Multigrid convergence of discrete geometric estimators

David Coeurjolly, Jacques-Olivier Lachaud, Tristan Roussillon

Multigrid Convergence of Discrete Geometric Estimators

David Coeurjolly, Jacques-Olivier Lachaud, and Tristan Roussillon

Abstract The analysis of digital shapes requires tools to determine accurately their geometric characteristics. Their boundary is by essence discrete and is seen by continuous geometry as a jagged continuous curve, either straight or not derivable. *Discrete geometric estimators* are specific tools designed to determine geometric information on such curves. We present here global geometric estimators of area, length, moments, as well as local geometric estimators of tangent and curvature. We further study their *multigrid convergence*, a fundamental property which guarantees that the estimation tends toward the exact one as the sampling resolution gets finer and finer. Known theorems on multigrid convergence are summarized. A representative subsets of estimators have been implemented in a common framework (the library `DGtal`), and have been experimentally evaluated for several classes of shapes. Thus, the interested users have all the information for choosing the best adapted estimators to their applications, and readily dispose of an efficient implementation.

13.1 Introduction

Since early developments in image processing and image understanding, many tools have been developed in order to quantify the geometry of a digital shape. Such digital shapes can be defined for instance either from a segmentation process as a subset of image pixels sharing the same colorimetric information, or as the result of the digitization of a continuous object.

D. Coeurjolly (✉) · T. Roussillon
LIRIS, UMR CNRS 5205, Université de Lyon, 69676 Villeurbanne, France
e-mail: david.coeurjolly@liris.cnrs.fr

T. Roussillon
e-mail: tristan.roussillon@liris.cnrs.fr

J.-O. Lachaud
LAMA, UMR CNRS 5127, University of Savoie, 73376 Le Bourget du Lac, France
e-mail: jacques-olivier.lachaud@univ-savoie.fr

In many applications, it is important to have a geometrical quantification or description from measurements which are invariant under a specific class of transforms (rotation, translation, and scaling) or which preserve important geometrical features (characteristic points, local convexity, etc.). In this context, we usually consider differential or integral quantities evaluated either on the digital shape or its boundary. Beside such type of quantification, we can distinguish two classes of geometrical descriptors. The first class corresponds to global descriptors which associate a global numerical quantity with each shape. In this class, we have arc length or perimeter estimators of digital shape boundaries, but also integral quantities such as geometrical moments approximated on the digital shape. The second class contains local estimators which usually associate a numerical quantity with each point of the shape. For example, curvature or normal vector estimators at boundary points belong to this class.

When defining an algorithm that evaluates such descriptors on digital shape, so called estimator, the evaluation of such estimator accuracy may be challenging. In the literature, several approaches have been proposed. The first one is application driven and consists in validating the estimators within a complete shape description pipeline. For instance, one can evaluate a curvature estimator in a global characteristic points estimation framework of contours.

One can also evaluate the accuracy of the estimator in terms of expected properties. For instance, we can evaluate the stability of a curvature estimator when rotations of input shapes are given.

A more formal evaluation process consists in comparing the estimated quantities with exact Euclidean values on a family of continuous shapes in a multigrid asymptotic framework. More precisely, let $\mathbb{X}$ be a family of compact simply connected subsets of $\mathbb{R}^2$ with continuous curvature fields. We denote by $D(X,h)$ the Gauss digitization of $X \in \mathbb{X}$ with grid step h, seen as a union of pixels of side h in $\mathbb{R}^2$. For sake of clarity, we shorten in the sequel $D(X,h)$ into D and denote its complementary by $\bar{D}$. Moreover, let us assume that D contains at least one pixel, i.e. $|D| \geq 1$.

In this multigrid framework, comparing the estimated quantity to the expected Euclidean one when h tends to zero is called the *multigrid convergence* analysis of an estimator [35]. Indeed, at a given resolution, infinitely many shapes have the same digitization, which hampers the objective comparison of estimators. For estimators of local geometric quantities like tangent or curvature, few results exist. We may quote some convergence results for tangent estimators [21, 42, 51]. And there is no correct convergence results for curvature as far as we know.

In this chapter, we use this multigrid comparison framework in order to review and evaluate existing local and global estimators on digital shapes. A important contribution is to have considered a large set of estimators in a unique technical framework: the DGtal open-source library [18]. DGtal is a generic open source library for Digital Geometry programming for which the main objective is to structure different developments from the digital geometry and topology community. For the purpose of this chapter, we use DGtal to represent multigrid digital objects and shapes, to define the geometric estimators and we provide ways to compare estimated values to expected Euclidean ones.

The chapter is organized as follows: Sect. 13.2 focuses on global estimators (area, moments and arc length) and Sects. 13.3, 13.4, and 13.5 are devoted to local estimators, tangent, and curvature, respectively. In all cases, each section starts with a formal definition of the multigrid convergence of an estimator. In Sect. 13.6, we discuss on implementation details of both the estimator and the comparative evaluation framework.

13.2 Global Estimators

13.2.1 Multigrid Convergence for Global Estimators

Multigrid convergence is an interesting way of relating digital and Euclidean geometries. The idea is to ask for discrete geometric estimations to converge toward the corresponding Euclidean quantity when considering finer and finer shape digitizations (here, Gauss digitization). The following definition is taken from Definition 2.10 of [35].

Definition 1 (*Multigrid convergence for global geometric quantities*) A discrete geometric estimator $\hat{E}$ of some geometric quantity E is *multigrid convergent* for a family of shapes $\mathbb{X}$ and a digitization process D iff for all shape $X \in \mathbb{X}$, there exists a grid step $h_X > 0$ such that the estimate $\hat{E}(D(X,h),h)$ is defined for all $0 < h < h_X$ and

$$\left|\hat{E}\big(D(X,h),h\big) - E(X)\right| \leq \tau_X(h),$$

where $\tau_X : \mathbb{R}^+ \to \mathbb{R}^+$ is with null limit at 0. This function is the *speed of convergence* of the estimator.

The convergence of most estimators depends on the family of shapes in the Euclidean plane that is considered. We therefore introduce several standard families that will be used to define the range of validity of multigrid convergence theorems. A curve is said to be C^n if it has continuous n-th order derivatives.

- The family of all finite convex shapes in the Euclidean plane is denoted with $\mathbb{X}^C$.
- The family of convex sets whose boundary is a C^n arc with positive curvature everywhere is denoted by $\mathbb{X}^{n\text{-}SC}$.
- The family of all planar piecewise n-smooth convex set is denoted with $\mathbb{X}^{n\text{-}PW\text{-}SC}$. These sets are convex sets whose boundary consists of a finite number of C^n arcs with positive curvature everywhere except at arc endpoints. Clearly $\mathbb{X}^{n\text{-}SC} \subsetneq \mathbb{X}^{n\text{-}PW\text{-}SC}$.

For experiments, we will use shapes that are representative of these families. Several representative shapes digitized at different scales are illustrated in Fig. 13.1. They will be used for the upcoming experiments on global and local geometric estimators.

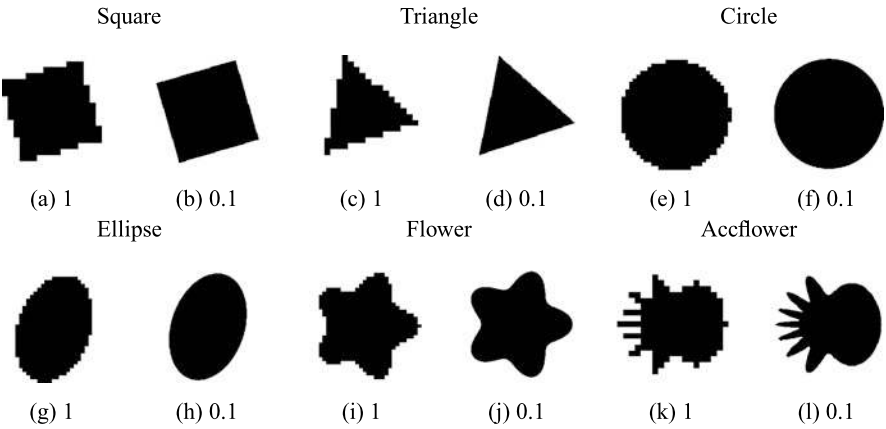


Fig. 13.1 Digitization at two different grid steps ($h = 1$ or $h = 0.1$) of tests shapes: (**a**–**d**) the *square* and *triangle* are in $\mathbb{X}^C$; the *circle* (**e**, **f**) and the *ellipse* (**g**, **h**) belongs to $\mathbb{X}^{3\text{-SC}}$; the *flower* (**i**, **j**), and the "*accflower*" (**k**, **l**) are in $\mathbb{X}^{3\text{-PW-SC}}$. All shapes have diameter 20

13.2.2 Area and Moments

Designing a multigrid convergent estimator of the area is fairly simple. We define the *area estimator by counting* $\hat{A}$ as

$$\hat{A}(Y,h) = h^2 \sum_{(i,j)\in Y} 1, \tag{13.1}$$

where Y is an arbitrary digital shape and h the grid step. This estimator just counts the number of h-grid square in Y and normalizes the result with the area of each grid square.

As reported in [36], Gauss and Dirichlet knew already that this area estimator was multigrid convergent for finite convex shape ($\mathbb{X}^C$) with a speed $O(l \cdot h)$, where l is the shape perimeter. Huxley [28] improves the bound to $O(h^{\frac{15}{11}}(\log \frac{1}{h})^{\frac{47}{22}})$ for the family $\mathbb{X}^{3\text{-}PW\text{-}SC}$. This simple estimator has thus super-linear convergence for a rather wide class of shapes.

Klette and Žunić [36] follow the idea of (13.1) to design the *discrete* (p,q)-*moment estimator* $\widehat{m_{p,q}}$, for integers $p, q \geq 0$, as follows:

$$\widehat{m_{p,q}}(Y,h) = h^{2+p+q} \sum_{(i,j)\in Y} i^p \cdot j^q. \tag{13.2}$$

These estimators approximate the (p,q)-moments of a shape X, which are defined as $m_{p,q}(X) = \iint_X x^p y^q dxdy$. Their speed of convergence is summed up in Table 13.1. In a similar way, central moments may be approximated. We refer the reader to [36] for further details. Note that moments can be used to determine for instance the center of gravity or the orientation of a shape. Furthermore, several rotational invariant quantities can be obtained as combination of (p,q)-moments. For

instance, Zernike and Legendre moments widely used in many 2D and 3D shape indexing and retrieval are linear combination of (p,q)-moments [56, 67]. Hence, convergence results on (p,q)-moments lead to convergence of Zernike and Legendre moments as well.

The previous estimators require to visit all points of the digital object, and not only its boundary. The computational complexity of these estimators is thus $O(1/h^2)$. However, a discrete variant of Green theorem allows to compute these quantities by simply visiting the shape boundary, thus reducing the computational complexity to $O(1/h)$ for convex shapes. See Lien [47] for a generic discrete Green theorem framework and Brlek et al. for a digital geometry application [3].

