CVGIP: IMAGE UNDERSTANDING
Vol. 58, No. 3, November, pp. 366–382, 1993

Digital Curvature Estimation

MARCEL WORRING* AND ARNOLD W. M. SMEULDERS

Department of Computer Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Received September 25, 1992; accepted February 17, 1993

This paper sets out to study the effect of digitization on curvature. It is put forward that digital curvature is an estimate with accuracy and precision. A theoretical analysis of the curvature estimation problem identifies the possible sources of errors in digital curvature estimation. In the literature, five essentially different classes of methods are found. From theoretical analysis of the methods, as well as by random experiments on generated arcs, we establish that almost all existing methods suffer from a severe directional inaccuracy and/or poor precision. Errors depend on the method, orientation, and scale, ranging from 1% to over 1000%. We present a practical solution with a residual error between 1% and 60%, also giving recommendations for the required resolution of the image. © 1993 Academic Press, Inc.

1. INTRODUCTION

The accurate and precise estimation of curve properties plays an important role in the interpretation of digital binary data such as maps, engineering drawings, and the like. The accurate measurement of length was discussed in [4, 5].

Curvature also is a key notion in the recognition of objects from digital pictures. Specifically maxima, minima, and zero crossings of curvature carry important shape clues. Applications of these specific events of the curvature function are found for example in robotics [2] and remote sensing [11]. In those applications, curvature is used as a qualitative measure. Used as a quantitative measure, the curvature squared has an important interpretation in the energy it takes to bend an elastic object into its shape. As a consequence, the estimated curvature can be related directly to the physical forces acting upon the object to get it into or keep it in the shape it has. This interpretation of curvature is used in the recognition of cells [25] and in the analysis of heart walls [6].

Digital curvature is computed from a discrete set of points, either representing a digital line or the discrete boundary of some digital object. In fact, this discrete set of points is a representation of some continuous predigitized object. In this paper the interest is not in the curvature of the digital curve, but in the curvature of the predigitized object. In the digitization, exact information on the continuous object is lost and therefore curvature cannot be calculated exactly, but it can only be estimated. In fact, there is a lower bound on the error in measurement that can be achieved [16, 22]. In this paper we analyze methods on the basis of their ability to estimate properly the predigitized curvature. The ability of a method is defined by the accuracy and precision of estimation. These notions are quantitatively captured by the bias and deviation under repeated placement of the continuous object on a random position with respect to the digitization grid.

In the literature on the differential geometry of curves [18], three equivalent formulations of curvature are found. They are respectively based on the orientation of the tangent, the second derivative of the curve considered as a path, or on the local touching circle. Methods in literature on digital curvature measurement are based on either of the three formulations. In [1, 2, 6, 12, 13, 25] digital curvature is based on the orientation of the tangent, [9–11] use a path-based scheme, and [20] fits a circular arc. As a consequence, the methods differ widely in the approach they take to curvature estimation. For the continuous case, the three formulations are equivalent, but not so in the digital case. In this paper we show that their behavior and results vary largely; early results on this subject were presented in [23]. It should be noted that the methods in [1, 2, 9–11, 13] were designed for shape description, rather than estimation. The results in this paper suggest improvements to these methods, which make estimation results more robust and improves on the shape description as well.

All methods for digital curvature to be considered in this paper are local. At every point of the discrete curve any of the methods uses a limited window of fixed size in the computation. From the points in the window a local curvature estimate is made. The choice of a fixed window size implies that curvature features should have compatible levels of detail. For curves with details of different sizes an adaptive window size can be used [19]. Alterna-

* This work was supported by the Dutch Ministry of Economic Affairs through SPIN Grants "3D computer Vision" and "3D Image Analysis."

1049-9660/93 $5.00

tively, the curvature can be calculated at multiple scales simultaneously and studied in scale-space [2, 9, 11]. In contrast to local methods, curvature estimation can also be attacked using global methods, for example by splines [14] in conjunction with generalized cross validation [3]. For the curves on a discrete grid considered here, the method is similar (fixing the scale parameter, see [21]) to the methods in [9, 11].

Global methods which are applied to the grey value image data directly are based on the use of snakes [7] or Fourier parameterized contours [17, 24]. Both methods are based on parameterized boundary templates, yielding curvature directly from the template parameters. These methods are unfit to estimate curvature from binary data and difficult to compare otherwise with the above methods. In this paper only local methods will be considered in further depth.

The framework for curvature estimation and criteria to evaluate curvature estimation are presented in Sections 2.1 and 2.2. A theoretical analysis of the impact of digitization on curvature is presented in Section 2.3. Description of the curvature estimation methods and an evaluation on the basis of theoretical considerations is given in Section 2.4. In Section 3 experimental data is presented to verify the theoretical statements. Recommendations for proper curvature estimation procedures will be presented in the concluding section.

2. CURVATURE ESTIMATION FROM DIGITAL DATA

2.1. Definition of Curvature

Consider a continuous object X with boundary ∂X. Let $\mathbf{x}(s) = (x(s), y(s))^T$ be the length parameterized path following ∂X in a counterclockwise fashion. Definitions of curvature can be found in text books on differential geometry, see for example [18].

By definition the curvature κ of a curve or path $\mathbf{x}$ is given by the directional change of the tangent $\mathbf{t}$ of $\mathbf{x}$.

DEFINITION 1 (Orientation-based continuous curvature).

$$\kappa(s) = \theta'(s)$$
$$\theta(s) = \angle(\mathbf{t}, x^+\text{-axis}).$$

Formulated in this way, the sign of $\kappa(s)$ indicates whether the curve locally at s is convex ($\kappa(s) > 0$), or concave ($\kappa(s) < 0$).

Alternatively one can express the curvature in the norm of the second derivative of path $\mathbf{x}$.

DEFINITION 2 (Path-based continuous curvature).

$$\kappa(s) = \begin{cases} +\|\mathbf{x}''(s)\| & \text{(contour locally convex)} \\ -\|\mathbf{x}''(s)\| & \text{(contour locally concave).} \end{cases}$$

For an arbitrary (nonpathlength) variable u, Definition 2 is reformulated into the following equation giving the correct magnitude of curvature as well as the correct sign:

$$\kappa(u) = \frac{x'(u)y''(u) - x''(u)y'(u)}{(x'(u)^2 + y'(u)^2)^{3/2}}. \tag{1}$$

A third definition is derived from the osculating circle touching at $\mathbf{x}$ in s, defined as the limiting circle through $\mathbf{x}(s - \Delta s)$, $\mathbf{x}(s)$, and $\mathbf{x}(s + \Delta s)$, when $\Delta s \to 0$. Let $r(s)$ be the radius of the osculating circle in $\mathbf{x}(s)$ then:

DEFINITION 3 (Osculating circle based continuous curvature).

$$\kappa(s) = \begin{cases} +\dfrac{1}{r(s)} & \text{(contour locally convex)} \\ -\dfrac{1}{r(s)} & \text{(contour locally concave).} \end{cases}$$

The three definitions are illustrated in Fig. 1.

2.2. Digitization Errors

When the continuous object $\mathbf{X}$ is digitized on a regular grid with gridconstant h a discrete object $\mathbf{X}^*$ results. We employ the following commonly used digitization operator D:

$$D(\mathbf{X}) = \{(x, y) \in \mathbb{Z}^2 | (x/h, y/h) \in \mathbf{X}\}. \tag{2}$$

For the moment, without loss of generality, the gridconstant h is set to one. The role of h in curvature measurement is studied in Section 2.5. The boundary $\partial\mathbf{X}^*$ is the set of all points of $\mathbf{X}^*$ that have a four-connected neighbor in the complement of $\mathbf{X}^*$. The contour $\mathbf{x}(i) = (x(i), y(i))_{i=0,\ldots,N-1}^T$ is the eight-connected path following

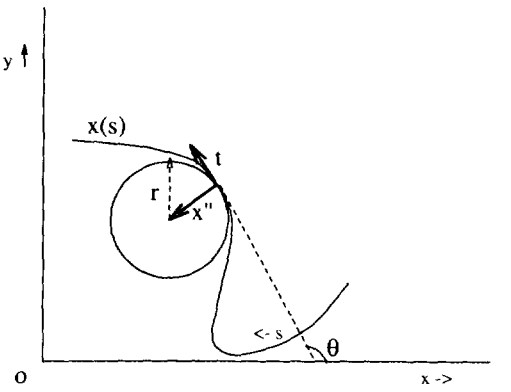

FIG. 1. This figure illustrates the three different definitions of curvature: 1. The derivative of tangent orientation θ; 2. The norm of the second derivative $\mathbf{x}''(s)$; 3. The inverse of the radius r of the osculating circle.

$\partial \mathbf{X}^*$ in a counterclockwise fashion. From the discrete contour, an estimate $\hat{\kappa}$ of the (local) curvature of the predigitized curve is made. As curvature is translation and rotation invariant in the sense that it is not dependent on the ordering of the translation, the rotation, and the curvature calculation step, there are two degrees of freedom in the placement of the object $\mathbf{X}$ on the grid. However, the resulting discrete contour is *not* invariant under translation and rotation of the continuous object. Thus, digital curvature is not only dependent on the predigitized curvature, but also on the position and orientation of the object. As the grid is nonisotropic, we are particularly interested in the orientation dependency. Therefore, the measurement error is studied as an average over all possible positions of the object.

Any continuous curve can locally be approximated as a circular arc with radius equal to the inverse of local curvature (see Definition 3). Therefore, to establish measurement errors, digitized circular arcs of M digital points are considered with varying position and orientation. This can conveniently be achieved by placing discs $B(c, r)$ of fixed radius on a random position c_j with respect to the grid. For every point $\mathbf{x}(i)$ in the digital contour of the disc placed at position c, a local window of size M is considered. From the points in this window, curvature is estimated. The random position is governed by the probability density function $p(c)$. To establish correspondence between the results at different positions c and to study the anisotropy of the grid, the contour $\mathbf{x}$ and the curvature estimate $\hat{\kappa}$ are specified in terms of θ, the predigitized tangent orientation.

Thus, our measure of the quality of each of the methods is the curvature measurement error $E_M(\theta)$ which is an average over all random positions, specified as a fraction of the squared true value κ_0.

DEFINITION 4 (Curvature error). *Given* $\kappa = \kappa_0$,

$$E_M(\theta) = \frac{\int_{c \in \Re^2} p(c)(\hat{\kappa}(\mathbf{x}_c^M(\theta)) - \kappa_0)^2 \mathrm{d}c}{\kappa_0^2} \times 100\%.$$

E_M is composed of two parts. The curvature bias, B_M, expresses the accuracy in analogy with the bias known from measurements in physics.

