

Research Article

Robust and Accurate Curvature Estimation Using Adaptive Line Integrals

Wei-Yang Lin,¹ Yen-Lin Chiu,² Kerry R. Widder,³ Yu Hen Hu,³ and Nigel Boston³

¹Department of CSIE, National Chung Cheng University, Min-Hsiung, Chia-Yi 62102, Taiwan

²Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Yang-Mei, Taoyuan 32601, Taiwan

³Department of ECE, University of Wisconsin-Madison, Madison, WI 53706, USA

Correspondence should be addressed to Wei-Yang Lin, wylin@cs.ccu.edu.tw

Received 18 May 2010; Accepted 4 August 2010

Academic Editor: A. Enis Cetin

Copyright © 2010 Wei-Yang Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The task of curvature estimation from discrete sampling points along a curve is investigated. A novel curvature estimation algorithm based on performing line integrals over an adaptive data window is proposed. The use of line integrals makes the proposed approach inherently robust to noise. Furthermore, the accuracy of curvature estimation is significantly improved by using wild bootstrapping to adaptively adjusting the data window for line integral. Compared to existing approaches, this new method promises enhanced performance, in terms of both robustness and accuracy, as well as low computation cost. A number of numerical examples using synthetic noisy and noiseless data clearly demonstrated the advantages of this proposed method over state-of-the-art curvature estimation algorithms.

1. Introduction

Curvature is a widely used invariant feature in pattern classification and computer vision applications. Examples include contour matching, contour segmentation, image registration, feature detection, object recognition, and so forth. Since curvature is defined by a function of higher-order derivatives of a given curve, the numerically estimated curvature feature is susceptible to noise and quantization error. Previously, a number of approaches such as curve/surface fitting [1–5], derivative of tangent angle [6, 7], and tensor of curvature [8–11] have been proposed with moderate effectiveness. However, an accurate and robust curvature estimation method is still very much desired.

Recently, the integral invariants [12–14] have begun to draw significant attention from the pattern recognition community due to their robustness to noise. These approaches have been shown as promising alternatives for extracting geometrical properties from discrete data. While curvature is just a special instance of invariant features under the rigid transformations (composition of rotations and translations), it is arguably the most widely used one in computer vision applications.

In this paper, we propose a novel curvature estimator based on evaluating line integrals over a curve. Since our method does not require derivative evaluations, it is inherently robust with respect to sampling and quantization noise. In contrast to the previous efforts, we are interested here in the *line integral*. It should be noted that the strategy presented by Pottmann et al. [14] can be trivially changed to compute curvature on curves. However, the resultant curvature estimate requires surface integrals taken over local neighborhoods. Compared with surface integral (also known as double integral), the line-integral formulation for curvature estimation has a reduced computational complexity in general. We will further discuss the complexity of numerical integration in Section 3.

Our method is also a significant improvement over the previously reported work [14] in terms of estimation accuracy. This is because the earlier work evaluates integrals over a user-defined, fixed-size window surrounding the point where curvature is to be evaluated. Depending on the sharpness of the curvature, the window size may be too large or too small. An over-sized window would dilute the distinct curvature feature by incorporating irrelevant points on the curve into the integral. An under-sized window,

on the other hand, would be less robust to noise and quantization errors.

In this proposed curvature estimation algorithm, we evaluate line integrals over a window whose size is adaptively determined using the wild bootstrap procedure [15]. As such, the size of the data window will be commensurate to the sharpness of the curvature to be estimated, and the resulting accuracy is expected to be significantly improved. The performance advantage of this proposed adaptive window curvature estimation algorithm has been examined analytically, and has been validated using several numerical experiments.

The rest of this paper is organized as follows. Section 2 provides a brief review on the related work. In Section 3, the curvature estimation method based on line integrals is introduced. We subsequently formulate the problem of choosing an optimal window size and derive an adaptive curvature estimator in Section 4. In Section 5, we provide experimental results to show the robustness and accuracy of the proposed method. Comparisons with existing curvature estimation methods are also included. Finally, we make concluding remarks and discuss future works in Section 6.