13.2.3 Perimeter and Length Estimators

It is more complex to estimate the perimeter of a digital shape. Indeed, enumerating the number of grid steps of the digital shape boundary does not lead to a reliable perimeter estimator. It is called the *naive perimeter estimator* $\hat{L}^{\text{naive}}$ and is defined as

$$\hat{L}^{\text{naive}}(Y,h) = h \sum_{\sigma \in \partial Y} 1. \tag{13.3}$$

This estimator overestimates the shape perimeter. Indeed, it is clear that it always returns the perimeter of the axis-aligned bounding box of the shape.

Therefore first approaches to length estimation tried to assign different weights to different local configurations so as to be more precise. The Rosen-Proffitt estimator [59] and BLUE (best linear unbiased) estimator [19] belong to this category. However it was proved in Tajine and Daurat [66] that all these approaches can never achieve multigrid convergence, whatever the (finite) number of configurations taken into account.

More complex approaches are required to achieve convergence. We list below several of them, which are also experimentally compared (see Fig. 13.2). Most of them are not only valid for perimeter estimation but also for curve length estimation.

- The *DSS length estimator* $\hat{L}^{\text{DSS}}$, proposed by Kovalevsky and Fuchs [37], relies on a greedy decomposition of the input digital contour into Digital Straight Segments (DSS). It starts from a point, then finds the longest DSS starting from that point. The end point of the DSS defines a new starting point. The process is repeated till the whole contour has been visited. The DSS end points form a polygonal line. The length or perimeter of the digital contour is then simply defined as Euclidean length of this polygonal line.
- The *MLP length estimator* $\hat{L}^{\text{MLP}}$, proposed by Sloboda et al. [65], also relies on a polygonal approximation of the digital contour. For a given simple digital shape, the Minimum Length Polygon (MLP) is indeed the shortest Euclidean curve which separates the interior pixel centers from the exterior pixel centers. The length is then defined as the perimeter of this curve. Several linear-time algorithms for computing the MLP are available [60, 63].

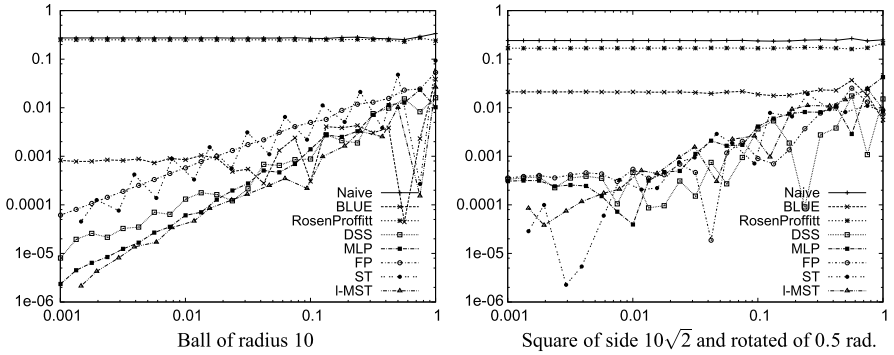


Fig. 13.2 Absolute relative error for several length and perimeter estimators. It is clear that the naive length estimator does not converge. The other estimators (DSS, MLP, FP, ST, λ-MST) present a multigrid convergence. Note that experimentally the convergence speed for DSS, FP, and ST on the ball is $O(h)$ while MLP and λ-MST achieve a better bound of $O(h^{\frac{4}{3}})$. However, on the boundary of a shape with linear parts, the convergence speed is $O(h)$ for all the estimators except the naive one

- The *FP length estimator* $\hat{L}^{\mathrm{FP}}$, proposed in [63], relies on yet another polygonal approximation of the digital contour. One can see it is a translated version of the MLP, where convex turns are translated outwards and concave turns are translated inwards by half-unit diagonal vectors. The advantage is that the polygon vertices form a subset of the grid points of the input contour.
- Another approach to local length estimation and thus perimeter estimation is to integrate the tangent estimation along the curve [8, 11] (see also the next section on tangent estimation). The *ST length estimator* $\hat{L}^{\mathrm{ST}}$ is based on the symmetric tangent while the *λ-MST length estimator* $\hat{L}^{\lambda\text{-MST}}$ is based on the λ-convex combination of maximal segments [42]. More precisely, if a grid edge σ has direction vector $\mathbf{t}(\sigma)$ and estimated unit tangent vector $\hat{\mathbf{T}}(\sigma)$, these two estimators are defined as:

$$\begin{aligned}\hat{L}^{\mathrm{ST}}(Y,h) &= h \sum_{\sigma\in\partial Y} \hat{\mathbf{T}}^{\mathrm{ST}}(\sigma)\cdot\mathbf{t}(\sigma),\\ \hat{L}^{\lambda\text{-MST}}(Y,h) &= h \sum_{\sigma\in\partial Y} \hat{\mathbf{T}}^{\lambda\text{-MST}}(\sigma)\cdot\mathbf{t}(\sigma).\end{aligned} \tag{13.4}$$

Some experimental evaluation of the multigrid convergence has been carried out for these estimators and is illustrated in Fig. 13.2. It appears that the perimeter of shapes with rectilinear boundaries is accurately estimated with any of the presented length estimators but for the naive one. However, for shapes with sufficiently smooth boundaries and positive curvature, the MLP and λ-MST have super-linear convergence and should be preferred. Note finally that only DSS, MLP, and λ-MST have proved multigrid convergence, but the found bounds are not necessarily tight. In Fig. 13.3, we present computational time for estimators implemented in DGtal release 0.4 (Naive, BLUE, RosenProffitt, DSS, MLP, FP). Convergence results for ST and λ-MST have been obtained from ImaGene library [29]. In these graphs, we can

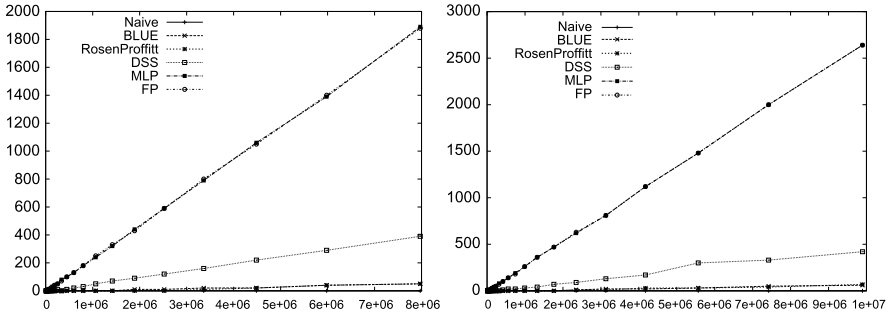


Fig. 13.3 Computation time for length estimators implemented in DGtal (Naive, BLUE, RosenProffitt, DSS, MLP, FP). For the sake of clarity, *abscissa* corresponds to the size of the contour in number of grid points. *Ordinate* corresponds to timings in millisecond (Intel Xeon 2.27 GHz, DGtal 0.4)

observe the linear computational cost of all estimators with respect to the size of the contours. As expected, local estimators outperform the other ones, but a DSS based estimator is a good compromise between efficiency and theoretical multigrid convergence.

13.2.4 Summary

Table 13.1 summarizes multigrid convergence results for estimators of global geometric quantities. It appears that some theoretical bounds are not tight and that some others are yet to be proved.

13.3 Local Estimators

13.3.1 Multigrid Convergence for Local Estimators

Tangent direction, normal vector, curvature are local geometric quantities along the shape boundary. Thus, each of them is a function of the shape boundary. However, the contour of the shape digitization does not define the same domain. Therefore we cannot directly compare the true geometric function with the estimated geometric function. We provide below a definition of multigrid convergence for discrete local estimators. It is neither a parametric definition as in [21] nor a point-wise definition as the standard multigrid convergence reported in [35]. Furthermore, for the sake of simplicity, there is no direct mapping between the contour and its digitized counterpart as proposed in [39]. It is a geometric definition, stating that any digital point sufficiently close to the point of interest has its estimated geometric quantity

Table 13.1 Known multigrid convergence for several estimators of global geometric quantities

Quantity	Estimator	Shape family	Convergence speed		References
			Upper bound	Observed	
area	$\hat{A}$	$\mathbb{X}^{C}$	$O(h)$		Gauss, Dirichlet
area	$\hat{A}$	$\mathbb{X}^{3\text{-}PW\text{-}SC}$	$O(h^{\frac{15}{11}+\epsilon})$		[28]
moments	$\widehat{m_{p,q}}$	$\mathbb{X}^{3\text{-}PW\text{-}SC}$	$O(h)$		[36]
moments	$\widehat{m_{p,q}}$	$\mathbb{X}^{3\text{-}SC}$	$O(h^{\frac{15}{11}+\epsilon})$		[36]
length	$\hat{L}^{\mathrm{DSS}}$	Convex polygons	$\approx 4.5h$		[37]
length	$\hat{L}^{\mathrm{DSS}}$	$\mathbb{X}^{3\text{-}PW\text{-}SC}$	(unknown)	$O(h)$	[37]
length	ϵ-sausage	Convex polygons	$\approx 5.844h$		[2]
length	$\hat{L}^{\mathrm{ST}}$	$\mathbb{X}^{C}$	(unknown)	$O(h)$	[11]
length	$\hat{L}^{\mathrm{FP}}$	$\mathbb{X}^{C}$	(unknown)	$O(h)$	[63]
length	$\hat{L}^{\mathrm{MLP}}$	$\mathbb{X}^{C}$	$\approx 8h$		[65]
length	$\hat{L}^{\mathrm{MLP}}$	$\mathbb{X}^{3\text{-}PW\text{-}SC}$	$O(h)$	$O(h^{\frac{4}{3}})$	[65]
length	$\hat{L}^{\lambda\text{-MST}}$	$\mathbb{X}^{3\text{-}PW\text{-}SC}$	$O(h^{\frac{1}{3}})$	$O(h^{\frac{4}{3}})$	[39]

which tends toward the expected local value of the geometric function. This definition of multigrid convergence imposes shapes with continuous geometric fields. Of course, one can afterward relax this constraint by splitting the shape boundary into individual parts where the geometric function is continuous.

Let us recall that $\mathbb{X}$ is some family of shapes in the Euclidean plane. We denote by $D(X,h)$ the Gauss digitization of $X \in \mathbb{X}$ with grid step h. For any x in the topological boundary ∂X of X, let $Q(X,x)$ be some local geometric quantity of ∂X at x. A *discrete local estimator* $\hat{Q}$ is a mapping which associates with any digital contour C, a point $y \in C$ and a grid step h, some value in a vector space (e.g., $\mathbb{R}$ for the curvature). We are now in position to define the multigrid-convergence of this estimator:

Definition 2 The estimator $\hat{Q}$ is *multigrid-convergent* for the family $\mathbb{X}$ if and only if, for any $X \in \mathbb{X}$, there exists a grid step $h_X > 0$ such that the estimate $\hat{Q}(D(X,h),y,h)$ is defined for all $y \in \partial D(X,h)$ with $0 < h < h_X$, and for any $x \in \partial X$,

$$\forall y \in \partial D(X,h) \text{ with } \|y-x\|_1 \leq h, \quad \left|\hat{Q}\big(D(X,h),y,h\big) - Q(X,x)\right| \leq \tau_{X,x}(h),$$

where $\tau_{X,x} : \mathbb{R}^{+*} \to \mathbb{R}^{+}$ has null limit at 0. This function defines the *speed of convergence* of $\hat{Q}$ toward Q at point x of ∂X. The convergence is *uniform* for X when every $\tau_{X,x}$ is bounded from above by a function τ_X independent of $x \in \partial X$ with null limit at 0.