DEFINITION 5 (Curvature bias).

$$B_M(\theta) = \frac{\int_{c \in \Re^2} p(c)(\hat{\kappa}(\mathbf{x}_c^M(\theta)) - \kappa_0)\mathrm{d}c}{\kappa_0} \times 100\%.$$

The bias in the result is either a consequence of the digitization on the grid or it is due to causes not effected by the digitization. In the latter case, the results of curvature estimation can often be improved by a bias correction.

The second component of the curvature error is the curvature deviation S_M, quantifying the precision of a curvature estimation method. The deviation expresses how the pixel digitization propagates into the estimation of curvature. It is given by

DEFINITION 6 (Curvature deviation).

$$S_M(\theta) = \frac{\sqrt{\int_{c \in \Re^2} p(c)(\hat{\kappa}(\mathbf{x}_c^M(\theta)) - \bar{\kappa}(\theta))^2 \mathrm{d}c}}{\kappa_0} \times 100\%,$$

where $\bar{\kappa}$ denotes the estimated curvature averaged over different positions.

The bias and the deviation relate to the curvature error as

$$E_M(\theta) = B_M^2(\theta) + S_M^2(\theta). \tag{3}$$

2.3. Digital Discs

In this paragraph we study the effect of digitization on curvature. Let $\mathbf{x}_{c,r}(i)$ be the discrete path resulting from digitizing a disc $B(c, r)$ on a grid, where i indexes the different points. First we consider the appropriate parameterization of this digital circular path. Appropriate means that the digital path can be found by taking discrete steps in the parameter i, followed by truncation of the resulting (continuous) coordinate values. If digitization would be isotropic, the parameterization of a digital circular path of N points would be based on a polar coordinate system:

$$\mathbf{x}_{c,r}(i) = \begin{pmatrix} c_x + r\cos\left(\frac{2\pi i}{N}\right) \\ c_y + r\sin\left(\frac{2\pi i}{N}\right) \end{pmatrix}. \tag{4}$$

However, this parameterization is not correct for the digitization of a disc on a regular grid, as for a regular grid the actual change in orientation varies for different points on the discrete contour.

To find the proper regular parameterization, consider the digitization following Eq. (2) of a disc of arbitrary radius centered at origin. Let $\mathbf{x}_0$ be an arbitrary point of $\mathbf{x}(i)$ in the first octant. Considering all neighbors of $\mathbf{x}_0$ on the grid, it is found that neighbors connected to $\mathbf{x}_0$ by a Freemancode 2 or 3 are the only pixels that could possibly follow x_0 in the eight-connected contour of the disc (see Fig. 2). In the eighth octant, 1 and 2 are the only codes found. The Freemancodes 1, 2, and 3 have in common that the associated change in y-value is one. This constant difference of y-coordinates has important consequences for the parameterization of the digital disc and also for the subsequent curvature estimation. Taking an arbitrary

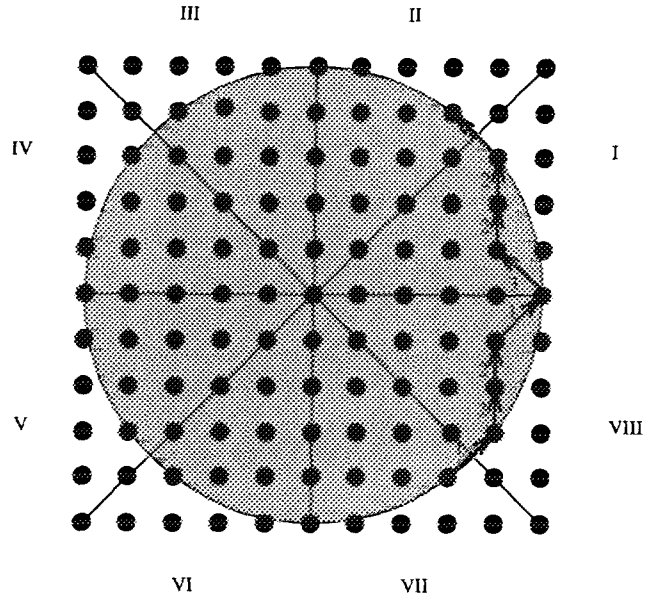

FIG. 2. A continuous disc and its digitization are shown. The contour is represented using Freemancodes. For the part of the boundary shown, the difference in y-coordinate for successive pixels is always equal to one. This result holds for any (continuous) radius of the disc.

starting point, the appropriate parameterization in the eighth and first octant is $y(i) = i$ and $x(i) = \sqrt{r^2 - i^2}$. The digital curve in those octants can be found by taking the floor of the continuous value $x(i)$.

As the y-coordinate has constant difference in octants VIII, I and IV, V, and the x-coordinate in the remaining four, the parameterization is different for every pair of octants. For the octants VIII, I, II, and III the parameterization is given explicitly, the remaining octants are parameterized in similar way:

$$\mathbf{x}_{c,r}(i) = \begin{cases} \begin{pmatrix} c_x + \sqrt{r^2 - i^2} \\ c_y + i \end{pmatrix}, & \left(-\frac{r}{2}\sqrt{2} \le i < \frac{r}{2}\sqrt{2}\right) \\ \begin{pmatrix} c_x + r\sqrt{2} - i \\ c_y + \sqrt{r^2 - (r\sqrt{2} - i)^2} \end{pmatrix}, & \left(\frac{r}{2}\sqrt{2} \le i < \frac{3r}{2}\sqrt{2}\right). \end{cases} \tag{5}$$

Note that the parameterization is derived for a disc centered at the origin. If the disc is placed on a random position with respect to the grid, the parameterization is still correct. However, both x and y have to be truncated. Further, at $i \approx (r/2)\sqrt{2}$, one has to decide whether the parameterization for the first or second octant is valid, based on the values of c and r.

This regular parameterization of a digital disc provides us with a powerful tool to find possible sources of curvature estimation errors. Note that without truncation, both parameterizations yield the same (correct) curvature values.

Consider the norm of the tangent vector. In the isotropic parameterization (Eq. (4)) the tangent has constant size. However, in the regular parameterization the norm of the tangent varies as illustrated in Fig. 3. Two important observations can be made from the figure. First, as the norm varies between 1 and $\sqrt{2}$, any method based on the assumption of unit tangent norm will result in a bias of 41%. Second, the first derivative of the norm shows a discontinuity. This difference in tangent size is also reflected in the length between discrete points of the contour $\mathbf{x}$. As the contour is eight-connected, the distance between successive pixels is either 1 or $\sqrt{2}$. In the main directions of the grid the distance will be dominantly 1, where in the diagonal directions the distance is mostly $\sqrt{2}$. For random lines of varying orientation this is extensively studied in [4]. It is found that the average distance $\bar{d}_{\text{pix}}$ between two discrete grid points on a random line is not 1.000, but

$$\bar{d}_{\text{pix}} = 1.107. \tag{6}$$

Although this value is based on random straight lines, it is shown in the reference that it holds for discrete curved arcs as well.

Now, we consider the x- and y-coordinates of the parameterization. We have particular interest for the coordinates at $i = r\sqrt{2}/2$ at the edge of octants I and II. Taking the limit of the first and second derivatives of the x-coordinate to approach $r\sqrt{2}/2$ from the two different sides leads to:

$$x'_{i\uparrow(r/\sqrt{2})}(i) = -1, \qquad x'_{i\downarrow(r/\sqrt{2})}(i) = -1$$
$$x''_{i\uparrow(r/\sqrt{2})}(i) = \frac{-2\sqrt{2}}{r}, \qquad x''_{i\downarrow(r/\sqrt{2})}(i) = 0.$$

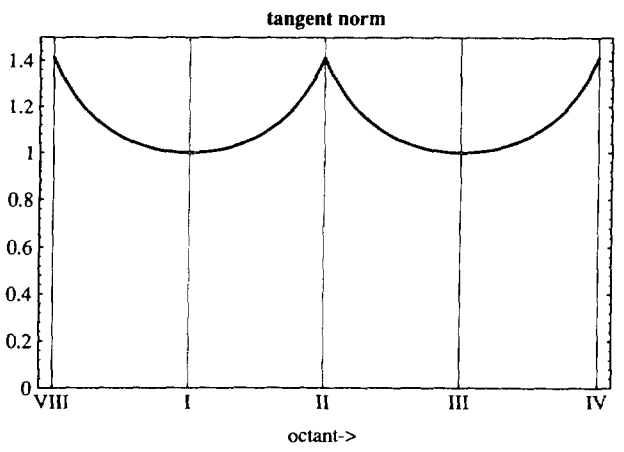

FIG. 3. This figure shows the norm of the tangent vector for the regular parameterization. The norm function is not constant and does not have a continuous derivative.

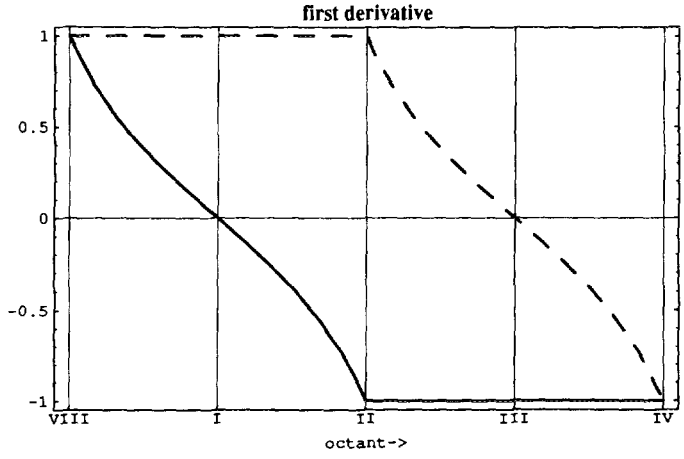

FIG. 4. This figure shows the first derivative of the x-coordinate (solid line) and the y-coordinate (dashed line) of the regular parameterization. Note that for large parts, one of the coordinates has a constant derivative.

It follows that the curve has a continuous first derivative, but its second derivative has a discontinuity of size $2\sqrt{2}/r$. In Fig. 4 and Fig. 5 the first and second derivatives of the parameterization are shown. The second-order discontinuity introduces high frequencies in the Fourier transform of the coordinates. Therefore, any low pass filtering of the coordinates in order to reduce noise, alters the shape of the curve. The difference between the Fourier transform of the isotropic parameterization and the regular parameterization is prominently present as for the isotropic parameterization the Fourier transform of a coordinate yields two, low frequency, δ-pulses only.