2. Related Work

Due to the needs of many practical applications, extensive research has been conducted on the problem of curvature estimation. In a real-world application, data are often given in discrete values sampled from an object. Hence, one is required to estimate curvature or principal curvatures from discrete values. Flynn and Jain [4] report an empirical study on five curvature estimation methods available at that time. Their study's main conclusion is that the estimated curvatures are extremely sensitive to quantization noise and multiple smoothings are required to get stable estimates. Trucco and Fisher [16] have similar conclusion. Worring and Smeulders [7] identify five essentially different methods for measuring curvature on digital curves. By performing a theoretical analysis, they conclude that none of these methods is robust and applicable for all curve types. Magid et al. [17] provide a comparison of four different approaches for curvature estimation on triangular meshes. Their work manifests the best algorithm suited for estimating Gaussian and mean curvatures.

In the following sections, we will discuss different kinds of curvature estimation methods known in the literature. Also, we will review some related work in integral invariants and adaptive window selection.

2.1. Derivative of the Tangent Angle. The approaches based on the derivative of tangent can be found in [6, 18–20]. Given a point on a curve, the orientation of its tangent vector is first estimated and then curvature is calculated by Gaussian differential filtering. This kind of methods are preferable when computational efficiency is of primary concern. The problem associated with these approaches is that estimating tangent vector is highly noise-sensitive

and thus the estimated curvature is unstable in real world applications.

2.2. Radius of the Osculating Circle. The definition of osculating circle leads to algorithms which fit a circular arc to discrete points [2, 3, 21]. The curvature is estimated by computing the reciprocal of the radius of an osculating circle. An experimental evaluation of this approach is presented in the classical paper by Worring and Smeulders [7]. The results reveal that reliable estimates can only be expected from arcs which are relatively large and of constant radius.

2.3. Local Surface Fitting. As the acquisition and use of 3D data become more widespread, a number of methods have been proposed for estimating principal curvatures on a surface. Principal curvatures provide unique view-point invariant shape descriptors. One way to estimate principal curvatures is to perform surface fitting. A local fitting function is constructed and then curvature can be calculated analytically from the fitting function. The popular fitting methods include paraboloid fitting [22–24] and quadratic fitting [1, 25–27]. Apart from these fitting techniques, other methods have been proposed, such as higher-order fitting [5, 28] and circular fitting [29, 30]. Cazals and Pouget [5] perform a polynomial fitting and show that the estimated curvatures converge to the true ones in the case of a general smooth surface. A comparison of local surface geometry estimation methods can be found in [31].

The paper written by Flynn and Jain [4] reports an empirical evaluation on three commonly used fitting techniques. They conclude that reliable results cannot be obtained in the presence of noise and quantization error.

2.4. The Tensor of Curvature. The tensor of curvature has lately attracted some attention [8–11, 32]. It has been shown as a promising alternative for estimating principal curvatures and directions. This approach is first introduced by Taubin [8], followed by the algorithms which attempt to improve accuracy by tensor voting [9–11, 32]. Page et al. [9] present a voting method called normal voting for robust curvature estimation, which is similar to [10, 32]. Recently, Tong and Tang [11] propose a three-pass tensor voting algorithm with improved robustness and accuracy.

2.5. Integral Invariants. Recently, there is a trend on so-called integral invariants which reduce the noise-induced fluctuations by performing integrations [12, 13]. Such integral invariants possess many desirable properties for practical applications, such as locality (which preserves local variations of a shape), inherent robustness to noise (due to integration), and allowing multiresolution analysis (by specifying the interval of integration). In [14], the authors present an integration-based technique for computing principal curvatures and directions from a discrete surface. The proposed method is largely inspired by both Manay et al. [13] and Pottmann et al. [14], in which they use a convolution approach to calculate an integral. In this paper, we investigate

