with  $\omega = \sup(|a|, |b|)$ ).

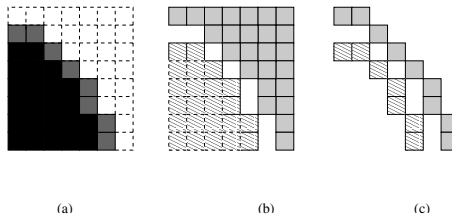
Based on this definition, Vialard defines the discrete tangent at a point  $p$  of a curve as the longest discrete straight line centered in  $p$  that belongs to the curve. This calculus is driven by the Debeld's straight line recognition algorithm [DR95]. Vialard defines the curvature at a point  $p$  of a discrete curve with the equation:

$$k_h(p_i) = \frac{\theta(p_i) * \mathcal{G}'_\sigma}{1.107} \quad \text{where} \quad \theta(p_i) = \angle(\mathcal{T}(p_i), x - axis)$$

where  $\mathcal{T}(p_i)$  denotes the discrete tangent centered at  $p_i$ .

In [FT99], Feschet and Tougne proposed an optimal algorithm to compute the tangent at each point of a discrete curve.

**Osculating Circle based Methods.** In this approach, the estimation of the best fitting circle often leads to statistical or optimization processes. Worring [WS93] proposed to optimize a distance between the smoothed data in a given window and Euclidean circles. In [Kov90], Kovalevsky proposed a geometrical approach to compute the set of possible arcs (maybe empty) separating two discrete sets of points using a *Voronoi-like* algorithm. We can use this algorithm to solve our arc fitting problem with computing the separating circle between the complement of the discrete boundary which has two connected components. Obviously we can reduce the size of the two sets considering the points connected to the boundary (see figure 1). The problem of this algorithm is that the computational cost of the calculus of the Voronoi cell in  $O(n^2)$  ( $n$  denotes the size of the curve) makes this approach not be tractable as compared with the Vialard's algorithm computational cost in  $O(n)$ .



**Fig. 1.** Preprocessing to the Kovalevsky's algorithm : (a) the discrete object  $X$  with its discrete boundary  $dX$ , (b) the two connected components of the complement of the boundary, (c) the resulting two sets which are the input of the Kovalevsky's algorithm.

**Discussion.** In the tangent orientation based approach, Vialard's algorithm shows how the discrete model and discrete operators solve the statistical problem of the line fitting process and so the curvature calculus in an optimal time.

However, such an algorithm needs a derivative filter and this leads to two problems: we have to estimate a smoothing parameter  $\sigma$  according to the data, and we use a non discrete operator. If we want to define in a purely discrete way the curvature on a discrete curve, the only possible approach is the geometrical one if we are able to recognize or estimate osculating discrete circles.

## 2 A Purely Discrete Optimal Time Algorithm

### 2.1 Discrete Circle Analysis

First, we have to define and to characterize discrete circles. Andres [And94] proposed a discrete arithmetical circle definition based on a double diophantian inequation:  $p(x, y)$  belongs to the discrete arithmetical circle  $C(x_o, y_o, r)$  (with  $x_o, y_o, r \in \mathbb{Z}$ ) if and only if it satisfies:

$$(r - \frac{1}{2})^2 \leq (x - x_o)^2 + (y - y_o)^2 < (r + \frac{1}{2})^2 \tag{1}$$

In the discrete straight line recognition algorithms, two main approaches can be extracted: the first one is a geometrical and a arithmetical approach [DR95,Kov90] roughly based on computing the narrowest strip that encloses the points. The second one is an algebraic approach [FST96] that solves the inequation system in  $\mathbb{Z}^n$  given by the double diophantian inequation at each point. Due to the non-linearity of the equation 1, none of these two approaches can be easily extended to the discrete circle recognition problem in a similar computational time as the Vialard’s algorithm. Therefore the idea is to develop an approximation algorithm.

In [DMT99] Tougne and al. present a new approach to define discrete circles: whatever the digitization process is, a discrete circle is the result of the intersections between the continuous circle  $x^2 + y^2 = r^2$  and the discrete grid. In the first octant, if we analyze the intersections between a continuous circle and the vertical discrete lines  $x = r - k$  ( $0 \leq k \leq r$ ), we obtain a bundle of parabolas  $\mathcal{H}_k: y = \sqrt{2kx + k^2}$ . Hence, each kind of discrete circle (arithmetical circles, Pitteway’s circles or Breshenam’s circles) is the result of a specific digitization of these parabolas (see figure 2 for an example of the discrete bundle of parabolas associated to the Pitteway’s circles).