Another source of error, which is not directly dependent on the parameterization, is the fact that circular arcs of limited size are considered rather than circles. This

FIG. 5. This figure shows the derivative of the x-coordinate (solid line) and the y-coordinate (dashed line) of the regular parameterization. For both coordinates a discontinuity in the second derivative is found.

$$\left(\frac{\Delta\mathbf{x}_t}{\Delta\mathbf{x}_n}\right)_\alpha = \frac{2\sin\alpha}{1-\cos\alpha}. \tag{7}$$

Example values are

$$\left(\frac{\Delta\mathbf{x}_t}{\Delta\mathbf{x}_n}\right)_{\pi/8} \approx 10, \quad \left(\frac{\Delta\mathbf{x}_t}{\Delta\mathbf{x}_n}\right)_{\pi/4} \approx 4.8, \quad \left(\frac{\Delta\mathbf{x}_t}{\Delta\mathbf{x}_n}\right)_{\pi/2} = 2.$$

It follows that for small arcs, in the direction of the normal, digitization errors have a large impact on the measurement result. For example, in the case of $\alpha = \pi/8$ and $r = 40$ the pixel size is 3.3% of $\Delta\mathbf{x}_t$ and 33% of $\Delta\mathbf{x}_n$. The latter might lead to considerable errors. For the coordinates x and y it follows that if the tangent is oriented along a horizontal (vertical) line of the grid ($\theta_0 = 0 \bmod \pi/2$), digitization has a large impact on the precision of estimated properties of the y-coordinate (x-coordinate). If, on the other hand, the tangent is directed along a diagonal line ($\theta_0 = \pi/4 \bmod \pi/2$), the impact on both coordinates is equal.

2.4. Methods for Estimating Curvature

In the literature on curvature estimation several methods are presented for deriving the curvature of a digital curve. Some of them are explicitly designed to measure the curvature quantitatively, others are being used for their qualitative values. We consider them here for the purpose of measurement. The methods are based on either one of the three curvature definitions given above. The definitions are equivalent in the continuous case but *not* so in the digital case. We use the three definitions to divide the methods in three categories.

Before we get to a description of the actual methods, we study the effect of truncating the Gaussian kernel on the measurement values.

2.4.1. *The Truncated Gaussian Kernel*

The Gaussian kernel is a popular filter used in smoothing. To be used as a digital filter, the Gaussian kernel is truncated to some finite size $2k + 1$. Thus, the discrete Gaussian kernel is given by

$$G_\sigma(i) = \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-i^2/2\sigma^2}\right)_{i=-m,m}, \quad m = \lceil k\sigma \rceil. \tag{8}$$

Differentiating kernels based on the Gaussian kernel are defined in similar way.

Although it seems a minor detail, it is shown in this paper that it is important to consider the truncation of the kernel as a source of error in the curvature estimation. Authors rarely state the value of k used in their method. An exception is found in [9]. For an arbitrary filter F, depending on scale parameter σ, we define the truncation error $\varepsilon_\sigma(F, k)$ as

$$\varepsilon_\sigma(F, k) \equiv \frac{\int_{-\infty}^{-k\sigma} |F_\sigma(t)| + \int_{k\sigma}^{+\infty} |F_\sigma(t)|}{\int_{-\infty}^{+\infty} |F_\sigma(t)|} \times 100\%. \quad (9)$$

For $F_\sigma = G'_\sigma$ and $F_\sigma = G''_\sigma$ this leads to

$$\varepsilon_\sigma(G', k) = e^{-k^2/2} \times 100\%,$$

$$\varepsilon_\sigma(G'', k) = \frac{k}{2} e^{(1-k^2)/2} \times 100\%.$$

In this paper we take m as the smallest integer larger than 3σ following [9]. The truncation errors in this case are $\sigma(G', 3) = 1.1\%$ and $\varepsilon_\sigma(G'', 3) = 2.7\%$.

The truncation error by itself is a measure of the deviation of the filter from its ideal shape; it is not a measure of the performance of the filter in measurement. To illustrate the point, consider convolution of a linear function $f(x) = ax + b$ with a differentiating Gaussian kernel to estimate the first derivative in the point $x = 0$. Using continuous convolution and a nontruncated Gaussian kernel (denoted by $k = \infty$) we have

$$\hat{f}'_\infty(0) = f(x) * G_\sigma(x)|_{x=0} = a.$$

Let $B_{\text{trunc}}(f', k)$ denote the bias in the estimated derivative f' when using a truncated kernel of size $2k + 1$, relative to the true value a:

$$B_{\text{trunc}}(f', k) = \frac{a - \hat{f}'_k(0)}{a} \times 100\%. \quad (10)$$

In Table 1 the values of ε_k and B_{trunc} are given for sample values of k. Note, these results were obtained using continuous convolution and are therefore not dependent on the position of the curve or its digitization yet. The table demonstrates that a statement on the truncation error of the kernel does not reveal the estimation error it causes.

TABLE 1
The Truncation Error ε_σ (Eq. (9)) Due to Truncation of the Gaussian Kernel to a Filter of Size $2k + 1$ Points

k	$\varepsilon_\sigma(G', k)$	$B_{\text{trunc}}(f', k)$
1	60.7%	81.1%
2	13.5%	26.1%
3	1.1%	2.9%
4	0.0%	0.0%

Note. This indicates the deviation of the kernel from its ideal shape. Further, it gives the actual error in the estimation of the first derivative of a linear function according to Eq. (10).

2.4.2. *Methods Based on Orientation*

Digital estimation methods for curvature based on the orientation based definition (Definition 1) can be found in [1, 2, 6, 12, 13, 25]. In practice, an estimation of the orientation is made and from there the differential needed in Definition 1 is found.

Rosenfeld [13] and Young [25] use the direction of the Freeman contour chain code as the estimate of orientation, recognizing in each point eight different pixels as neighbors. Young then computes curvature by repeated uniform filtering of the differential of the Freeman codes. Rosenfeld follows a different procedure, but it can be reformulated into the previous one.

Asada [2] uses greyvalue gradient information rather than the tangent to estimate the orientation of the contour, also recognizing a limited set of different orientations. From the orientation, curvature is estimated by Gaussian differential filtering. The Gaussian differential kernel has a superior frequency response over the previously mentioned ones. Therefore, we investigate the principle of this method of differential filtering only. As we limit ourselves to geometrical information, we will not follow the orientation estimation step used in his procedure. The orientation of the Freeman contour chain code is used instead.

Let $*$ denote discrete convolution; then the above methods [2, 13, 25] summarized as Method I for the estimation of curvature.

DEFINITION 7 (Method I: Chain code).

$$\hat{\kappa}_1(i) = \hat{\theta}(i) * G'_\sigma$$
$$\hat{\theta}(i) = \tan^{-1}\left(\frac{y(i+1) - y(i)}{x(i+1) - x(i)}\right).$$

To gain insight into the curvature estimation properties of the method, we apply Definition 7 to the parameterization given in Eq. (5), leading to the analytic description of the orientation function used in the method,

$$\hat{\theta}(i) = \tan^{-1}\left(\frac{1}{\sqrt{r^2 - (i+1)^2} - \sqrt{r^2 - i^2}}\right).$$

Curvature $\hat{\kappa}(i)$ is the analytical derivative of this expression with respect to i, yielding a fairly complicated func-

tion. The relative error $B_{\text{chain}}(i, r)$ in the curvature estimation derived from the analytical model of the chain code method is given by

$$B_{\text{chain}}(i, r) = \frac{\kappa - \hat{\kappa}(i)}{\kappa} \times 100\%, \quad \kappa = 1/r. \tag{11}$$

Interesting values of i are $i = 0$ and $i = r\sqrt{2}/2 - 1$, corresponding to the main direction and the diagonal direction of the grid, respectively. Example values are

$$B_{\text{chain}}(0, 10) = 0.3\%,$$

$$B_{\text{chain}}(r\sqrt{2}/2 - 1, 10) = 33.6\%,$$

$$B_{\text{chain}}(0, 1000) = 0.0\%$$

$$B_{\text{chain}}(r\sqrt{2}/2 - 1, 1000) = 41.3\%.$$

The example values of the analytical model clearly show the influence of the anisotropy of the grid on the final result. In the main directions of the grid, curvature approximation is not effected by the use of chain codes. However, in the diagonal directions, for large r, the error gets close to $\sqrt{2}$, resulting from the incorrect assumption that the tangent has constant size. Note that the error is only an upper bound as in practice the directions are restricted to take one out of eight values. Apart from the fact that tangent size is different for different orientations, the average length between two discrete pixels (Eq. (6)) is not taken into account either. Using the equation, it follows that a relative bias B_{pix} results.

$$B_{\text{pix}} = 10.7\%. \tag{12}$$

One could try to avoid the problem of length differences in the eight-connected contour by using a four-connected contour instead. However, the orientation estimation becomes very poor in this case and therefore the use of four-connected chain codes is not considered here.

Another consequence of the varying distance between pixels is that the effective sampling distance on the orientation function θ varies along the contour, whereas the use of linear filtering techniques requires an equidistant sampling. Duncan [6] observed that this difference in effective sampling distance can be reduced by resampling of the discrete contour [15]. Thus, a new contour $\mathbf{x}_{\text{res}}(i) = (x_{\text{res}}(i), y_{\text{res}}(i))^T$ is defined with n equidistant samples on the continuous polygon connecting the points of $\mathbf{x}$. We choose to resample the curve using $n = N$, enabling a fair comparison of its performance to other methods. After resampling the method of Definition 7 can still be applied, lifting the restriction on the number of different estimated orientations.

We make one addition to the method. Using Eq. (6), we apply a multiplicative bias correction to correct for the bias B_{pix} as given in Eq. (12). This leads to the following curvature estimation method.

DEFINITION 8 (Method II: Resampling).

$$\hat{\kappa}_{\text{II}}(i) = \frac{\hat{\theta}(i) * G'_\sigma}{1.107}$$

$$\hat{\theta}(i) = \tan^{-1}\left(\frac{y_{\text{res}}(i+1) - y_{\text{res}}(i)}{x_{\text{res}}(i+1) - x_{\text{res}}(i)}\right).$$

Method II can be seen as a variant of Method I corrected, to some degree, for the anisotropic effects of the grid. Small anisotropic effects pertain, however.

In Methods I and II smoothing of the digital data is performed when estimating the derivative of orientation. An alternative method is followed by Anderson [1] and in the methods surveyed by O'Gorman [12], where smoothing is achieved in the estimation of orientation. Curvature is calculated as the angular difference between two straight lines fitted to the data in the neighborhood of the point under consideration. Linefitting in [12] is achieved by connecting two points of the curve, which are a fixed number of points apart. Methods differ only in the position of the two straight lines with respect to one another. Extreme cases are straight line segments having one endpoint in common and segments that are shifted one point only.