It is worthy to note that, for sufficiently regular shapes (par(r)-regular shapes [43]), there exists a grid step below which the boundary of the shape digitization has same topology as the shape boundary ([39], Theorem B.5). Furthermore, these two

boundaries are very close. Indeed, there exists a grid step below which for any $x \in X$ there is a $y \in \partial D(X,h)$ with $\|y-x\|_1 \leq h$ and conversely for any $y \in \partial D(X,h)$, there is a $x \in X$ with $\|y-x\|_1 \leq h$ [39, Lemma B.9].

Therefore the previous definition of multigrid convergence guarantees that the estimated local quantity converges toward the true local geometric quantity everywhere along the shape boundary.

13.3.2 Methodology for Experimental Evaluation

When multigrid convergence theorems have been established, we will reference them and indicate the known convergence rate. We nevertheless carry out an experimental evaluation of many different estimators for two reasons: (1) few convergence theorems exist for local estimators, and (2) practical error bounds at finite scale are also important for the end-user.

In the next sections, we apply the following methodology for analyzing estimators:

1. Test shapes. We use the shapes of Fig. 13.1 for the experiments. They are representative of the different shape families that we are studying. Indeed, shapes composed of linear parts, smooth parts and corners, arise naturally in image analysis. When the tangent field is not continuous (square, triangle), only the average error is significant.
2. Graphs of estimations with respect to ground truth. We display the graphs of the estimated values for different estimators (functions $\hat{Q}$) and the expected graph (function Q).
3. Error measures for decreasing h. We study the following measures:

$$\epsilon_{abs}(X,y,h) = \left|Q\big(X,x(y)\big) - \hat{Q}\big(D(X,h),y,h\big)\right| \tag{13.5}$$

or (when a vector)
$$\epsilon_{abs}(X,y,h) = \left|det\big(\mathbf{Q}\big(X,x(y)\big),\hat{\mathbf{Q}}\big(D(X,h),y,h\big)\big)\right| \tag{13.6}$$

$$\epsilon_{rel}(X,y,h) = \frac{\epsilon_{abs}(X,y,h)}{|Q(X,x(y))|} \tag{13.7}$$

$$\overline{\epsilon}_{abs}(X,h) = \frac{1}{\#D(X,h)} \sum_{y \in D(X,h)} \epsilon_{abs}(X,y,h) \tag{13.8}$$

$$\overline{\epsilon}_{rel}(X,h) = \frac{1}{\#D(X,h)} \sum_{y \in D(X,h)} \epsilon_{rel}(X,y,h) \tag{13.9}$$

   Here $x(\cdot)$ is a mapping associating to each digitized point a point on the shape boundary that is close enough ($\|y-x(y)\|_1 \leq h$). The same mapping is used for all estimators.
4. When known, computational complexities for computing estimators will be given. Otherwise, for fair comparisons, we only measure computation times for estimators implemented in the DGtal library (see Sect. 13.6).

This methodology allows us to evaluate experimentally the accuracy and multigrid convergence properties of discrete local geometric estimators. Section 13.4 studies tangent estimators and Sect. 13.5 studies curvature estimators. Their implementation in a common framework is discussed in Sect. 13.6.

13.4 Tangent

The aim of tangent estimators is to determine what is locally the shape boundary direction. For curves $\gamma(s)$ (at least C^1) defined as functions of a curvilinear abscissa s, the tangent vector is defined as $\frac{d\gamma}{ds}$, which is a unit vector. The tangent direction $\phi(s)$ is defined as the angle between this unit vector and the x-axis.

13.4.1 Tangent Estimators

Given a digital contour and a digital point, tangent estimators return a unit vector. For easier view, it is also possible to plot the angle of the tangent vector w.r.t. the x-axis. It is clear that the grid edge direction (see arrows in Fig. 13.9d for an illustration) is a very bad tangent estimator, since on any shape in $\mathbb{X}^{\text{1-SC}}$ it will have points with ϵ_{abs} or ϵ_{rel} close to $\frac{\pi}{2}$.

More complex approaches are necessary. Digital tangent estimators have been thoroughly compared in [41, 42]. They have been compared to continuous approaches in [15, 16]. We describe below some representative tangent estimators, which will be compared experimentally.

- A first natural approach is to use a local least-square fit of a polynomial [5, 46]. These techniques define a fixed window-size q. Around the point of interest they use $2q+1$ samples which are used to find the polynomial of given degree that best fit these data in the least-square sense. We focus here on low-degree polynomials. The *LR tangent estimator* $\hat{\mathbf{T}}^{LR\text{-}q}$ is the linear-regression with the window size q. The *ICIPF tangent estimator* $\hat{\mathbf{T}}^{ICIPF\text{-}q}$ is the implicit parabola fitting of window size q, made independently on each coordinate. They were found to be representative of that kind of methods [15, 16].
- A second approach is to see the digital contour as a discrete signal $(x[t], y[t])$ and to convolve it with a Gaussian derivative of a given kernel σ. This is very similar to the binomial convolution approach of [21, 23, 51]. Therefore, we choose the $H1\text{-}0GD$ *tangent estimator* $\hat{\mathbf{T}}^{H1\text{-}0GD}$ [16], which defines locally the window size as the longest maximal digital straight containing the point. A slight variant is proved to be multigrid convergent in $O(h^{\frac{1}{3}})$ for smooth shapes in $\mathbb{X}^{3\text{-}SC}$, while its experimental convergent rate is excellent [16] for smooth shapes.
- The *MCMS tangent estimator* $\hat{\mathbf{T}}^{\text{MCMS}}$ defines the tangent as the direction of the most-centered maximal digital straight segment containing the point of interest [22]. It is proved to be uniformly multigrid convergent in $O(h^{\frac{1}{3}})$ in [39, 42].

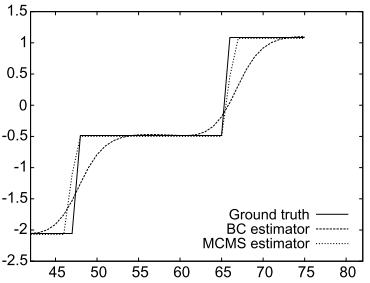
- The λ-*MST tangent estimator* $\hat{\mathbf{T}}^{\lambda\text{-MST}}$ is based on the λ-convex combination of the direction of the maximal digital straight segments containing the point of interest [42]. The function λ is a mapping governing the way these directions are combined. We use here a simple triangle function f, such that $f(0) = 0$, $f(1) = 0$, $f(0.5) = 1$. It is proved to be uniformly multigrid convergent in $O(h^{\frac{1}{3}})$ in [39, 42].
- The *BC tangent estimator* $\hat{\mathbf{T}}^{BC}$ considers the digital contour as a discrete signal $(x[t], y[t])$ and convolves it with discrete binomials and a discrete difference operator, so as to mimic the convolution by a Gaussian derivative [21, 51]. We use the suggested mask size of $d.h^{-\frac{4}{3}}$, where d is the continuous shape diameter. It is proved to be uniformly multigrid convergent in $O(h^{\frac{2}{3}})$ in [51].
- The *MATAS tangent estimator* $\hat{\mathbf{T}}^{MATAS}$ is an adaptation of the median filter commonly used in image processing [53]. If (P_i) are the vertices of the grid contour, this method consists in choosing the median orientation among the following $2q$ vectors centered on P_i: $(\mathbf{P}_{i-q}\mathbf{P}_i, \ldots, \mathbf{P}_{i-1}\mathbf{P}_i, \mathbf{P}_i\mathbf{P}_{i+1}, \ldots, \mathbf{P}_i\mathbf{P}_{i+q})$.

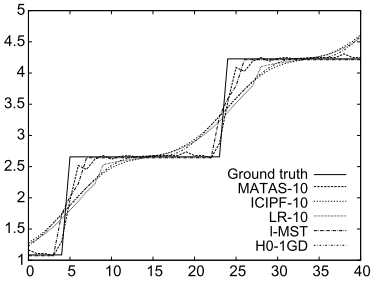
13.4.2 Experimental Evaluation

We have run these estimators on two representative shapes (the square is representative of $\mathbb{X}^C$, the ellipse is representative of $\mathbb{X}^{3\text{-SC}}$) at different steps (coarse $h = 1$, medium $h = 0.1$). Results are displayed in Fig. 13.4. Only MCMS and λ-MST detect perfectly straight parts and corners. Others tend to smooth around corners, the amount of smoothing being dependent on the (chosen) size of the window. Furthermore, LR and ICIPF oscillate around the correct value on straight parts. It is more difficult to tell which estimator is the best along the boundary of smooth curved parts. MCMS produces a staircase-like function but keeps the convexity of the shape. MATAS, LR and ICIPF may also oscillate and create false concavities. BC and λ-MST follow nicely the ground truth function. Overall, λ-MST seems to be the most versatile and accurate at these resolutions. Experiments on other shapes confirm the presented behaviors of these estimators.

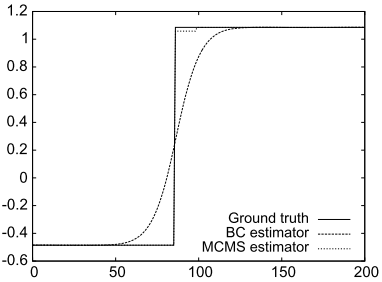
We now turn ourselves to the asymptotic behavior of these estimators, namely their possible multigrid convergence. We focus on the average absolute error of the tangent vector, i.e. $\overline{\epsilon}_{abs}(X, h)$ (see (13.8)). The error plots displayed in Fig. 13.5 show that tangent estimators with fixed window size are not multigrid convergent. This is the case of LR, ICIPF and MATAS estimators. Interestingly, but not surprisingly, small window sizes bring better precision at low scale while greater window sizes bring better precision at fine scale. This is clearly the problem of such estimators: they require a user to choose the best possible scale according to the input data.

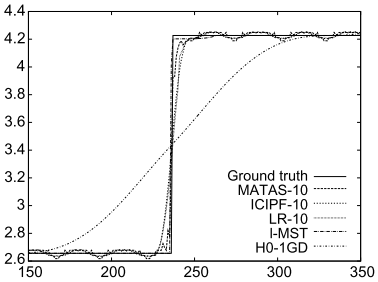
If we look at the other estimators (H1-0GD, MCMS, λ-MST, BC), the window size is automatically determined, either globally by $d.h^{-\frac{4}{3}}$ for BC estimator, or locally by maximal digital straight segments for the remaining three. All these four estimators are experimentally multigrid convergent for most of the considered shapes.



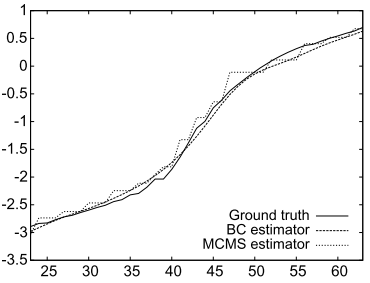


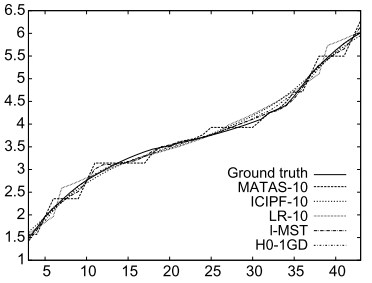
(a) The shape is the square digitized at $h = 1$.