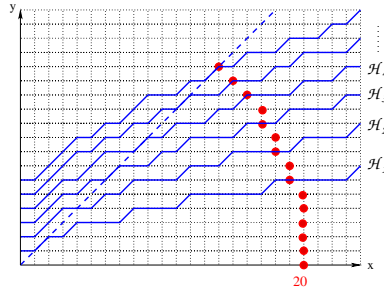
Using this bundle of parabolas, a discrete circle can be built in the first octant only with vertical patches. For example, if we consider the case of Pitteway’s circles, the height of the first vertical patch is given by the formula<sup>1</sup>:

$$h = \lceil \sqrt{2x + 1} \rceil$$

Just note that any kind of discrete circle can be entirely characterized with its bundle of discrete parabolas.

In the following, since Tougne shown the equivalence of discrete circles given by their parabolas, we choose to estimate arithmetical osculating circle and base

<sup>1</sup>  $\lceil x \rceil$  denotes the rounding to the closest integer.



**Fig. 2.** The Pitteway’s circle of radius 20 and its discrete parabolas given by the equation:  $y = \lfloor \sqrt{2kx + k^2} \rfloor$ .

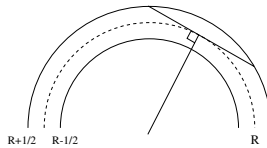
our calculus on the discrete bundle of parabolas that generates the arithmetical circles. This bundle of parabolas can be computed considering the intersection between the Euclidean circles  $x^2 + y^2 = (r - 1/2)^2$  and the grid, this leads to the bundle:

$$y = \left\lfloor \sqrt{r(1 + 2k) - k^2 + 1/4} \right\rfloor \quad \text{with } k \text{ a positive integer}$$

### 2.2 The Algorithm

We only use arithmetical circles due to their analogy to Euclidean rings: a discrete point belongs to an arithmetical circle if and only if it belongs to the Euclidean ring of radii  $r + 1/2$  and  $r - 1/2$ . Furthermore, the discrete tangent can be viewed as a digitization of the longest tangent of the circle of radius  $r$  that lies in the ring (see figure 3).

In the continuous space, it’s clear that the length of the inner tangent uniquely characterize the radius of the ring. In the discrete space, since we are



**Fig. 3.** Continuous ring and the inner tangent.

not able to recognize discrete circles, we have to make a link between discrete circles and well known discrete objects such as discrete straight lines. At this point of the discussion, we make the assumption that the length of the discrete tangent given by the Vialard’s algorithm is constant at each point of an arithmetical circle. In concrete case, we have a small variation of the tangent

length on a discrete circle but we will see that this variation does not interfere the results in the curvature calculus. Based on this assumption, we can propose a discrete osculating circle estimation: given a discrete curve, let  $l$  be the half length of the discrete tangent at a given point of the curve. Since the length of the discrete tangent on an arithmetical circle is constant, we can link the half discrete tangent to the first vertical patch given by the discrete parabolas associated to the arithmetical circles. Hence, by inverting the discrete equation of the first discrete parabola linked to arithmetical circle, we can compute the set  $\mathcal{S}_l$  of radii of discrete circles which have a first vertical patch of half length  $l$ :

$$\begin{aligned} \mathcal{S}_l &= \{r_{inf}, \dots, r_{sup}\} \\ r_{inf} &= \lceil (l - 1/2)^2 - 1/4 \rceil \\ r_{sup} &= \lfloor (l + 1/2)^2 - 1/4 \rfloor \end{aligned}$$

Now, given this set of possible discrete circles, we can compute the possible curvatures by inverting the radii. In practise, we return as an estimation of curvature the invert of the mean radius of this set:

$$k_h(\mathcal{S}_l) = \frac{2}{r_{inf} + r_{sup}}$$

Since the discrete tangent can be computed in an optimal time, we present, in algorithm 1, an optimal in time algorithm for the curvature computation of a discrete curve.