None of the methods connecting two points is robust as a one-pixel shift in one of the endpoints causes a large change in orientation and, hence, in the estimate of curvature. A better procedure is followed in [1], where linefitting is based on minimization in a window of size $M = 2m + 1$ of squared distances to the line. Small changes to the position of one of the points on the curve has a small influence on the estimated orientation. Therefore, the latter method will be considered in this paper as a representative of all linefit-based methods. In the reference, the change in orientation is computed using efficient matrix calculations. For the purpose of comparison, it is convenient to rewrite the change of orientation as a convolution of estimated orientation with the differentiating kernel $\{+1, -1\}$. Curvature is calculated by dividing the change of orientation by the distance between the centers of the two windows. Let l_θ denote a line with orientation θ; then the linefit methods can be summarized as

DEFINITION 9 (Method III: Linefit).

$$\hat{\kappa}_{\text{III}}(\,) = \frac{\hat{\theta}(i) * \{+1, -1\}}{d(\mathbf{x}(i+1), \mathbf{x}(i))}$$

$$\hat{\theta}(i) = \arg\min_\theta \left\{ \sum_{j=-m}^{m} w(j)(d(\mathbf{x}(i-j), l_\theta)^2) \right\}$$

$$w(j) = G_\sigma(j),$$

where $d(\cdot, \cdot)$ denotes the Euclidean distance. For a fair comparison with the other methods we introduce the weighting function w. This is a minor deviation from the original method as in the reference uniform weighting is used.

As orientation estimation is based on a larger number of points, it will be more accurate than orientation estimation in Method I or II. However, the differentiating filter used in the procedure has poor frequency characteristics as it amplifies high frequency noise considerably.

To find the effect of the filtering step on curvature estimation, assume the linefit procedure yields an estimated orientation corrupted with independent zero mean noise with standard deviation σ^*. The standard deviation of the noise in the estimate of the derivative is $\sigma^*\sqrt{2}$. As the true derivative value may be small, the effect of small amounts of noise might be of considerable influence on the quality of estimation. As an example, consider the curvature estimation of a disc with radius $r = 25$. If the standard deviation in estimated orientation is only 1°, its deviation is 61.7% of the curvature value.

Another source of inaccuracy is the fact that the estimated derivative is divided by the length between discrete pixels. The distance between pixels is restricted to either 1 or $\sqrt{2}$, a poor estimate of the predigitized arclength.

2.4.3. *Methods Based on Paths*

Lowe [9] and Mokhtarian [11] start from Definition 2. The authors obtain smooth derivative estimates of the x- and y-coordinate sequences by convolution with differentiated Gaussian kernels. As the parameter i in general is not a pathlength variable, Eq. (1) is used to compute curvature. The method by Medioni [10] is also based on Definition 2, but derivatives are found by fitting splines through the data points. It is easy to prove, by considering the different steps taken in the procedure, that the spline-fitting procedure again boils down to linear filtering of the coordinate sequences. Reformulated in this way, one can compare the frequency response of the method in [10] to the methods in [9, 11]. It turns out that a Gaussian kernel can be found (by tuning σ) with nearly identical frequency response. So, summarizing the three methods gives

DEFINITION 10 (Method IV: Path).

$$\hat{\kappa}_{\mathrm{IV}}(i) = \frac{\hat{x}'(i)\hat{y}''(i) - \hat{x}''(i)\hat{y}'(i)}{(\hat{x}'(i)^2 + \hat{x}'(i)^2)^{3/2}}$$

$$\hat{x}'(i) = x(i) * G'_\sigma(i),$$

$$\hat{x}''(i) = x(i) * G''_\sigma(i),$$

$$\hat{y}'(i) = y(i) * G'_\sigma(i),$$

$$\hat{y}''(i) = x(i) * G''_\sigma(i).$$

As regards the effectiveness of Method IV to measure curvature, it is noted that low pass filtering of the coordinates is used. In Section 2.3 it is found that due to the nonisotropic nature of the grid, low pass filtering yields inaccurate results as the second-order derivatives of the curve show a discontinuity.

As concerns the truncation of the Gaussian kernel it is important to consider the shape of the second derivative. Large parts of the derivative are zero. Therefore, if the truncated part of the kernel falls into the part where the derivative is nonzero, its impact on the total estimation result might actually be very large. Much larger than the errors following from Eq. (9).

It is also argued that, for small arcs, one of the coordinatee is significantly affected by the digitization if the tangent is in the main direction of the grid. Weighted average of this coordinate will therefore be very imprecise, resulting in unreliable curvature estimates.

The path-based methods have the unique property that the curve corresponding to the estimated curvature function $\hat{\kappa}$ can be calculated directly by convolution of the coordinates of the curve by a Gaussian kernel. The parameter of the kernel should be chosen equal to the parameter used in the differential filtering. As observed in [9], such Gaussian smoothing of the path coordinates causes a shrinking effect on the curve. That is, locally the smoothed curve tends to move towards the local center of curvature. In the reference this movement of the curve is corrected for. The equations in the reference can be rewritten to give the relative bias B_{shrink} in curvature estimation on a disc as a function of the ratio $\rho = \sigma/r$:

$$B_{\mathrm{shrink}}(\rho) = (e^{\rho^2/2} - 1) \times 100\%. \tag{13}$$

Example values are

$$B_{\mathrm{shrink}}(0.08) = 0.3\%$$

$$B_{\mathrm{shrink}}(0.16) = 1.3\%$$

$$B_{\mathrm{shrink}}(0.32) = 5.3\%$$

$$B_{\mathrm{shrink}}(0.64) = 22.7\%.$$

It follows that only for large values of the ratio σ/r is the error significant. Bias correction for the shrinking effect is not applied in this paper.

2.4.4. *Methods Based on the Osculating Circle*

Definition 3 leads to fitting a circular arc with radius r and center c on the local contour data [8, 20]. The difference in the two references lies in the choice of the criterion to optimize. The criterion in [8] is based on length, whereas the criterion in [20] is based on area. The latter is preferred as the area-based criterion leads to closed form solutions for the center and the radius of the disc. Although not used for that purpose in the reference, the

radius r can be used for the computation of curvature. The method is given by

DEFINITION 11 (Method V: Arcfit).

$$\hat{\kappa}_V(i) = \begin{cases} +\dfrac{1}{\hat{r}(i)} & \text{(contour locally convex)} \\ -\dfrac{1}{\hat{r}(i)} & \text{(contour locally concave)} \end{cases}$$

$$(\hat{c}, \hat{r}) = \arg\min_{(c,r)} \left\{ \sum_{j=-m}^{m} (w(j)(r^2 - |\mathbf{x}(i-j) - c|^2))^2 \right\}$$

$$w(j) = G_\sigma(j).$$

As before we introduce a weighting function w to allow for comparison. It should be noted that the arcfit method has the unique property, among all the methods considered, that it will not only give the radius of curvature, but it also yields $\hat{c}$, an estimate of the local center of curvature.

For the purpose of curvature measurement it is important to note that the arcfit method is meant for fitting arcs encompassing the full boundary of the disc. Here we apply the method to arcs of limited size. To make this observation more precise, we have to consider the actual steps taken in the method. First note that the method is invariant under translation and rotation of the discrete points. The equations in the reference are simplified if points are translated such that the averages of $x(i)$ and $y(i)$ in the local window are equal to zero. It is assumed in the following that this translation is already performed. Let Σ_f denote the average of f in the local estimation window, then the estimated center $\hat{c}$ is given as the solution of the following matrix equation:

$$\begin{pmatrix} \Sigma_{x^2} & \Sigma_{xy} \\ \Sigma_{xy} & \Sigma_{y^2} \end{pmatrix} \begin{pmatrix} \hat{c}_x \\ \hat{c}_y \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \Sigma_{xr^2} \\ \Sigma_{yr^2} \end{pmatrix}, \tag{14}$$

with $r^2(i) = x^2(i) + y^2(i)$. The left-hand side matrix is equal to the matrix used in Method III. There the direction of the eigenvector corresponding to the largest eigenvalue is taken as an estimate of the direction of the tangent. Consequently, the eigenvector corresponding to the smallest eigenvalue is an estimate of the normal to the curve. The eigenvectors can be used to diagonalize the left-hand side matrix. Let the resulting coordinate system be given by unit vectors in the direction of $\hat{\mathbf{t}}$ and $\hat{\mathbf{n}}$ and let (x_t, y_n) denote the point $\mathbf{x}$ with respect to this coordinate system with similar notation for c. Then the solution to Eq. (14) and the estimate $\hat{r}$ can be written as

$$\begin{pmatrix} \hat{c}_t \\ \hat{c}_n \end{pmatrix} = \begin{pmatrix} \Sigma_{x_t r^2} \\ \Sigma_{y_n r^2} \end{pmatrix}, \quad \hat{r}^2 = \hat{c}_t^2 + \hat{c}_n^2 + \frac{1}{2m+1} \Sigma_{r^2}.$$

These equations yield the same result as the equations in the reference, but rewritten into this form they yield more insight into the method. It follows that the estimate for one coordinate of the center is mainly based on information in the direction of the tangent, where the other coordinate is based on information in the direction of the normal. As indicated in Eq. (7), the information in the direction of the normal might be very inaccurate for small arcs. Errors in the estimate of the center directly propagate in the estimated radius and curvature. Therefore, curvature estimates are poor for small arcs.

2.5. Parameter Tuning and Resolution

An issue in practical curvature measurement is the question what resolution and scale parameter is required to satisfy a certain present constraint on accuracy or precision. To that end, we parameterize the notion resolution in the context of curvature measurement.

Consider the curvature measurement of a continuous arc of size $\Delta\theta$ rad and radius r_0. Without loss of generality the radius r_0 of the continuous disc is set to one. After digitization on a regular grid with gridconstant h (Eq. 2)), a disc with discrete radius r results, with $r \approx 1/h$. As we have a direct relation between r and h, the discrete radius r is taken as the parameter of resolution.

The discrete arc corresponding to the continuous arc of size $\Delta\theta$ consists of M discrete samples. As the average distance between points is 1.107 (Eq. (6)), M is related to $\Delta\theta$ and r by

$$M \approx \frac{r\Delta\theta}{1.107}.$$

The scale parameter σ should be chosen such that the window of size $M = 2m + 1$, with m the smallest integer larger than 3σ (Eq. (8), with $k = 3$), covers the complete arc of M points. Thus, σ and r are related by

$$\sigma \approx \frac{r\Delta\theta}{6 \times 1.107}. \tag{15}$$

This formula is used to generate Fig. 12 (sec Section 3.7).