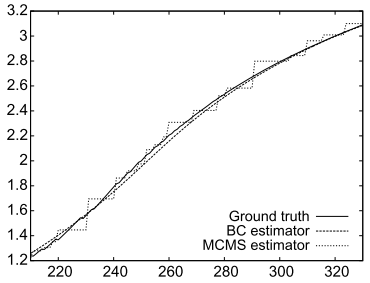


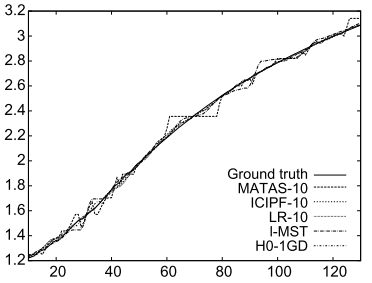
(b) The shape is the square digitized at $h = 0.1$.





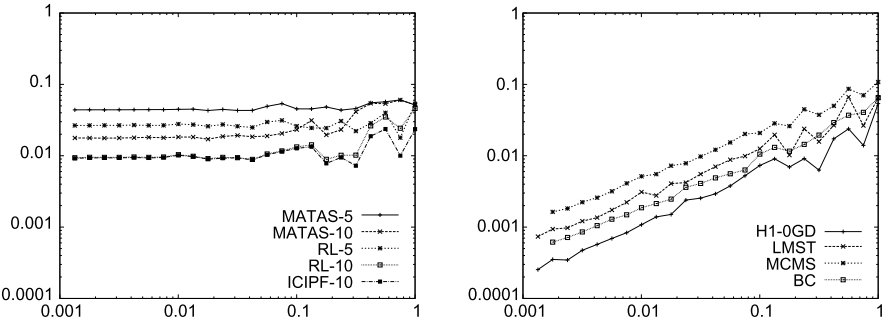
(c) The shape is the ellipse digitized at $h = 1$.



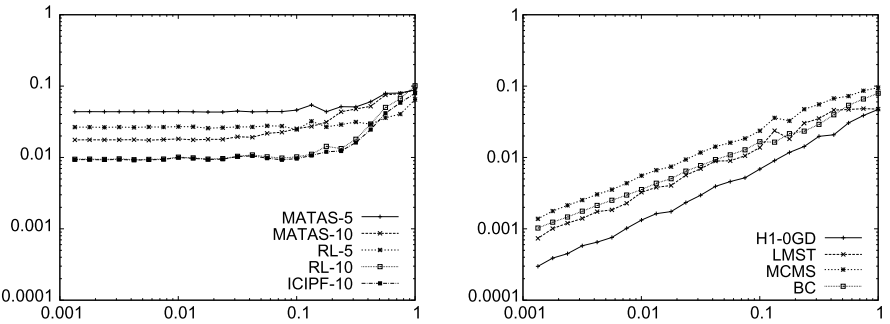


(d) The shape is the ellipse digitized at $h = 0.1$.

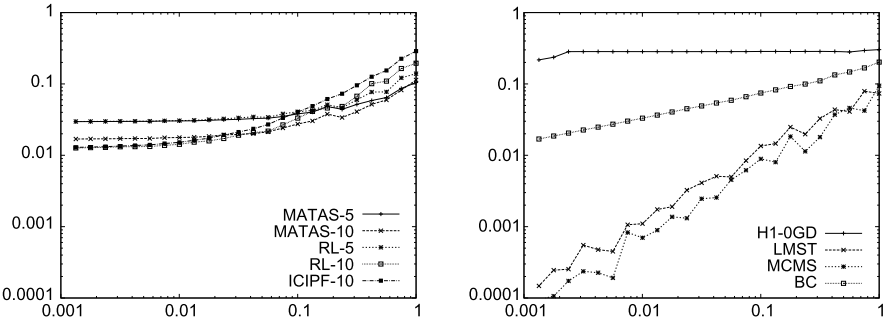
Fig. 13.4 Plots of tangent directions as a function of the grid edge index for several shapes and several tangent estimators. For each row, the shape and the digitization step is given. *Left column*: BC and MCMS estimators. *Right column*: MATAS estimator with window 10, ICIPF estimator with window 0, LR estimator with window 10, λ-MST estimator, H1-0GD estimator. Note that for a clearer view, only a representative part of the plot is displayed, and that due to implementation, grid edges indices of the *first column* are different from the ones of the *second column*



(a) The shape is the *circle* (in $\mathbb{X}^{\infty-SC}$) digitized for steps $h \in [0.001, 1]$.



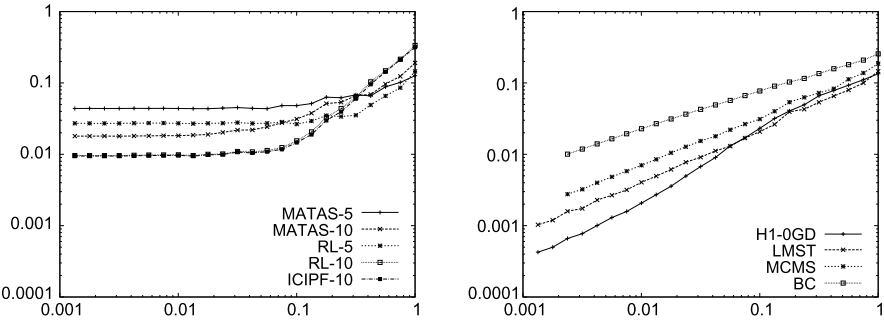
(b) The shape is the *ellipse* (in $\mathbb{X}^{\infty-SC}$) digitized for steps $h \in [0.001, 1]$.



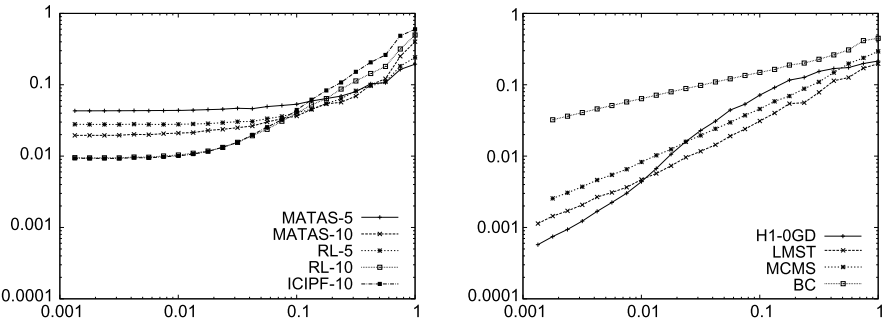
(c) The shape is the *triangle* (in $\mathbb{X}^{C}$) digitized for steps $h \in [0.001, 1]$.

Fig. 13.5 Plots in log-scale of the average absolute errors of tangent vectors as a function of the grid step for several shapes and several tangent estimators. For each row, the shape and the digitization step is given. *Left column*: MATAS estimator with windows 5 and 10, LR estimator with windows 5 and 10, ICIPF estimator with window 10. *Right column*: BC estimator, MCMS estimator, λ-MST estimator, H1-0GD estimator

However, their convergence speed may vary greatly. BC is good for smooth convex shapes, but has low convergence speed on shapes with inflexion points or linear parts. H1-0GD is excellent on smooth shapes for a fine enough sampling, but is not



(d) The shape is the *flower* (in $\mathbb{X}^{\infty-PW-SC}$) digitized for steps $h \in [0.001, 1]$.



(e) The shape is the *accflower* (in $\mathbb{X}^{\infty-PW-SC}$) digitized for steps $h \in [0.001, 1]$.

Fig. 13.5 (Continued)

Table 13.2 Known multigrid convergence for several tangent estimators. LR, MATAS, ICIPF are not multigrid convergent

Estimator	Shape family	Convergence speed		References
		Upper bound	Observed	
$\hat{\mathbf{T}}^{\text{BC}}$	Polygons	?	$O(h^{\frac{1}{3}})$	(here)
$\hat{\mathbf{T}}^{\text{BC}}$	$\mathbb{X}^{1\text{-}SC}$	$O(h^{\frac{2}{3}})$	$O(h^{\frac{2}{3}})$	[51]
$\hat{\mathbf{T}}^{\text{BC}}$	$\mathbb{X}^{1\text{-}PW\text{-}SC}$	?	$O(h^{\frac{1}{3}})$	(here)
$\hat{\mathbf{T}}^{\lambda\text{-MST}}$ and $\hat{\mathbf{T}}^{\text{MCMS}}$	Polygons	$O(h)$	$O(h)$	[39]
$\hat{\mathbf{T}}^{\lambda\text{-MST}}$ and $\hat{\mathbf{T}}^{\text{MCMS}}$	$\mathbb{X}^{1\text{-}PW\text{-}SC}$	?	$O(h^{\frac{2}{3}})$	[16]
$\hat{\mathbf{T}}^{\lambda\text{-MST}}$ and $\hat{\mathbf{T}}^{\text{MCMS}}$	$\mathbb{X}^{3\text{-}SC}$	$O(h^{\frac{1}{3}})$	$O(h^{\frac{2}{3}})$	[39]
$\hat{\mathbf{T}}^{\text{H1-0GD}}$	Polygons	?	not convergent	(here)
$\hat{\mathbf{T}}^{\text{H1-0GD}}$	$\mathbb{X}^{1\text{-}PW\text{-}SC}$	?	$\approx O(h^{\frac{2.5}{3}})$	[16]
$\hat{\mathbf{T}}^{\text{H1-0GD}}$	$\mathbb{X}^{3\text{-}SC}$	$O(h^{\frac{1}{3}})$	$O(h^{\frac{2.5}{3}})$	[16]

good on shapes with linear parts. MCMS and λ-MST are the most versatile. λ-MST is preferable to get a continuous tangent. Convergence results are summed up in Table 13.2.

13.5 Curvature

For curves $\gamma(s)$ (at least C^2) defined as functions of a curvilinear abscissa s, the curvature κ can be defined in three different, but essentially equivalent ways:

(i) norm of the second derivative: $\kappa(s) = |\frac{d^2\gamma}{ds^2}|$,
(ii) derivative of the tangent orientation: if $\phi(s)$ is the angle between the tangent and a given line, $\kappa(s) = \frac{d\phi}{ds}$,
(iii) inverse of the osculating circle radius: if $\rho(s)$ is the radius of the osculating circle, $\kappa(s) = 1/\rho(s)$.

Given a grid point of a digital contour, curvature estimators are expected to return a value in $\mathbb{R}$ close to the curvature of the underlying shape. Estimating the curvature by finite differences over the two neighbors of a given grid point returns either a positive (resp. negative) high value in convex (resp. concave) corners ($\pm 1/\sqrt{2}$) or a null one in runs and is thus a very bad solution.

Many curvature estimators have been proposed in the literature to cope with this problem. They are roughly based on one of the three above-mentioned definitions as it has been noticed in [27, 70].

In methods (i) and (ii), derivatives are often approximated from the convolution of either the tangent orientation [22, 68, 70] (i), or the digital contour viewed as a discrete signal $(x[t], y[t])$ [21, 23, 51] (ii). They can also be computed from some polynomials of a given degree locally fitted to the digital contour [27, 52].