---

**Algorithm 1** Curvature Estimation

---

```

COMPUTE-CURVATURE(CURVE  $c$ )
1: for all pixel  $p$  in  $c$  do
2:   Compute the tangent  $\tau$  in  $p$ 
3:   Estimate the radii  $r_{sup}$  and  $r_{inf}$  using  $l = length(\tau)/2$ 
4:   return  $(\frac{2}{r_{inf} + r_{sup}})$ 
5: end for
    
```

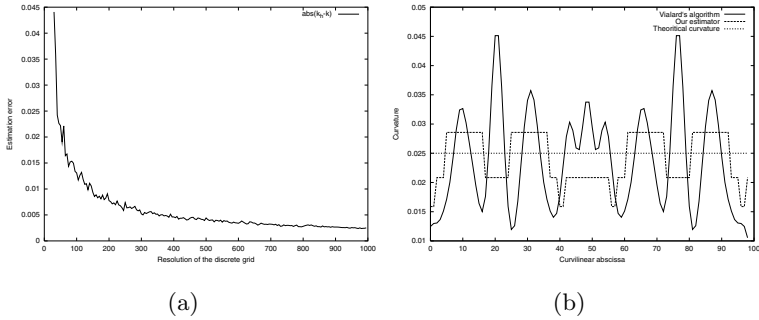
---

**2.3 Discussion**

In this section, we will discuss about the link between the grid of resolution  $h$  and our curvature calculus. Let us denote by  $k$  the curvature at the point  $x(s)$  of  $\partial\mathcal{X}$ . We consider a digitization of  $\partial\mathcal{X}$  on a grid of step  $h$  denoted by  $dX_h$ .

Our algorithm estimates the curvature  $\tilde{k}_h$  at the discrete point associated to  $s$ . A classical way in discrete geometry to justify a new discrete definition of an Euclidean tool is to prove the asymptotic convergence of the measure. Thus, if  $\tilde{k}_h$  is the estimation on a grid of size  $h$  and  $k$  the expected Euclidean curvature, we must proof the following equality :

$$\lim_{h \rightarrow 0} \frac{\tilde{k}_h}{h} = k$$



**Fig. 4.** Accuracy of the estimator: (a) Error in the curvature estimation ( $|1/r_h - k_h|$ ) when the digitization grid is re ned (b) comparison between the Vialard’s algorithm with  $\sigma = 1$  and our estimator on a discrete circle of radius 40.

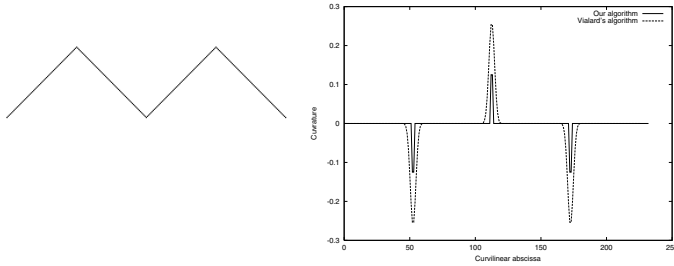
This equality means that the error of the estimation converges to 0 when we refine the discrete grid.

Since there is no formal proof of the convergence of the discrete tangent, we can only show an experimentally asymptotic convergence considering Euclidean disks and their discretization. Let  $R$  denotes the radius of an Euclidean disk  $\mathcal{X}$ . The radius of the discrete approximating circle of  $\partial\mathcal{X}$  is given by  $r_h = \lceil R/h \rceil$ . Decreasing  $h$  to 0, the estimated curvature  $\tilde{k}_h$  should converge to  $1/r_h$ . Instead of making  $h$  decreasing, we can compute the mean curvature of increasing discrete circles and check if the estimation error  $|\tilde{k}_h - 1/r_h|$  converges to 0. This experimental result can be found on figure 4-a.

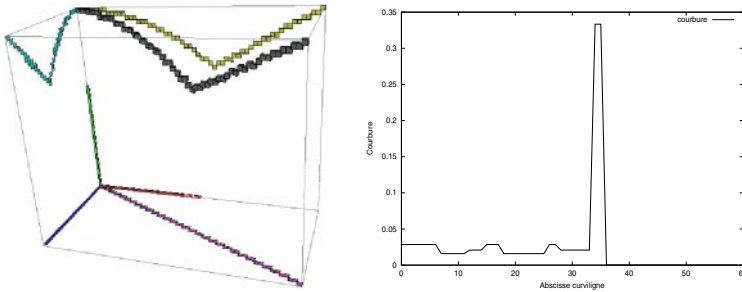
## 2.4 Results

First of all, we have to check the accuracy of our estimation for a given step grid  $h$ . Hence, we compare our algorithm to the Vialard’s one on a discrete circle (see figure 4-b). The results are quite similar but note that in our case, there is no smoothing process and our computation is purely discrete. We have also tested the accuracy in sense of localization of high curvature points (see figure 5).