This concludes the evaluation of the methods on the basis of theoretical considerations. In the next section experiments are performed to verify the theoretical conclusions.

3. EXPERIMENTS

In this paragraph, for the various methods, the integrals B_M (Definition 5) and S_M (Definition 6) are approximated using Monte Carlo experiments. As curvature is translation invariant, a uniform distribution of the centers c_j is assumed. Further, c can be assumed to lie in $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$. The experiments are performed on discs with the

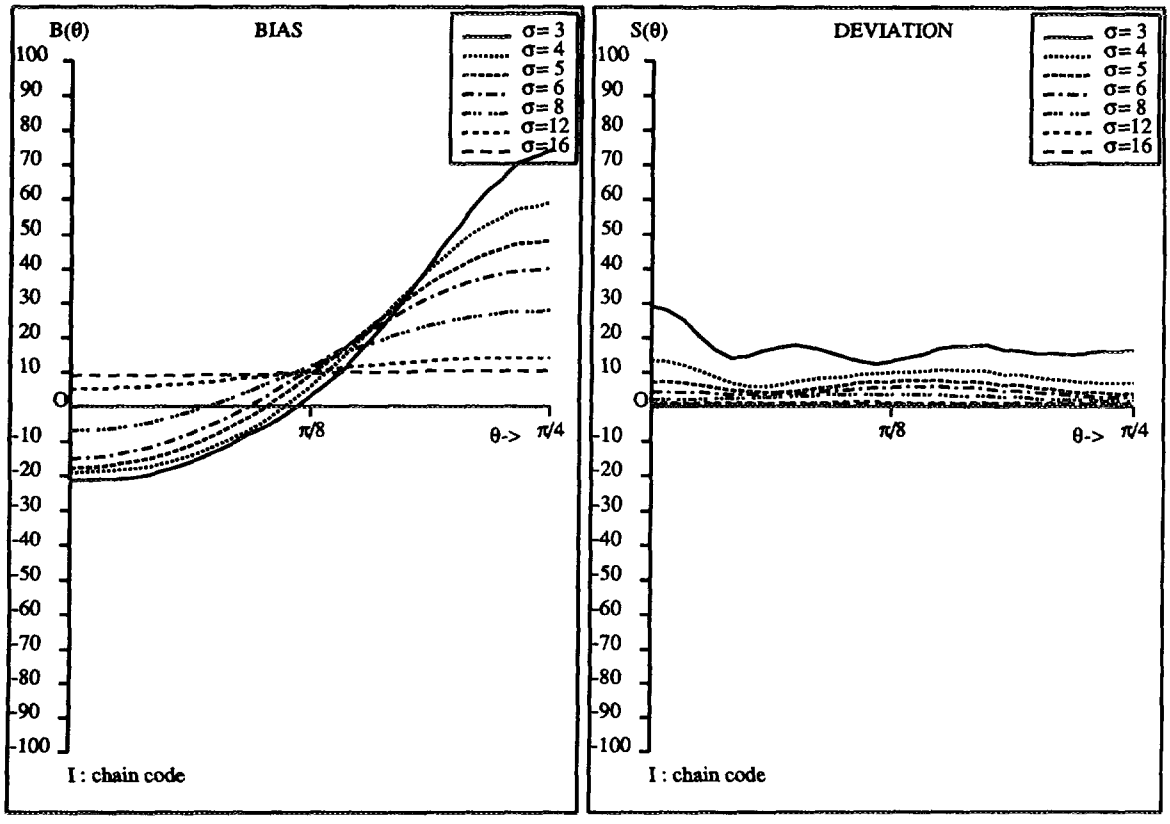

FIG. 6. The bias B and deviation S for curvature measurement on a disc with curvature $\kappa = \frac{1}{25}$ (i.e., $r = 25$) are shown for the method based on chain codes (I).

following curvature values, representing a typical range of curvature values to encounter in practical situations:

$$\kappa = \tfrac{1}{10}, \tfrac{1}{15}, \tfrac{1}{20}, \tfrac{1}{25}, \tfrac{1}{30}, \tfrac{1}{35}, \tfrac{1}{40}.$$

For each r, the scale parameter σ for all methods is subsequently set to

$$\sigma = 3.0, 4.0, 5.0, 6.0, 8.0, 12.0, 16.0.$$

For each σ, windowsize M is based on Eq. (8) with $k = 3$. The discs are placed on the grid in random position $N = 100$ times. Due to the symmetry of the grid, the results are equal modulo $\pi/2$. Furthermore, inside such an interval of size $\pi/2$, results are symmetric around $\theta = \pi/4$. These observations were exploited in the experiments. Therefore, in effect, the computation of B_M and S_M are based on $n = 800$ circular arcs of size $\pi/4$. In the sequel the subscript M will be omitted. For the disc of radius 25 ($\kappa = 1/25$) the results are shown in Figs. 6–10 and discussed for all κ in the subsequent paragraphs.

3.1. Chain Code

The chain code based method (I) has an orientation dependent bias. For the radii and scales considered, the bias is between -22% and 91%. The deviation ranges between 0.6% and 52%. For the disc of radius $r = 25$ see Fig. 6. For small σ it is observed that the difference in bias is larger than the 41% as expected from Eq. (11). An explanation is found in the fact that in the analysis orientation is not restricted to take one out of eight different values. For $r = 25$ and $\sigma = 3.0$, the difference in bias between $\theta = 0$ and $\theta = \pi/4$ equals 95%, substantially exceeding the deviation of 30% maximum. It is concluded that the estimation error E is dominated by the bias (see Eq. (3)).

For larger σ, the error due to length differences between pixels are smoothed out and therefore not prominently present. In the limit case of large σ (here $\sigma = 16.0$) a bias close to the bias B_{pix} is found (Eq. (12)).

3.2. Resampling

As expected, the resampled version (method II) shows a reduction of the orientation depending error found for Method I. The bias now ranges from -8.2% to 17%. The deviation for Method II is approximately equal to the deviation of method I (0.4% to 49%). In Method II, for $r = 25$ and $\sigma = 16.0$ the bias is only -1%, where the deviation is even lower than the absolute bias (see Fig. 7). Therefore, it is concluded that the resulting total error

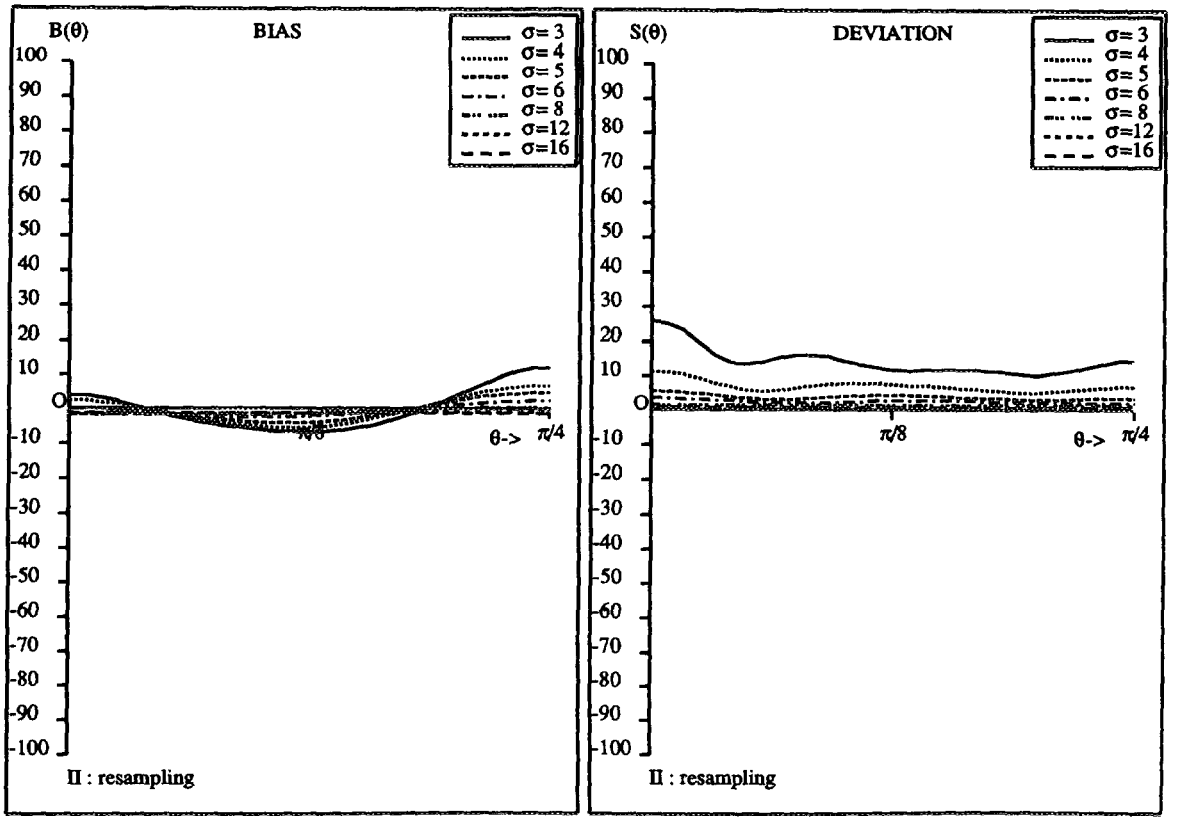

FIG. 7. The bias B and deviation S are shown for the resampling method (II) for $r = 25$.

E is very small. In fact, the overall performance of Method II is superior to all other methods considered.

As indicated earlier, small anisotropic effects remain after resampling. To study this phenomenon, we generated discs of different radii and centered at the origin. From there, we calculated the average distance between two points on the curve, before and after resampling. Further, we calculated the deviation from the mean length for both the original curves and the resampled curves. This resulted in the results shown in Table 2.

A number of interesting observations can be made from the table. The average length and the deviation for the length of the points before resampling seem to be approximately constant with respect to the radius r. For the resampled curve, deviation is a slowly decreasing function of r. In general, the deviation for the resampled curve is three to six times smaller than for the nonresampled curve. A considerable improvement, resulting in reliable curvature estimates.

TABLE 2
The Mean Distances between Points on the Curve Before and After Resampling as Well as the Deviation from the Mean

	r	10	20	30	40	100	200
Original	Mean	1.18	1.18	1.18	1.18	1.17	1.17
	Deviation	0.205	0.205	0.205	0.205	0.204	0.204
Resampled	Mean	1.13	1.13	1.14	1.15	1.14	1.14
	Deviation	0.058	0.049	0.045	0.041	0.039	0.036

Note. Values are given as a function of the radius r.