Tangents and osculating circles used in methods (ii) and (iii) often rely on fitting techniques, either in a continuous setting (least square line or arc fitting [70]), or in a discrete setting to limit the arithmetic effects: digital straight segments [22, 68], digital level layers (extension to polynomials of higher degrees) [25, 61], approximation of the osculating circle with digital straight segments [12, 13, 27], digital circular arcs [62].

In most approaches, a user-given window or smoothing parameter is used so as to remove the jaggedness of digital contours and to make it continuous [21–24, 49, 51, 68, 70]. Few curvature estimators do not require an external parameter and we chose to focus on these methods.

13.5.1 Curvature Estimators

The curvature estimators that do not require any parameter either rely on discrete primitives such as digital straight segment (DSS), digital circular arc (DCA), or on global optimization such as the Global Minimum Curvature estimator [32].

In this section, we compare the following curvature estimators:

- The *MS estimator* $\hat{\kappa}^{\mathrm{MS}}$ [12] used only the length of maximal DSS to estimate the radius of the osculating circle.

  The method relies on the assumption that maximal DSS of the digitization of a Euclidean circle behave like chords of height h and length $\Theta(h^{\frac{1}{2}})$. Maximal

DSS are however almost always tighter and the length of maximal DSS has been proved to be in $O(h^{\frac{1}{2}})$ but in $\Theta(h^{\frac{2}{3}})$ on average [17].

- The *CC estimator* $\hat{\boldsymbol{\kappa}}^{\mathrm{CC}}$ [13] (HK2005 in [27]), associates with any grid point of a digital contour, the curvature of the circumscribed circle of a triangle defined by the extremities of its two digital half-tangents.

  It has been proved to be convergent if the length of maximal DSS is in $\Omega(h^{\frac{1}{2}})$ [8]. This condition is however not fulfilled because the length of maximal DSS has been proved later to be in $\Theta(h^{\frac{2}{3}})$ on average [17].
- Another estimation of the osculating circles can be obtained from the maximal DCA along the digital contour [62]. The *MDCA estimator* $\hat{\boldsymbol{\kappa}}^{\mathrm{MDCA}}$ is the piecewise constant function that associates with any grid point of a digital contour the curvature value of the most centered maximal DCA.

  Although this approach seems quite natural, it has been proposed only recently due to the lack of available implementation of on-line DCA recognition algorithms [10, 38, 64]. It is a natural extension of the tangent estimator based on the most centered maximal DSS (*MCMS tangent estimator* in Sect. 13.4) to the osculating circle estimation problem. As a result, the λ-*MST tangent estimator* used to improve this tangent estimator may probably improve this curvature estimator.

  The *MDCA estimator* has been proved to be convergent [62] if the length of the maximal DCA along the digital contour of the digitization of strictly convex shapes with continuous curvature field is in $\Omega(h^{\frac{1}{2}})$, which is observed on average.
- The *GMC curvature estimator* $\hat{\boldsymbol{\kappa}}^{\mathrm{GMC}}$ [32] computes the curvature of the shape that minimizes its squared curvature among all the Euclidean shapes that may be digitized as a digital set close to the input.

  The first step consists in computing the whole set of maximal DSS. This processing provides tangent and arc length estimations (Sect. 13.2.3 and Sect. 13.4) used to bound the set of valid shapes in the tangential space $(\phi(s), s)$. In this tangential space, the polygonal line that minimizes its length is then computed to approximate the shape of piecewise constant curvature that minimizes its squared curvature.

  The minimization is performed by an iterative numerical technique that stops when the difference between the squared curvature of the last two solution shapes is less than a small quantity, set to 1.10^{-8} in what follows.
- Finally, for comparisons, we also introduce the *BC curvature estimator* $\hat{\boldsymbol{\kappa}}^{\mathrm{BC}}$ [21, 51], which is computed from derivative estimations, obtained through a discrete difference operator applied on the digital contour viewed as a discrete signal $(x[t], y[t])$ convolved by a binomial kernel of a given size. The mask size is an input parameter that is not easy to determine, but following [51], it has been set to $d.h^{-\frac{4}{3}}$ where d is the diameter of the continuous shape.

  The multigrid convergence of the estimation of the first (resp. second) derivative at rate $O(h^{\frac{2}{3}})$ (resp. $O(h^{\frac{4}{9}})$) has been proved in [21, 51].

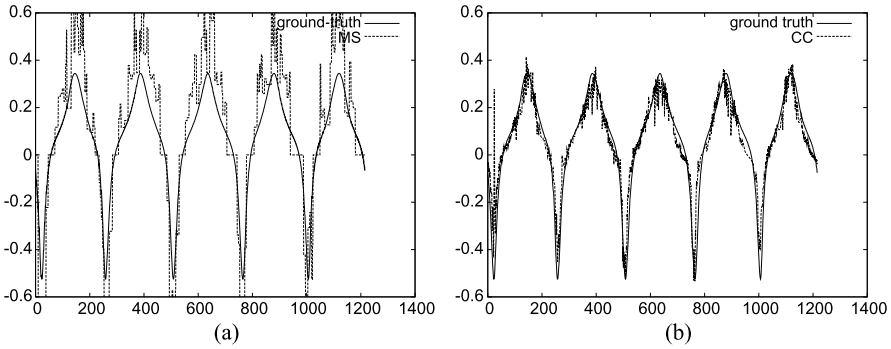


Fig. 13.6 Curvature plots for the flower, digitized with a grid step equal to 0.1 (Fig. 13.1j). *MS estimator* in (**a**) and *CC estimator* in (**b**)

13.5.2 Experimental Evaluation

We first plot the curvature values provided by the *MS estimator* (resp. *CC estimator*) in Fig. 13.6a (resp. Fig. 13.6b) when applied to the digital contour of Fig. 13.1j.

Because a maximal DSS is a good neighborhood to check the local convexity and concavity of a digital curve [63], the *MS estimator* provides positive curvature values in convex parts, negative curvature values in concave parts and null curvature values around inflection points (Fig. 13.6a). However, the *MS estimator* systematically over-estimates the true curvature values in the convex parts and underestimates the true curvature values in the concave parts. The deviation is important at low resolution and increases as the grid step h decreases. This is clearly a bad (and not convergent) estimator.

The *CC estimator* does not respect the convex and concave parts of the digital contour (see the peak of positive curvature value near the starting point in Fig. 13.6b), it oscillates a lot but gives correct results on average at low resolution.

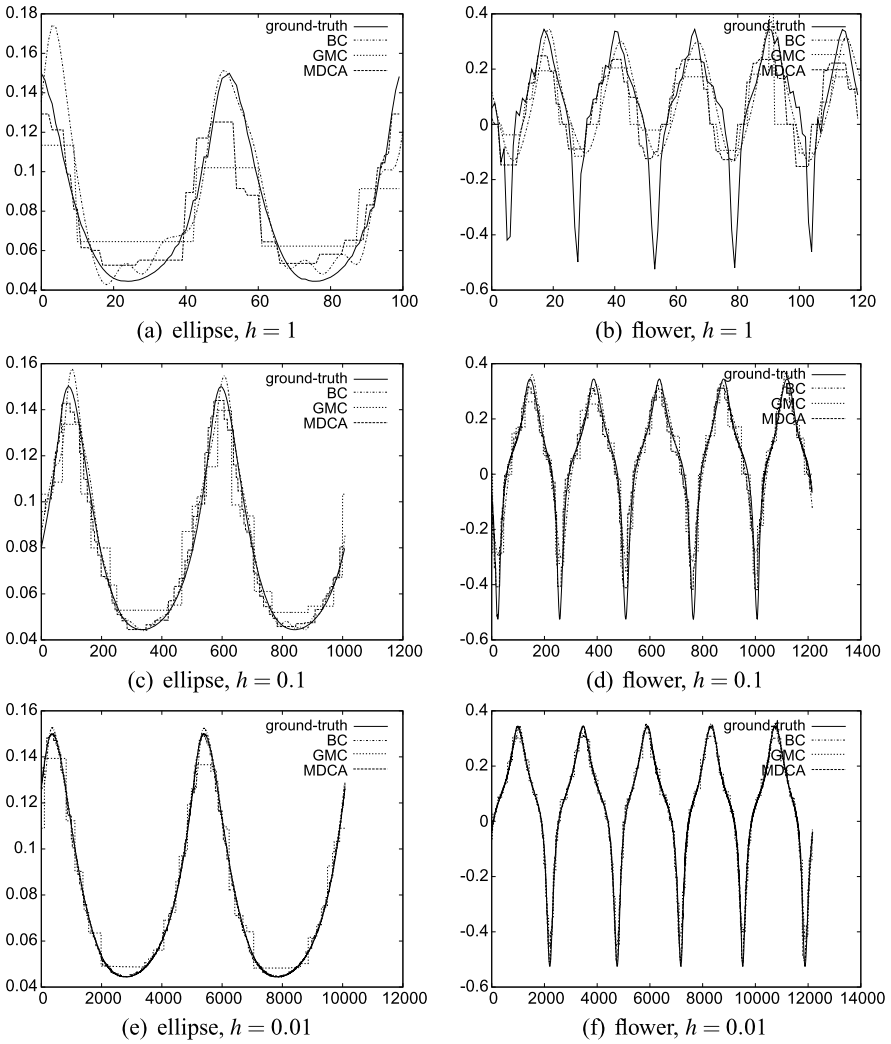
In Fig. 13.7, we compare the curvature plots derived from the *MDCA, GMC, BC estimators* to the ground-truth.

The visual deviation between the estimated graphs and the ground-truth graph reflects the average absolute error. For either estimator, the curvature estimations are more accurate for the ellipse than for the flower. For either shape, the curvature values obtained from any estimator get closer to the ground-truth (Fig. 13.7) and the absolute error decreases as the grid step h decreases.

For the ellipse and the flower, at grid step $h = 0.01$, the *MDCA estimator* and the *BC estimator* are better than the *GMC estimator*. In Fig. 13.7, their graphs are hardly confounded with the ground-truth graph.

In Fig. 13.8, the average absolute error has been plotted against the grid step h. The *CC estimator* is not convergent and has the highest errors. However, the other estimators ($\hat{\boldsymbol{\kappa}}^{\mathrm{MDCA}}$, $\hat{\boldsymbol{\kappa}}^{\mathrm{GMC}}$, $\hat{\boldsymbol{\kappa}}^{\mathrm{BC}}$) appear to be multigrid convergent.