Since Figueiredo and Réveillès [FR95] have proposed an arithmetical definition of 3D discrete lines based on the two dimensional ones: a 3D set of point is a 3D discrete straight line if and only if two of its three canonical projections on the grid planes are 2D discrete straight lines. Based on this definition Debled [DR95] has proposed an optimal time recognition algorithm of 3D lines. In the same way we can define a 3D discrete tangent using the recognition algorithm and thus we can use our algorithm to compute in an optimal time the curvature of a 3D discrete curve (see figure 6).



**Fig. 5.** Curvature of an unfolded square.



**Fig. 6.** An example of a 3D discrete curve (drak gray) composed of a circular arc and a straight lines, and its unsigned curvature graph.

## Conclusion

In this article, we have extended the Worring and Smeulders's [WS93] classification to new discrete algorithms and have presented a purely discrete and optimal curvature algorithm based on circle estimation that solves the arc fitting problem. We have shown the good results of this algorithm compared to the Vialard's algorithm which needs a smoothing parameter. Furthermore, we do not provide any filtering process to make our results stable. As future work, we will try to extend our method to 3D discrete surface curvature and thus give an useful tool for snow sample analysis [Pie99] or medical imaging [MLD94].

## References

- [And94] Eric Andres. *Cercles Discrets et Rotations discrettes*. PhD thesis, Universite Louis Pasteur - Strasbourg, 1994.
- [BD75] J.R. Bennett and J.S. Mac Donald. On the measurement of curvature in a quantized environment. *IEEE Transactions on Computers*, 24(8):803–820, August 1975.

- [Cha00] Raphaëlle Chaine. *Segmentation d'ensembles non organisés de points 3D d'une surface : propagation anisotropique d'étiquettes basée sur les graphes*. PhD thesis, Université Claude Bernard - Lyon 1, January 2000.
- [CM91] J-M. Chassery and A. Montanvert. *Géométrie discrète en analyse d'images*. Éditions Hermes, 1991.
- [DMT99] M. Delorme, J. Mazoyer, and L. Tougne. Discrete parabolas and circles on 2d cellular automata. *Theoretical Computer Science*, 218:347–417, 1999.
- [DR95] Isabelle Debled-Rennesson. *Etude et reconnaissance des droites et plans discrets*. PhD thesis, Université Louis Pasteur, 1995.
- [FR95] Oscar Figueiredo and Jean-Pierre Reveillès. A contribution to 3D digital lines. *Discrete Geometry for Computer Imagery*, 1995.
- [FST96] J. Françon, J.M. Schramm, and M. Tajine. Recognizing arithmetic straight lines and planes. *Discrete Geometry for Computer Imagery*, 1996.
- [FT99] Fabien Feschet and Laure Tougne. Optimal time computation of the tangent of a discrete curve : Application to the curvature. In *Discrete Geometry for Computer Imagery*, 1999.
- [HHTS98] H. Hagen, S. Heinz, M. Thesing, and T. Schreiber. Simulation based modeling. In *International Journal of Shape Modeling*, volume 4, pages 143–164, 1998.
- [Kov90] V.A. Kovalevsky. New definition and fast recognition of digital straight segments and arcs. *Proceedings of the tenth international conference on Pattern Analysis and Machine Intelligence*, June 1990.
- [KR96] Kong and Rosenfeld. Digital topology – A brief introduction and bibliography. In Kong and Rosenfeld, editors, *Topological Algorithms for Digital Image Processing (Machine Intelligence and Pattern Recognition, Volume 19)*, Elsevier. 1996.
- [MLD94] O. Monga, R. Lengagne, and R. Deriche. Crest lines in volume 3D medical images : a multi-scale approach. Technical report, INRIA, Projet SYNTIM, 1994.
- [Pie99] J.B. Brzoska B. Lesaffre C. Coléou K. Xu R.A. Pieritz. Computation of 3D curvature on a wet snow sample. *The European Physical Journal Applied Physics*, 1999.
- [Rev91] J.P. Reveilles. *Géométrie discrète, calcul en nombres entiers et algorithmique*. PhD thesis, Université Louis Pasteur - Strasbourg, 1991.
- [RJ73] A. Rosenfeld and E. Johnston. Angle detection on digital curves. *IEEE Transactions on Computers*, pages 875–878, September 1973.
- [TG92] J.P. Thirion and A. Gourdon. The 3D Marching Lines Algorithm and its Application to Crest Lines Extraction. Technical Report 1672, INRIA, April 1992.
- [Via96] Anne Vialard. Geometrical parameters extraction from discrete paths. *Discrete Geometry for Computer Imagery*, 1996.
- [WS93] Marcel Worring and Arnold W. M. Smeulders. Digital curvature estimation. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 58(3):366–382, November 1993.