3.3. Linefit

Over all discs, Method III, based on line fitting, has a bias between -27% and 24%. Deviation never gets below 5.8% and is 68% maximum. For $r = 25$ and small σ, Method III, shows a small bias (see Fig. 8). It shows high deviation as expected from the larger errors in the orientation due to the small number of points considered. These errors propagate to much larger errors in the final outcome due to the method of differential filtering used in Method III. For larger σ, the bias increases and the deviation remains poor, although orientation estimation is better.

To substantiate the observations, we performed an experiment by placing the disc of radius $r = 25.0$ at 100 random positions. From there, we calculated the deviation in estimated orientation, derivative, and curvature. Results are given in Table 3, relative to true curvature $\frac{1}{25}$. It follows that propagation of the noise is not as bad as expected from the considerations in Section 2.4.2 for

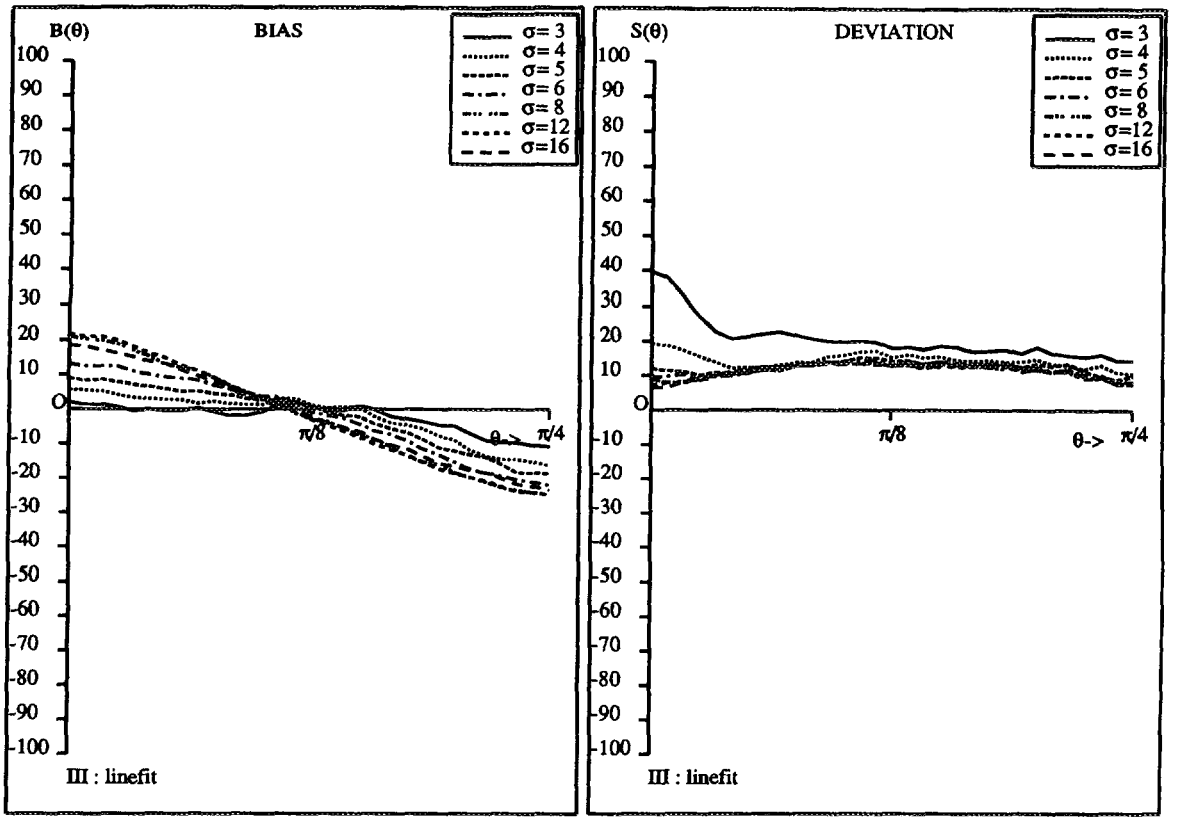

FIG. 8. The bias B and deviation S are shown for the linefit method (III) for $r = 25$.

independent noise. This indicates that the errors for one point are correlated with the errors of its neighbors. For the large kernel with $\sigma = 16.0$ the deviation in the derivative is much smaller than the deviation in curvature. This is a consequence of the poor method of estimation of the predigitized arclength used.

3.4. Path

For all discs and scales, the path-based method (IV) shows poor performance as expected from the problems arising in smoothing a discontinuous function. Bias is between 8.6% to 492% and deviation is 16%–378%. For $r = 25$ and $\sigma = 3.0$ (see Fig. 9), bias ranges between 74% and 113%, where the deviation reaches even higher values (151%–236%). (see Fig. 9). This improves for higher scales, but for $\sigma = 16.0$ bias is still 26% to 33%. Standard deviation for this value of σ is approximately 16%. It is also observed that indeed errors are largest in the main directions of the grid.

Results are expected to improve for larger kernels (for fixed σ) as the errors due to truncation are reduced (cf. Section 2.3). To study the effect of the truncation of the kernel, we consider the disc of radius $r = 25$ and repeat the experiments for the path-based method using a kernel with $k = 5$ instead of $k = 3$. The minimum and maximum bias and deviation for the two cases are shown in Table 4.

The table indicates that a drastic improvement in performance is achieved by using larger kernel sizes (for fixed σ). In fact, as illustrated in Section 2.4.1, truncation errors for $k = 3$ are much larger than the truncation error as given by Eq. (9). The use of larger values of k does not pose a problem for closed curves as the coordinate functions are periodic. For open curves the larger kernel causes practical problems at the endpoints. Further, the use of a large kernel reduces the local character of curvature estimation.

TABLE 3
To Study the Performance of the Linefit Method a Disc of Radius $r = 25$ Is Placed at Random Positions on the Grid 100 Times

σ	3.0		5.0		16.0	
Deviation in	Min	Max	Min	Max	Min	Max
$\hat{\theta}(i)$	24.2	84.7	15.7	40.3	11.4	35.5
$\hat{\theta}'(i)$	8.0	43.3	2.7	12.9	0.9	3.3
$\hat{\theta}(i)$	13.0	45.9	5.8	18.8	1.6	14.3

Note. The deviations in estimated orientation, derivative, and curvature are established. Shown are the minimum and maximum deviations found, relative to the true curvature $\kappa = \frac{1}{25}$.

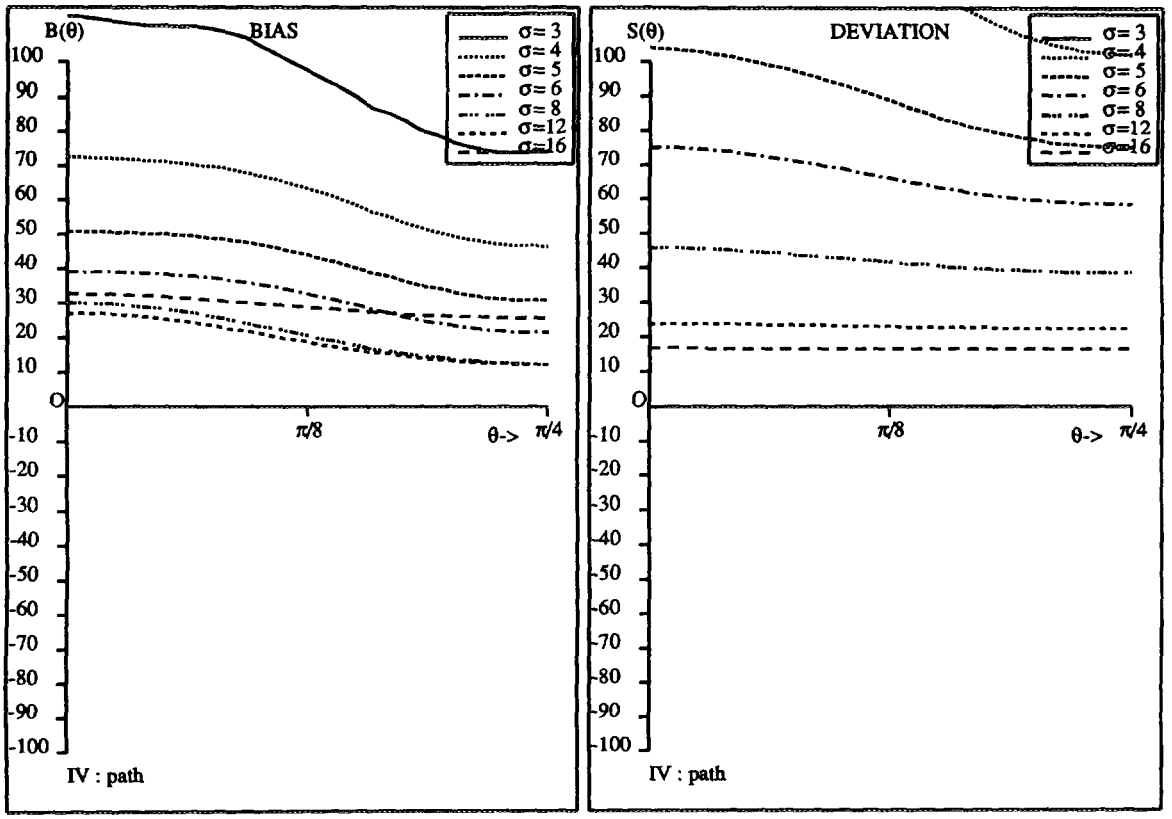

FIG. 9. The bias B and deviation S are shown for the path based method (IV) for $r = 25$.

For both kernels $\sigma = 16.0$ leads to a bias of approximately 30%. This is close to the error predicted to occur as a consequence of the shrinking of the curve ($B_{shrink}(\frac{16}{25}) = 22.7\%$).

3.5. Arcfit

Method V, based on arc fitting shows the largest maximum errors (bias 6.2%–511%, deviation 0.4%–819%). Those maximum values are found for the disc of radius $r = 40$ and scale $\sigma = 3.0$. For all discs the deviation for small σ is large (see, for example, Fig. 10). This especially holds for small σ and large r, a direct consequence of the observation that for small arcs, information in the direction of the normal is not reliable (cf. Section 2.4.4). For most of the cases, the error is largest for $\theta = 0$; however, for the larger discs large errors are found for an orientation of approximately $\sigma = \pi/16$. It turns out that for these values of r the quantization error has its largest impact in those directions. Larger σ result in improved deviation, as larger arcs are considered.