We experimentally observed that the *MDCA estimator* has low absolute errors that decrease as the grid step h decreases. The convergence speed on average of the



(a) ellipse, $h = 1$ (b) flower, $h = 1$

(c) ellipse, $h = 0.1$ (d) flower, $h = 0.1$

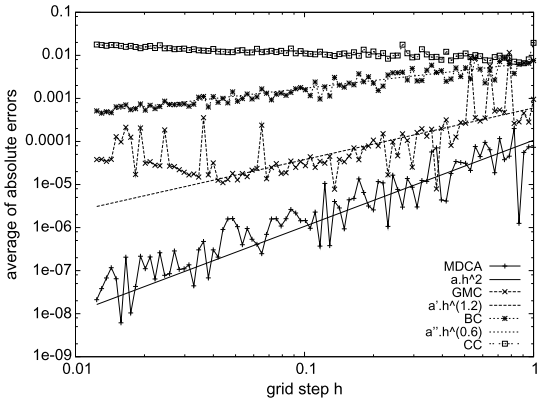
(e) ellipse, $h = 0.01$ (f) flower, $h = 0.01$

Fig. 13.7 Curvature plots for two shapes digitized at three different resolutions, computed from *MDCA, GMC, BC estimators*

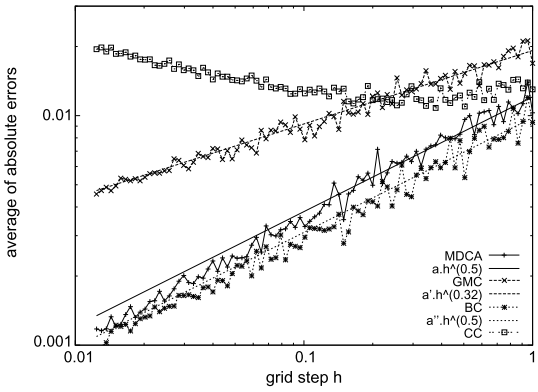
MDCA estimator is $O(h^{0.5})$ (even maybe $O(h^{\alpha})$ with $\alpha > 0.5$) for the ellipse and the flower (Fig. 13.8b and c) but $O(h^2)$ for the circle.

The *GMC estimator* and the *BC estimator* have usually higher errors. The *BC estimator* has lower errors than the *MDCA estimator* for the ellipse and for the flower at low resolution (when the grid step is decreasing from 1 to 0.3). The *GMC estimator* has however always higher errors than the *MDCA estimator*.

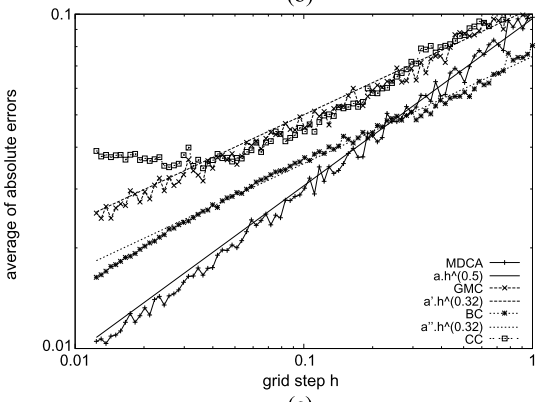
The *GMC estimator* and the *BC estimator* have usually a slower convergence speed:



(a)



(b)



(c)

Fig. 13.8 The average absolute error has been plotted against the grid step h for the digitization of a circle in (**a**), an ellipse in (**b**) and a flower in (**c**)

- respectively $O(h^{1.2})$ and $O(h^{0.6})$ for the circle (note that the *GMC estimator* is sensitive to the stop criterion of its optimization process when errors are small),
- respectively $O(h^{0.32})$ and $O(h^{0.5})$ for the ellipse,

- $O(h^{0.32})$ for the flower (but note that the error graph of the *BC estimator* is not straight and further experiments should be done at smaller grid steps to get the convergence speed).

Eventually the *MDCA, GMC, BC estimators* appear to be experimentally multigrid-convergent, but there is no correct theoretical convergence results for curvature estimation as far as we know, contrary to the case of tangent estimation (Sect. 13.4).

13.6 Implementation

In this section, we discuss about implementation details of both the geometrical estimators presented in the previous sections, and the experimental evaluation framework. All the estimators described in this chapter have been implemented in DGtal [18]. DGtal is an open-source C++ library focusing on the implementation of digital geometry objects and concepts. For short, it allows to represent images and objects in n-dimensional digital spaces equipped with both geometrical and topological tools.

In the context of this chapter, we will only consider the representation and the analysis of shape in dimension 2. As discussed in the introduction, the input digital object can be obtained either from an explicit description, from a segmentation process of an image (iso-level, ...), or as the digitization $D(X,h)$ of a continuous shape $X \in \mathbb{X}$. For the first two cases, DGtal provides mechanisms to construct such digital sets either explicitly or from a contour tracking process. For the last case, DGtal implements various implicit and parametric continuous shapes for which some global and local geometrical quantities are known. All such shape implementations are model of a concept of `CEuclideanShapes`[1] (see Fig. 13.1 for an illustration of DGtal Euclidean shapes). A digital object is thus obtained from a `GaussDigitizer` parametrized by a model of `CEuclideanShapes` and a grid step h. `CEuclideanShapes` models will be crucial for multigrid convergence analysis.

As discussed above and whatever the way the input digital object is specified, we need to access to its geometrical information in various ways:

- As a sequence of grid points, subset of $\mathbb{Z}^2$, e.g. for area and moment descriptors;
- As a representation of its boundary, e.g. for tangent or curvature estimators.

In the latter case, several options exist to define and represent a shape contour. Most of the options depend on the underlying topological model (Kong's like digital topology or cellular topology). Furthermore, depending on the algorithm used to perform the analysis, one may prefer a sequence of chain codes, a sequence

[1] DGtal uses a generic programming paradigm based on concepts and models of concepts. If a structure name starts from a capital "C", we describe a concept.

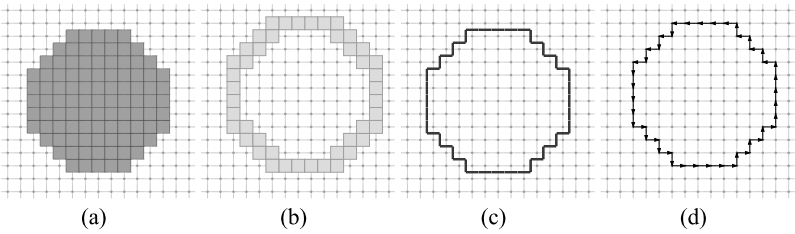


Fig. 13.9 Different representations of a Euclidean shape digitization: as a set of pixels (**a**), as a sequence of 4-connected pixels (**b**), as a sequence of 1-cell or *linels* (**c**), as a sequence of grid point displacements (**d**)

of linels or a sequence of 4-connected grid points to describe the contour (cf. Fig. 13.9).

To obtain a generic and extensible implementation of contour based estimators, we have defined a `GridCurve` structure constructed upon a topological cellular model which aims to provide several facets of a shape contour. More precisely, given the result of the contour tracking process, it provides mechanism (`Ranges` and `Iterators` on `Ranges`) to process the boundary sequence either as a set of grid points or a set of linels. Hence, a local geometrical estimator on contour, or more precisely a model of `CLocalGeometricalEstimator`, have an interface containing at least the two following methods:

- `void init(double h, ConstIterator & begin, ConstIterator & end,...)`: initialize the geometrical estimator with grid step h on a contour defined between iterators `begin` and `end`;
- `Quantity eval( ConstIterator & it)`: evaluate the estimator at the position `it` of the contour and return a `Quantity`.

In our framework, the type `ConstIterator` is a template parameter chosen in the contour iterator types provided in `GridCurve`.

Similarly, we have a concept of `CGlobalGeometricalEstimator` and models of this concept have an `eval()` method which returns a unique quantity for a shape (or subset of it).

Based on models of `CEuclideanShapes`, we can obtain expected continuous values using `TrueLocalEstimatorOnPoints` and `TrueGlobalEstimatorOnPoints`. Since both expected and estimated values are given by estimators with a consistent interface, it makes the multigrid comparison very simple. Indeed, it allows to design a generic `CompareLocalEstimators` which return a statistic on the difference of two estimator values.

In the following example, we illustrate the multigrid Euclidean shape construction and the comparison of three length estimators (RosenProffitt, DSS and MLP). In this example, we have detailed the overall process: shape construction and digitization, domain and Khalimsky space construction, contour tracking and finally, evaluation of estimators.

```
//....
// h and radius are parameters here
//....
// Types
typedef Ball2D<Space> Shape;
typedef Space::Point Point;
typedef Space::RealPoint RealPoint;
typedef Space::Integer Integer;
typedef HyperRectDomain<Space> Domain;
typedef KhalimskySpaceND<Space::dimension,Integer> KSpace;
typedef KSpace::SCell SCell;
typedef GridCurve<KSpace>::PointsRange PointsRange;
typedef GridCurve<KSpace>::ArrowsRange ArrowsRange;
typedef PointsRange::ConstIterator ConstIteratorOnPoints;

//Euclidean ball
Shape aShape(Point(0,0), radius);

//Gauss Digitization
GaussDigitizer<Space,Shape> dig;
dig.attach( aShape ); // attaches the shape.
dig.init( aShape.getLowerBound(), aShape.getUpperbound(),h);

// The domain size is given by the digitizer according to
// the window and the step.
Domain domain = dig.getDomain();

// Create cellular space
KSpace K;

bool ok = K.init( dig.getLowerBound(), dig.getUpperBound(), true );
if ( ! ok ) {
  std::cerr << " "
            << " error in creating KSpace." << std::endl;
  return false;
}
try {
  // Extracts shape boundary
  SurfelAdjacency<KSpace::dimension> SAdj( true );
  SCell bel = Surfaces<KSpace>::findABel( K, dig, 10000 );

  // Getting the consecutive surfels of the 2D boundary
  std::vector<Point> points;
  Surfaces<KSpace>::track2DBoundaryPoints( points,
                                           K, SAdj,
                                           dig, bel );
  trace.info() << "# tracking..." << endl;

  // Create GridCurve
  GridCurve<KSpace> gridcurve;
  gridcurve.initFromVector( points );
  trace.info() << "#grid curve created, h=" << h << endl;

  //ranges
  ArrowsRange ra = gridcurve.getArrowsRange();
  PointsRange rp = gridcurve.getPointsRange();

  //Three length estimators working on different contour
  //representations

  RosenProffittLocalLengthEstimator< ArrowsRange::ConstIterator >
       RosenProffittlength;
  RosenProffittlength.init(h, ra.begin(), ra.end(), gridcurve.isClosed());

  DSSLengthEstimator< PointsRange::ConstIterator > DSSlength;
  DSSlength.init(h, rp.begin(), rp.end(), gridcurve.isClosed());
```

```
  MLPLengthEstimator< PointsRange::ConstIterator > MLPlength;
  MLPlength.init(h, rp.begin(), rp.end(), gridcurve.isClosed());

  trace.info() << "#Estimations" <<std::endl;
  trace.info() << "#h␣true␣RosenProffitt␣DSS␣MLP␣" <<std::endl;
  trace.info() << h << "␣" << M_PI*2.0
               << "␣" << RosenProffittlength.eval()
               << "␣" << DSSlength.eval()
               << "␣" << MLPlength.eval()
               << std::endl;
} catch ( InputException e ) {
  std::cerr << "␣"
            << "␣error␣in␣finding␣a␣bel." << std::endl;
  return false;
}
```

13.7 Related Problems and Perspectives

13.7.1 Geometric Estimators Along Damaged or Noisy Contours

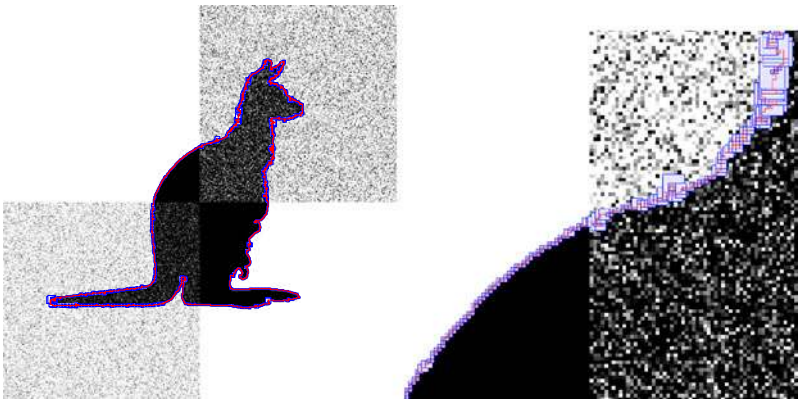
In real applications, images may have been damaged or acquisition devices may induce noise in the image data. Furthermore, binarization algorithms and segmentation algorithms may also damage the boundary of the regions or shapes. These contours are thus not anymore the perfect digitization of "nice" Euclidean shapes (e.g. shapes in $\mathbb{X}^{n\text{-}PW\text{-}SC}$), and have parts that are winding where they should be straight. We will call them hereafter *noisy contours* (an example is given in Fig. 13.10 where a kangaroo shape has been damaged by Gaussian noise).

In the pattern recognition community, a lot of tools have been developed to analyze the geometry of noisy contours, especially to detect corner or dominant points (see for instance [52]). These points are related to curvature information. However, these tools are not designed for estimating quantitatively the geometric characteristics of the contours but rather qualitatively. We only quote here works that give quantitative geometric information on perfect or noisy contours.

A common way to tackle noise along contours is to filter the contour with a smoothing kernel. The size of the kernel given by the user is more or less proportional to the amount of damage along the contour. The BC tangent estimator and the BC curvature estimator are members of this family of techniques [21, 23, 51]. These techniques are efficient when the contour is rather uniformly damaged, but they smooth indifferently noise and high-curvature parts of the contour (like corners).

Approaches based on fitting like the LR or ICIPF estimators [5, 46, 70] are also able to tackle noise along contours, since they tend to find the median or average polynomial that best approaches locally the data. Again, the window size parameter is used to suppress at the same time arithmetic effects and noise artifacts. This parameter is generally set by the user.

In the digital geometry community, a common technique is to use the so-called *blurred segments* [14] instead of digital straight segment. Compared to digital straight segment whose thickness is always less than 1, blurred segments have a



(a) Noise detection (b) Zoom on kangaroo's back

Fig. 13.10 Noisy contour and noise detection along it by the method of [30]. The input image has been damaged by two different Gaussian noises in different regions. The thresholded shape has a boundary which is damaged in these regions (noisy contour in *solid dark gray*). The detector indicates for each contour point what is the local scale at which this part of the contour should be analyzed: the scale or noise level at a point is indicated by the size of the *light gray box* around it. It is worthy to note that the noise level is almost everywhere proportional to the amount of contour degradation

user-given maximal thickness. The noisy parts of contours are thus ignored when using a larger thickness. Several estimators just replace standard segments with blurred segments so as to take into account noisy contours. For instance, the curvature estimator presented in [55] is the noisy variant of the CC estimator. The GMC estimator also uses blurred segments to handle noisy contours. The thickness is generally set by the user.

Note that digital estimators based on digital straight segments (like the H0-1GD, MCMS and λ-MST tangent estimators, or the MDCA curvature estimator) can also be adapted to noisy contours by sub-sampling the input contour. For instance, we can use a 3×3 tile over the input contour so as to remove perturbation no greater than 1 pixel along the contour. However we have yet not run a full set of experiments so as to know if this approach leads to better estimators than the ones quoted in the preceding paragraphs.

Finally, all these techniques require the determination of one or several parameters in order to process at best noisy contours. This scale or smoothing parameter must not be too low otherwise damaged parts are considered high-curvature places or corners, but it must not be too high also in order to preserve features and to have accurate estimates of geometric information.

If a gray-level input noisy image is available, scale space analysis may provide information on the amount of noise [7, 20, 26, 34]. They cannot handle directly noisy contours. For noisy contours, Kerautret and Lachaud [30] have recently proposed a method to automatically detect the meaningful scales of digital contours. It can give locally along the contour what is the amount of noise and the first scale at which the

contour should be analyzed (see Fig. 13.10, and on-line demonstration [33]). Their technique relies on the asymptotic properties of maximal digital straight segments. They have proposed a variant of λ-MST estimator for noisy contours, which uses the noise information given by the meaningful scales [31].

13.7.2 Geometric Estimators in 3D and nD

In higher dimensions, several 2D estimators or frameworks can be extended. However, many open problems exist and beside the fact that a few proofs of multigrid convergence exist, a complete experimental multigrid evaluation of curvature estimators for instance has not been done yet on digital surfaces in $\mathbb{Z}^3$. In this section, we just give a brief overview of existing techniques:

- *Surface area*: to compute the area of a surface in $\mathbb{Z}^3$, a first solution is based on weighted local configurations [48, 69]. The idea is to associate weights with local configurations of surface voxels or surfels. Then, given an object, the surface area is approximated by summing all weights associated with all configurations defined on the object surface. Similarly to the BLUE estimator, weights are given by a statistical analysis to minimize surface area error for a given class of shapes. By deriving results from the length case in dimension 2 [66], surface weighted configuration estimators can never achieve multigrid convergence. In [50], the authors use statistical analysis and integral geometry to design a fast estimation of the surface area. Again, the quality of the estimation is controlled by a parameter (number of line probes). Another option is to generalize the discrete normal vector integration scheme as described in [8, 11]. As detailed in [9], we can prove that if the normal vector estimation is multigrid convergent, then the integration of the vector field leading to the surface area estimation is multigrid convergent as well.
- *Normal vector field computation*: at a point x on a smooth surface, the normal vector at x can be defined as the cross product of first order derivatives at x (tangent) of two curves lying on the surface crossing at x. In a digital context, given a surface element of a cellular representation of a digital surface, two natural digital 4-connected curves can be defined by the intersection of the surface with the two coordinate planes containing the surfel elementary normal vector. Hence, Lenoir et al. suggested to compute the normal vector at a surfel as the cross product of tangent computed on the two 2D digital curves [45]. Following this framework, multigrid convergence can be achieved if the tangent estimator used on the 2D curves is multigrid convergent [9, 39]. The normal vector field of a digital surface in $\mathbb{Z}^n$, for arbitrary n, can be computed with a similar approach [40].
- *Curvature*: For curvature computation on digital surfaces, only few estimators have been proposed in the digital geometry framework. We can cite Lenoir's slice based approach for the mean curvature estimation [44], Gauss map area evaluation for the Gaussian curvature [8], techniques based on integral invariants for both mean and Gaussian curvatures [4, 57, 58]. Integral invariant techniques are definitely relevant in the digital geometry context, even in case of a noisy surface.

However, they require a window parameter which could be difficult to set for a large class of shapes.

In computational geometry, several techniques have been proposed to construct accurate curvature estimators with bounded errors. Usually, bounds are parametrized by a sampling parameter for a given sampling hypothesis. An example of a sampling hypothesis for a smooth surface would be that the sampling density should be proportional to the curvature. In this context, convergence or stability of geometric estimators have been proposed as a function of the sampling parameter. In many situations the sampling theorems used in computational geometry do not match with the specific isotropic behavior of digital surfaces. In [1, 6, 54], estimators are defined on point sets based on Voronoi structures and the error is given in terms of Hausdorff distance (which is consistent with digital surfaces). Investigating the links between computational geometry and digital geometry on this subject is a challenging problem.

13.7.3 Current Bottlenecks and Open Problems

As detailed in the previous sections, we can overview current theoretical bottlenecks in the design of discrete geometric estimators:

- *Stability w.r.t. noise*: Prior detection of the contour local noise level to be used as an estimator parameter, or with estimators which are theoretically robust to a given noise model.
- *Estimators of differential quantity of order 2*: From our point of view, existing curvature estimators are not yet satisfactory since either no proof of multigrid convergence exists, or the convergence is controlled by an external parameter (e.g., window size or Gaussian kernel width). It would be interesting, for instance, to focus on the multigrid behavior of circular arc segment on digital 2D contours. Indeed, many proofs related to the length or the tangent estimation are based on the multigrid behavior of DSS. On digital surfaces and in higher dimension, we think that a better understanding of links between computational and digital geometry results would lead to new results in this domain.

Beside these theoretical bottlenecks, complete experimental multigrid evaluations are now mandatory when designing a new discrete estimator. With the help of both a theoretical methodology (multigrid shape database and error measures) and open-source libraries (`ImaGene` [29] or `DGtal` [18]), we expect to have a complete and stable experimental framework. An important future work would be to continue the implementation of existing estimators with comparative studies. In dimension 3, main bottlenecks are related to efficiency and computational costs. Indeed, in many Material sciences or Medical imaging applications, we may have to analyze digital shapes whose size achieves up to 2048^3. In the implementation of 3D estimators, several theoretical and technical problems have to be addressed, such as out-of-core techniques, hierarchical data representation and adaptive algorithms, and others.

Acknowledgements This work was partially funded by project KIDICO (ANR-2010-BLAN-0205) of the French Research Agency (ANR).

References

1. Amenta, N., Kil, Y.: Defining point-set surfaces. In: ACM SIGGRAPH, vol. 23, p. 270. ACM, New York (2004)
2. Asano, T., Kawamura, Y., Klette, R., Obokata, K.: Minimum-length polygons in approximation sausages. In: 4th International Workshop on Visual Form. Lecture Notes in Computer Science, vol. 2059, pp. 103–112. Springer, Berlin (2001)
3. Brlek, S., Labelle, G., Lacasse, A.: The discrete green theorem and some applications in discrete geometry. Theor. Comput. Sci. **346**(2–3), 200–225 (2005)
4. Bullard, J.W., Garboczi, E.J., Carter, W.C., Fullet, E.R.: Numerical methods for computing interfacial mean curvature. Comput. Mater. Sci. **4**, 103–116 (1995)
5. Cazals, F., Pouget, M.: Estimating differential quantities using polynomial fitting of osculating jets. Comput. Aided Geom. Des. **22**, 121–146 (2005)
6. Chazal, F., Cohen-Steiner, D., Lieutier, A., Thibert, B.: Stability of Curvature Measures (2008)
7. Chen, K.: Adaptive smoothing via contextual and local discontinuities. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1552–1566 (2005)
8. Coeurjolly, D.: Algorithmique et géométrie pour la caractérisation des courbes et des surfaces. Ph.D. thesis, Université Lyon 2 (2002)
9. Coeurjolly, D., Flin, F., Teytaud, O., Tougne, L.: Multigrid convergence and surface area estimation. In: Theoretical Foundations of Computer Vision "Geometry, Morphology, and Computational Imaging". Lecture Notes in Computer Science, vol. 2616, pp. 101–119. Springer, Berlin (2003)
10. Coeurjolly, D., Gérard, Y., Reveillès, J.P., Tougne, L.: An elementary algorithm for digital arc segmentation. Discrete Appl. Math. **139**(1–3), 31–50 (2004)
11. Coeurjolly, D., Klette, R.: A comparative evaluation of length estimators of digital curves. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 252–258 (2004)
12. Coeurjolly, D., Miguet, S., Tougne, L.: Discrete curvature based on osculating circle estimation. In: 4th International Workshop on Visual Form. Lecture Notes in Computer Science, vol. 2059, pp. 303–312 (2001)
13. Coeurjolly, D., Svensson, S.: Estimation of curvature along curves with application to fibres in 3d images of paper. In: 13th Scandinavian Conference on Image Analysis. Lecture Notes in Computer Science, vol. 2749, pp. 247–254 (2003)
14. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear times. Comput. Graph. **30**, 30–36 (2006)
15. de Vieilleville, F., Lachaud, J.O.: Experimental comparison of continuous and discrete tangent estimators along digital curves. In: 12th International Workshop on Combinatorial Image Analysis. Lecture Notes in Computer Science, vol. 4958, pp. 26–37. Springer, Berlin (2008)
16. de Vieilleville, F., Lachaud, J.O.: Comparison and improvement of tangent estimators on digital curves. Pattern Recognit. **42**(8), 1693–1707 (2009)
17. de Vieilleville, F., Lachaud, J.O., Feschet, F.: Convex digital polygons, maximal digital straight segments and convergence of discrete geometric estimators. J. Math. Imaging Vis. **27**(2), 139–156 (2007)
18. DGtal: digital geometry tools and algorithms library. http://liris.cnrs.fr/dgtal
19. Dorst, L., Smeulders, A.W.M.: Length estimators for digitized contours. Comput. Vis. Graph. Image Process. **40**(3), 311–333 (1987)
20. Elder, J., Zucker, S.W.: Local scale control for edge detection and blur estimation. IEEE Trans. Pattern Anal. Mach. Intell. **20**(7), 669–716 (1998)