TABLE 4

The Differences in Bias $B(\theta)$ and the Deviation $S(\theta)$ When Different Values of κ (see Eq. (8)) Are Used for Truncation of the Kernels in the Path-Based Method

σ		3.0		5.0		16.0	
Measure	Size	Min	Max	Min	Max	Min	Max
B	k = 3	73.7	113.1	30.6	50.8	25.5	32.6
	k = 5	1.5	3.9	−1.5	7.7	28.7	35.7
S	k = 3	150.6	235.9	75.3	104.0	16.4	16.7
	k = 5	6.7	38.8	2.9	9.9	0.9	1.2

Note. Curvature is measured on the disc of radius $r = 25$.

The arcfit method also yields the local center of curvature; to illustrate its performance we calculate the average center position for some radii r and scales σ. The average position is presented as the curve $(\bar{\hat{c}}_x(\theta), \bar{\hat{c}}_y(\theta))$ and the average of $\hat{r}(\theta)$ as the curve $(\bar{\hat{r}}(\theta)\cos\theta, \bar{\hat{r}}(\theta)\sin\theta)$. Note that ideally the average center is a point and the curve representing the average radius is equal to the original disc. Results are shown in Fig. 11.

For $r = 40$ the large error for $\theta = \pi/16$ is observed. From the figure it follows that this large error in fact is mainly a consequence of errors in the estimate of the center in the direction of the normal. By comparison of the figures for $r = 40.0$, $\sigma = 3.0$ and $r = 100.0$, $\sigma = 7.5$ it is concluded that bias is due to the quantization error

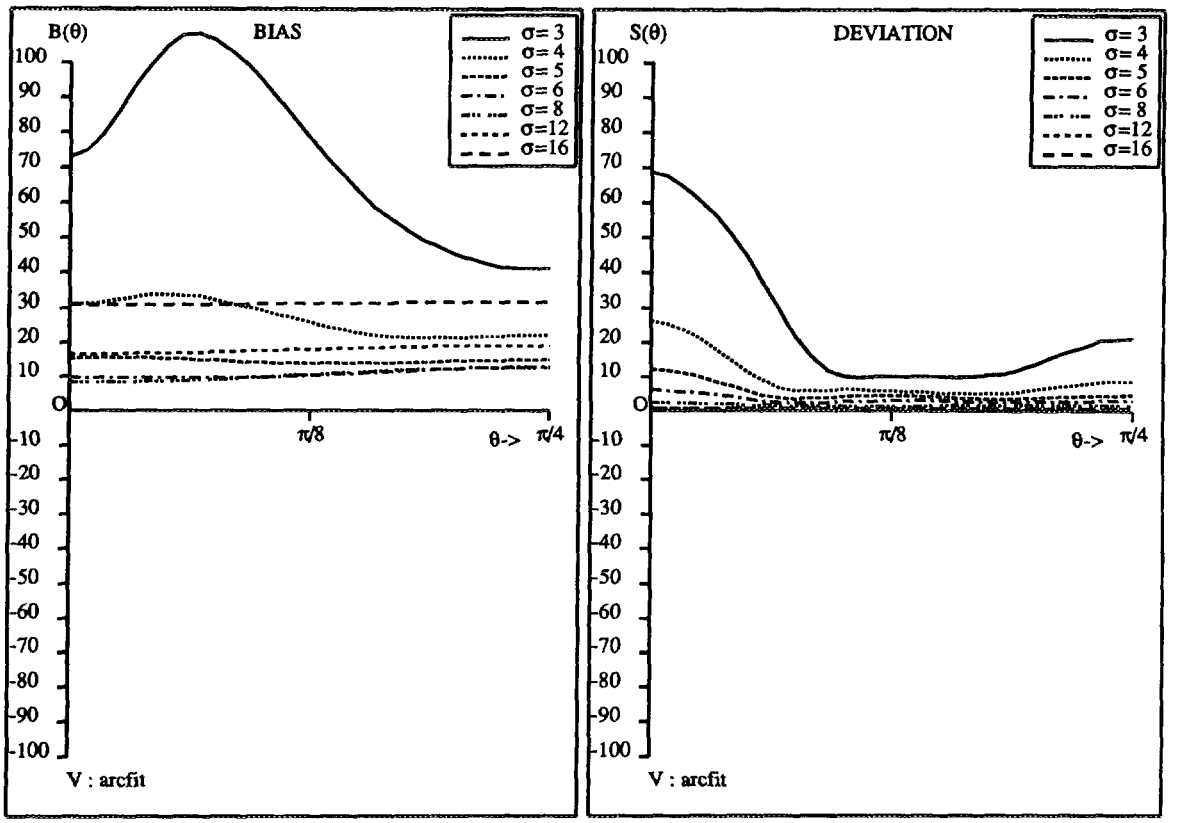

FIG. 10. The bias B and deviation S are shown for the arcfit method (V) for $r = 25$.

as well, where we adhered to the relationship between r and σ in Eq. (15).

3.6. Experiments on Elliptical Figures

The analysis so far was based on measurement of objects with constant curvature function. To establish the robustness against change of curvature, experiments are performed in much the same way as before, but using ellipses instead of discs. These ellipses are chosen such that the maximum and minimum curvature of the corresponding curvature function are equal to 1/10 and 1/40, respectively. The experiments are done for fixed orientation of the ellipse and repeated for several orientations.

Results are not shown here. It was found that the quality of the methods is comparable with the results on the discs. Considering the average estimated curvature it is found that Methods I, III, and IV introduce spurious maxima and minima of curvature, where in many cases the true curvature extremes are missed.

3.7. Experiments on Resolution

The relation between resolution and measurement errors is studied on the basis of Section 2.5. For increasing r and fixed $\Delta\theta$, Eq. (15) is used to find corresponding values for σ.

To show the relation between resolution, on the one hand, and accuracy and precision, on the other, the bias and deviation for arcs with radii between 10 and 140 are computed for $\Delta\theta = \pi/4$ and $\Delta\theta = \pi/2$, respectively. For the resampling method, the method with best performance, errors are shown in Fig. 12. Note that in the figures the scale of the y-axis has changed with respect to previous figures.

From the figure and further analysis of the experiments we conclude that bias is mainly a function of the local orientation and size of the arc and not of resolution. In fact for the smaller of the two arcs ($\Delta\theta = \pi/4$), bias ranges between -7% and $+5\%$, depending on the orientation. For the larger arc the range is between -3% and -1%.

In contrast, the precision of the method does improve with the number of samples. For the smaller arc, many more samples are needed to reach a similar precision. So, as a rule of thumb when measuring curvature, resolution is found from Fig. 12 based on the required precision on the smallest arc of interest. From there, using Eq. (15) one computes the appropriate scale parameter σ.

4. CONCLUSIONS

Computation of curvature from a digitized curve is a nontrivial task which should be considered with care. It

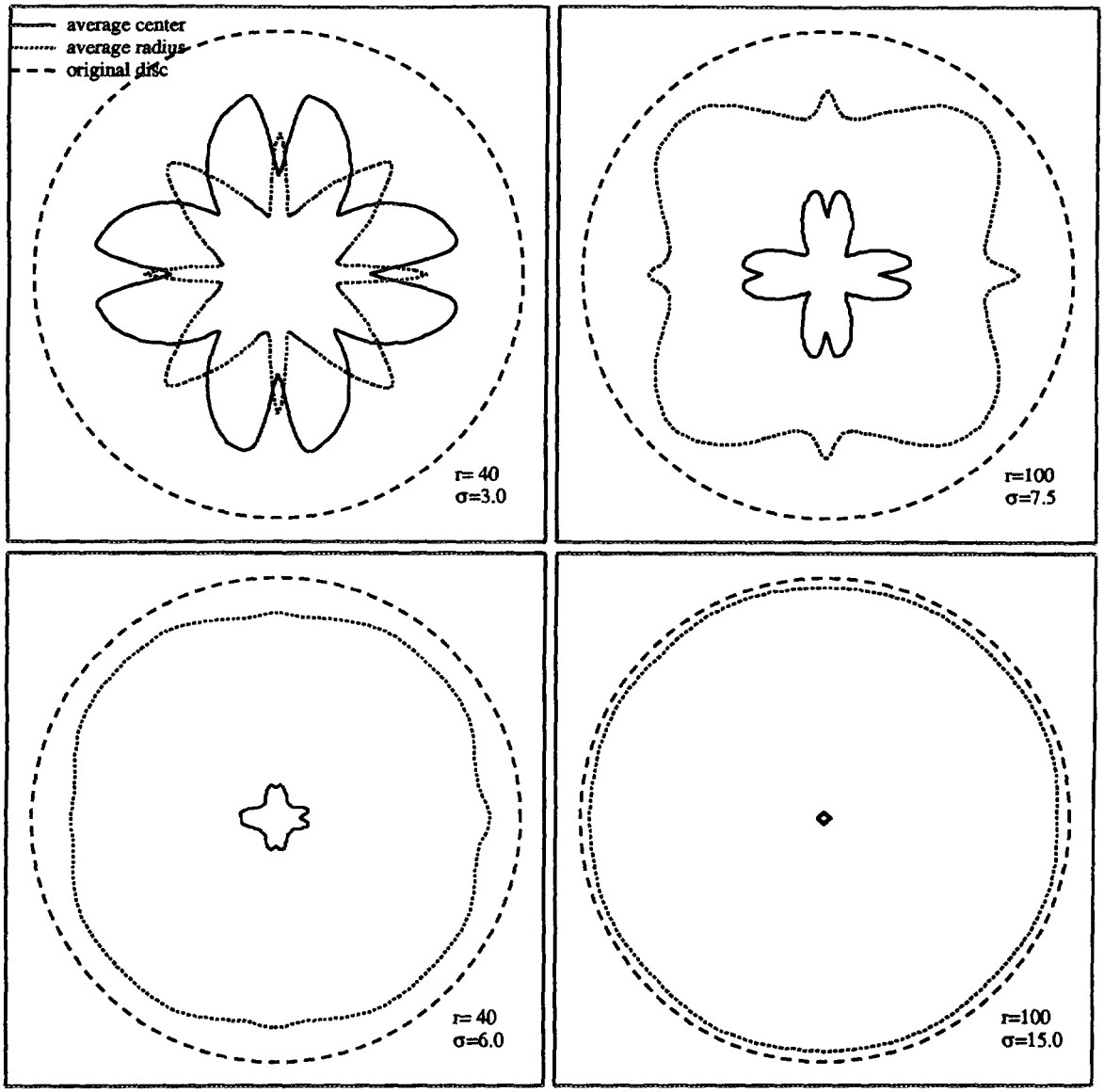

FIG. 11. This figure illustrates the performance of the arcfit method. In each figure three different curves are presented for some fixed value of r and σ. One curve denotes the average center found as a function of orientation θ. A second curve shows $(\bar{r}(\theta)\cos\theta, \bar{r}(\theta)\sin\theta)$, illustrating the average estimate $\bar{r}$ as function of orientation. The original disc is given as a reference curve. Note that the average center should be one point (the origin) and the average radius curve should be equal to the original disc.

is important to conceive of the task as an estimation problem in which measurements can be made of the error due to digitization.