21. Esbelin, H.A., Malgouyres, R.: Convergence of binomial-based derivative estimation for c2-noisy discretized curves. In: 15th Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 5810, pp. 57–66 (2009)
22. Feschet, F., Tougne, L.: Optimal time computation of the tangent of a discrete curve: application to the curvature. In: Proc. 8th Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 1568, pp. 31–40 (1999)
23. Fiorio, C., Mercat, C., Rieux, F.: Curvature estimation for discrete curves based on auto-adaptive masks of convolution. In: Computational Modeling of Objects Presented in Images. Lecture Notes in Computer Science, vol. 6026, pp. 47–59 (2010)
24. Fleischmann, O., Wietzke, L., Sommer, G.: A novel curvature estimator for digital curves and images. In: 32th Annual Symposium of the German Association for Pattern Recognition. Lecture Notes in Computer Science, vol. 6376, pp. 442–451 (2010)
25. Gérard, Y., Provot, L., Feschet, F.: Introduction to digital level layers. In: 16th IAPR International Conference Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 6607, pp. 83–94. Springer, Berlin (2011)
26. Goshtasby, A., Satter, M.: An adaptive window mechanism for image smoothing. Comput. Vis. Image Underst. **111**, 155–169 (2008)
27. Hermann, S., Klette, R.: A comparative study on 2d curvature estimators. In: 17th International Conference on Computer Theory and Applications, pp. 584–589 (2007)
28. Huxley, M.N.: Exponential sums and lattice points. Proc. Lond. Math. Soc. **60**, 471–502 (1990)
29. ImaGene: generic digital image library. https://gforge.liris.cnrs.fr/projects/imagene
30. Kerautret, B., Lachaud, J.: Multiscale analysis of discrete contours for unsupervised noise detection. In: 13th International Workshop on Combinatorial Image Analysis. Lecture Notes in Computer Science, vol. 5852, pp. 187–200. Springer, Mexico (2009)
31. Kerautret, B., Lachaud, J.: Meaningful scales detection along digital contours for unsupervised local noise estimation. IEEE Trans. Pattern Anal. Mach. Intell. (submitted)
32. Kerautret, B., Lachaud, J.O.: Curvature estimation along noisy digital contours by approximate global optimization. Pattern Recognit. **42**(10), 2265–2278 (2009)
33. Kerautret, B., Lachaud, J.O.: Noise level and meaningful scale detection online demonstration. http://kerrecherche.iutsd.uhp-nancy.fr/MeaningfulBoxes (2010)
34. Kervrann, C.: An adaptive window approach for image smoothing and structures preserving. In: 8th European Conference on Computer Vision. Lecture Notes in Computer Science, vol. 3023, pp. 132–144. Springer, Berlin (2004)
35. Klette, R., Rosenfeld, A.: Digital Geometry—Geometric Methods for Digital Picture Analysis (2004)
36. Klette, R., Žunić, J.: Multigrid convergence of calculated features in image analysis. J. Math. Imaging Vis. **13**, 173–191 (2000)
37. Kovalevsky, V., Fuchs, S.: Theoretical and experimental analysis of the accuracy of perimeter estimates. In: Proc. Robust Computer Vision, pp. 218–242 (1992)
38. Kovalevsky, V.A.: New definition and fast recognition of digital straight segments and arcs. In: 10th International Conference on Pattern Analysis and Pattern Recognition, pp. 31–34 (1990)
39. Lachaud, J.O.: Espaces non-euclidiens et analyse d'image : modèles déformables riemanniens et discrets, topologie et géométrie discrète. Habilitation diriger des recherches, Université Bordeaux 1, Talence (2006) (in French)
40. Lachaud, J.O., Vialard, A.: Geometric measures on arbitrary dimensional digital surfaces. In: 11th International Conference Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 2886, pp. 434–443. Springer, Berlin (2003)
41. Lachaud, J.O., Vialard, A., de Vieilleville, F.: Analysis and comparative evaluation of discrete tangent estimators. In: 12th International Conference on Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 3429, pp. 140–251. Springer, Berlin (2005)
42. Lachaud, J.O., Vialard, A., de Vieilleville, F.: Fast, accurate and convergent tangent estimation on digital contours. Image Vis. Comput. **25**(10), 1572–1587 (2007)

43. Latecki, L.J., Conrad, C., Gross, A.: Preserving topology by a digitization process. J. Math. Imaging Vis. **8**(2), 131–159 (1998)
44. Lenoir, A.: Fast estimation of mean curvature on the surface of a 3d discrete object. In: 7th International Workshop on Discrete Geometry for Computer Imagery, pp. 175–186. Springer, Berlin (1997)
45. Lenoir, A., Malgouyres, R., Revenu, M.: Fast computation of the normal vector field of the surface of a 3d discrete object. In: 6th International Workshop on Discrete Geometry for Computer Imagery, vol. 1176, pp. 101–109 (1996)
46. Lewiner, T., Gomes, J.D., Jr., Lopes, H., Craizer, M.: Curvature and torsion estimators based on parametric curve fitting. Comput. Graph. **29**, 641–655 (2005)
47. Lien, S.L.C.: Combining computation with geometry. Ph.D. thesis, California Institute of Technology (1984)
48. Lindblad, J.: Surface area estimation of digitized 3D objects using weighted local configurations. Image Vis. Comput. **23**(2), 111–122 (2005)
49. Liu, H., Latecki, L., Liu, W.: A unified curvature definition for regular, polygonal, and digital planar curves. Int. J. Comput. Vis. **80**(1), 104–124 (2008)
50. Liu, Y.S., Yi, J., Zhang, H., Zheng, G.Q., Paul, J.C.: Surface area estimation of digitized 3d objects using quasi-Monte Carlo methods. Pattern Recognit. **43**(11), 3900–3909 (2010)
51. Malgouyres, R., Brunet, F., Fourey, S.: Binomial convolutions and derivatives estimation from noisy discretizations. In: 14th Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 4992, pp. 370–379 (2008)
52. Marji, M.: On the detection of dominant points on digital planar curves. Ph.D. thesis, Wayne State University, Detroit (2003)
53. Matas, J., Shao, Z., Kittler, J.: Estimation of curvature and tangent direction by median filtered differencing. In: 8th International Conference on Image Analysis and Processing. Lecture Notes in Computer Science, vol. 974, pp. 83–88 (1995)
54. Merigot, Q., Ovsjanikov, M., Guibas, L.: Robust Voronoi-based curvature and feature estimation. In: SIAM/ACM Joint Conference on Geometric and Physical Modeling (2009)
55. Nguyen, T., Debled-Rennesson, I.: Curvature estimation in noisy curves. In: 12th International Conference on Computer Analysis of Images and Patterns. Lecture Notes in Computer Science, vol. 4673, pp. 474–481 (2007)
56. Novotni, M., Klein, R.: Shape retrieval using 3D Zernike descriptors. Comput. Aided Des. **36**(11), 1047–1062 (2004)
57. Pottmann, H., Wallner, J., Huang, Q., Yang, Y.: Integral invariants for robust geometry processing. Comput. Aided Geom. Des. **26**(1), 37–60 (2009)
58. Pottmann, H., Wallner, J., Yang, Y., Lai, Y., Hu, S.: Principal curvatures from the integral invariant viewpoint. Comput. Aided Geom. Des. **24**(8–9), 428–442 (2007)
59. Proffitt, D., Rosen, D.: Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges. Comput. Graph. Image Process. **10**(4), 318–332 (1979)
60. Provençal, X., Lachaud, J.O.: Two linear-time algorithms for computing the minimum length polygon of a digital contour. In: 15th International Conference on Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 5810, pp. 104–117 (2009)
61. Provot, L., Gérard, Y.: Estimation of the derivatives of a digital function with a convergent bounded error. In: 16th IAPR International Conference in Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 6607, pp. 284–295. Springer, Berlin (2011)
62. Roussillon, T., Lachaud, J.O.: Accurate curvature estimation along digital contours with maximal digital circular arcs. In: 14th International Workshop in Combinatorial Image Analysis. Lecture Notes in Computer Science, vol. 6636, pp. 43–55. Springer, Berlin (2011)
63. Roussillon, T., Sivignon, I.: Faithful polygonal representation of the convex and concave parts of a digital curve. Pattern Recognit. **44**(10–11), 2693–2700 (2011)
64. Roussillon, T., Sivignon, I., Tougne, L.: On three constrained versions of the digital circular arc recognition problem. In: 15th IAPR International Conference on Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 5810, pp. 34–45 (2009)

65. Sloboda, F., Zaťko, B., Stoer, J.: On approximation of planar one-dimensional continua. In: Advances in Digital and Computational Geometry, pp. 113–160 (1998)
66. Tajine, M., Daurat, A.: On local definitions of length of digital curves. In: 11th International Conference on Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 2886, pp. 114–123. Springer, Berlin (2003)
67. Teague, M.R.: Image analysis via the general theory of moments. J. Opt. Soc. Am. **70**(8), 920–930 (1980)
68. Vialard, A.: Geometrical parameters extraction from discrete paths. In: 7th Discrete Geometry for Computer Imagery. Lecture Notes in Computer Science, vol. 1176, pp. 24–35 (1996)
69. Windreich, G., Kiryati, N., Lohmann, G.: Voxel-based surface area estimation: from theory to practice. Pattern Recognit. **36**(11), 2531–2541 (2003)
70. Worring, M., Smeulders, A.W.M.: Digital curvature estimation. CVGIP, Image Underst. **58**(3), 366–382 (1993)