Curvature estimation interferes in many ways with the spatial sampling of an object using a regular grid. As the grid is anisotropic it causes considerable difficulty in reliably estimating something essentially circular. A theoretical analysis of the problem revealed that the parameterization of the digital curve does not equal the isotropic parameterization of a continuous circle. This has a direct consequence on curvature estimation as it turns out that the digital coordinate sequences have discontinuous second derivatives. Those discontinuities introduce high frequencies in the Fourier transform of the coordinates, making smoothing unreliable.

We considered methods from literature to measure curvature from a digital curve. Methods are based on the three different formulations of curvature. We found five essentially different methods. Methods I, II, and III are all based on the formulation that curvature equals the change of orientation of the tangent. They differ in their method of orientation estimation and in the subsequent differentiation step. Method IV is based on the second derivative of the curve considered as a path. Finally, Method V is based on the osculating circle touching the curve. Methods were evaluated on the basis of theoretical

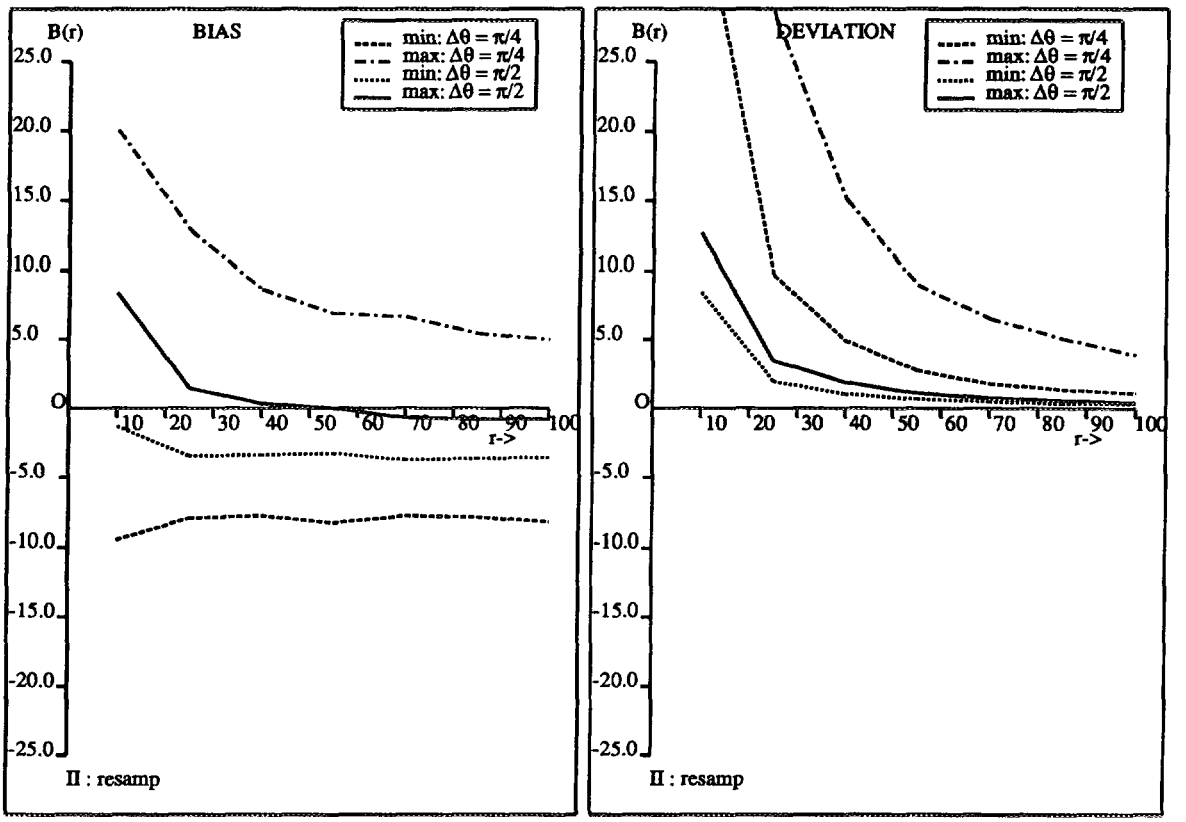

FIG. 12. In this figure, for the resampling method, the relation is shown between the resolution of the image and the minimum and maximum bias (a) and deviation (b) over all orientations. Measurements are shown for an arc of size $\Delta\theta = \pi/4$ and for an arc of size $\Delta\theta = \pi/2$. The radius r of the digital disc serves as the parameter of resolution. Note that the y-scale is different from previous figures.

and practical considerations. Monte Carlo experiments were performed to establish the bias and deviation in curvature estimation.

The experiments reveal that almost none of the existing methods to measure curvature performs well. We found, under typical circumstances, bias values of -27% to 511% and deviations between 0.4% and 819%.

Among the methods considered, Method III, based on fitting two straight lines [1, 12], has poor performance. Bias is -27% to 24%, where deviation is 5.8%–68%. The errors are a direct consequence of the method of differential filtering and of the arclength estimator used. Orientation estimation is reliable, however. Curvature estimates are improved by using a Gaussian differential kernel and a better arclength estimator [5].

Fitting a circular arc, Method V [20] is only suited for curvature estimation when arcs are large and of constant radius. Bias ranges from 6.2% to 511%. Deviations also reach large values (0.4%–819%).

The performance of the path-based method (IV) [9–11] is poor, as smoothing is hampered by the discontinuous derivatives of the coordinate functions. Another source of inaccuracy is the error due to truncation of the filter kernel at 3σ, combined, for large σ, with the error, due to shrinking of the curve. Bias values are 8.6%–492% with deviation 16%–378%. Truncation at 5σ is needed to obtain more reliable curvature estimates. With such sufficiently large kernels, this method is the method to use when the aim is to estimate curvature as well as the smoothed curve. For large σ, a correction for the shrinking effect should be applied [9].

Methods based on chain codes (Method I) [2, 13, 25] perform poorly as the anisotropy of the grid is not accounted for, resulting in a bias between -22% and 91% and deviations 0.6%–52%. Correction for this anisotropy results in resampling of the curve [15] followed by curvature estimation from the resampled chain codes (Method II) [6]. This method has the best overall performance. The remaining bias is between -8.2% and 17% and deviation ranges from 0.4% to 49%. It is recommended as the method to be used in practice, with scale parameter σ and resolution based on Fig. 12.

REFERENCES

1. I. M. Anderson and J. C. Bezdek, Curvature and deflection of discrete arcs: A theory based on the commutator of scatter matrix

pairs and its application to vertex detection in planar shape data, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 1984, 27–40.

2. H. Asada and M. Brady, The curvature primal sketch, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 1986, 2–14.
3. P. Craven and G. Wahba, Smoothing noisy data with spline functions—Estimating the correct degree of smoothing by the method of generalized cross-validation, *Numer. Math.* **31**, 1979, 377–403.
4. L. Dorst and A. W. M. Smeulders, Best linear unbiased estimators for properties of digitized straight lines, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 1986, 276–282.
5. L. Dorst and A. W. M. Smeulders, Length estimators for digitized contours, *Comput. Vision Graphics Image Process.* **40**, 1987, 311–333.
6. J. S. Duncan, F. Lee, A. W. M. Smeulders, and B. L. Zaret, A bending energy model for measurement of cardiac shape deformity, *IEEE Trans. Med. Imaging* **10**(3), 1991, 307–320.
7. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active contour models, *Int. J. Comput. Vision* **1**, 1988, 321–331.
8. U. M. Landau, Estimation of a circular arc center and its radius, *Comput. Vision Graphics Image Process.* **38**, 1987, 317–326.
9. D. G. Lowe, Organization of smooth image curves at multiple scales, in *Proceedings, Second International Conference on Computer Vision, Tampa, 1988*, pp. 558–567.
10. G. Medioni and Y. Yasumoto, Corner detection and curve representation using cubic B-splines, *Comput. Vision Graphics Image Process.* **39**, 1987, 267–278.
11. F. Mokhtarian and A. Mackworth, Scale-based description and recognition of planar curves and two-dimensional shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 1986, 34–43.
12. L. O'Gorman, An analysis of feature detectability from curvature estimation, *Proceedings, IEEE Conf. Computer Vision and Pattern Recognition, Ann Arbor, MI, 1988*, pp. 235–240.
13. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed., Academic Press, New York, 1982.
14. B. Shahraray and D. J. Anderson, Optimal estimation of contour properties by cross-validated regularization, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(6), 1989, 600–610.
15. B. Shahraray and D. J. Anderson, Uniform resampling of digitized contours, *IEEE Trans. Pattern Anal. Mach. Intell.* **7**, 1985, 674–681.
16. A. W. M. Smeulders and M. Worring, Accurate measurement of shape at low resolution, in *Pattern Recognition and Artificial Intelligence*, Vol. 7, *Machine Intelligence and Pattern Recognition* (L. N. Kanal and E. S. Gelsema, Eds.), pp. 91–102, North-Holland, Amsterdam, 1988.
17. L. H. Staib and J. S. Duncan, Boundary finding with parametrically deformable models, *IEEE Trans. Pattern Anal. Mach. Intell.* to appear.
18. D. J. Struik, *Lectures on Classical Differential Geometry*, Dover, New York, 1984.
19. C. Teh and R. T. Chin, On the detection of dominant points on digital curves, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(8), 1989, 859–871.
20. S. M. Thomas and Y. T. Chan, A simple approach for the estimation of circular arc center and its radius, *Comput. Vision Graphics Image Process.* **45**, 1989.
21. V. Torre and T. A. Poggio, On edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(2), 1986, 147–163.
22. M. Worring and A. W. M. Smeulders, A lower bound on the precision of curvature estimation on a regular grid, in preparation.
23. M. Worring and A. W. M. Smeulders, The accuracy and precision of curvature estimation methods, in *Proceedings, ICPR, Image Speech and Signal Analysis, 1992*.
24. M. Worring, L. H. Staib, and J. S. Duncan, *Using Smoothness and Directional Information in Fourier Parameterized Boundary Extraction*, Technical Report 92-1, Yale University, Diagnostic Imaging, 1992.
25. I. T. Young, J. E. Walker, and J. E. Bowie, An analysis technique for biological shape I, *Inform. and Control* **25**, 1974, 357–370.