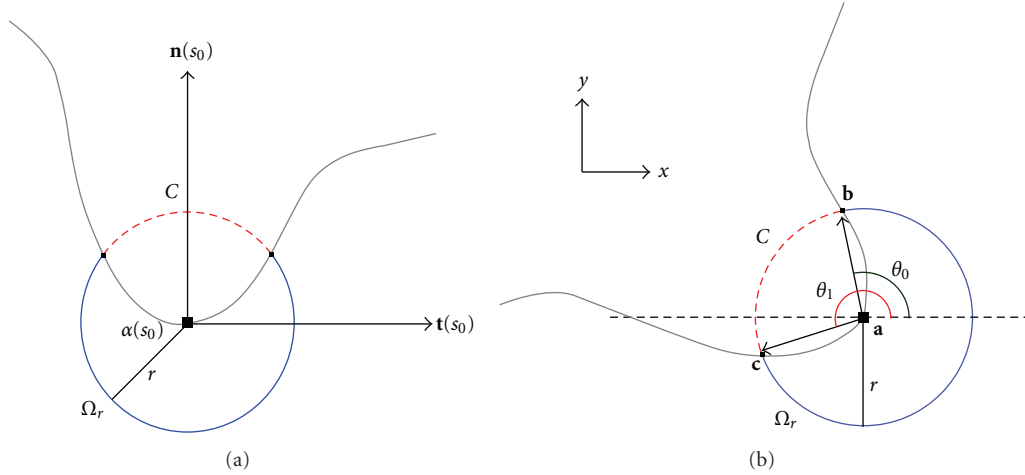


FIGURE 1: (a) For a point (the black square dot) on a curve $g(x)$ (the gray line), we draw a circle Ω_r centered at that point. The integral region $C = \{(x, y) \mid x^2 + y^2 = r^2, y \geq g(x)\}$ is denoted by red dashed line. It is convenient to write the equation of the curve, in the neighborhood of $\alpha(s_0)$, using $\mathbf{t}(s_0)$ and $\mathbf{n}(s_0)$ as a coordinate frame. (b) After obtaining θ_0 and θ_1 , the line integrals can be easily computed. It does not matter which coordinate system we use for computing θ_0 and θ_1 . One can always obtain a curvature estimate by performing eigenvalue decomposition.

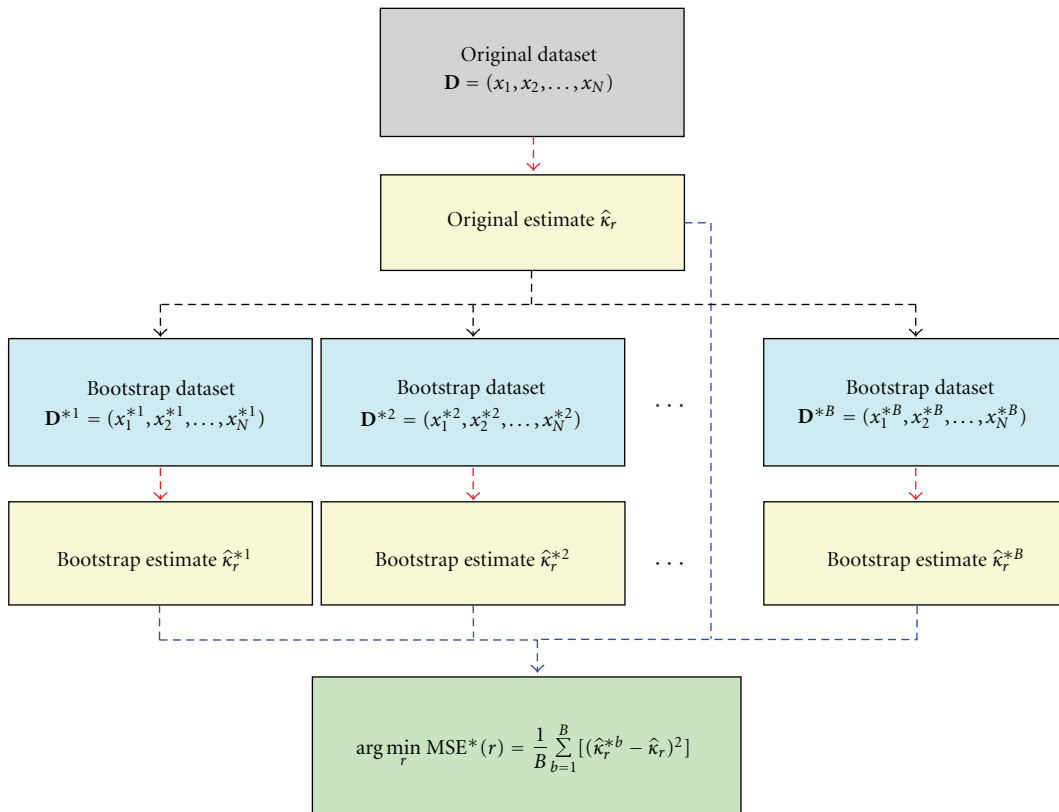


FIGURE 2: Block diagram of the radius selection algorithm using bootstrap method.

avoiding the convolution with polynomial complexity by instead using the one with constant complexity.

2.6. Adaptive Window Selection. The curvature estimation algorithms mentioned above have the shortcoming of using

a fixed window size. On one hand, if a large window is selected, some fine details on a shape will be smoothed out. On the other hand, if a small window is utilized, the effect of discretization and noise will be salient and the resultant estimate will have a large variance. To mitigate this

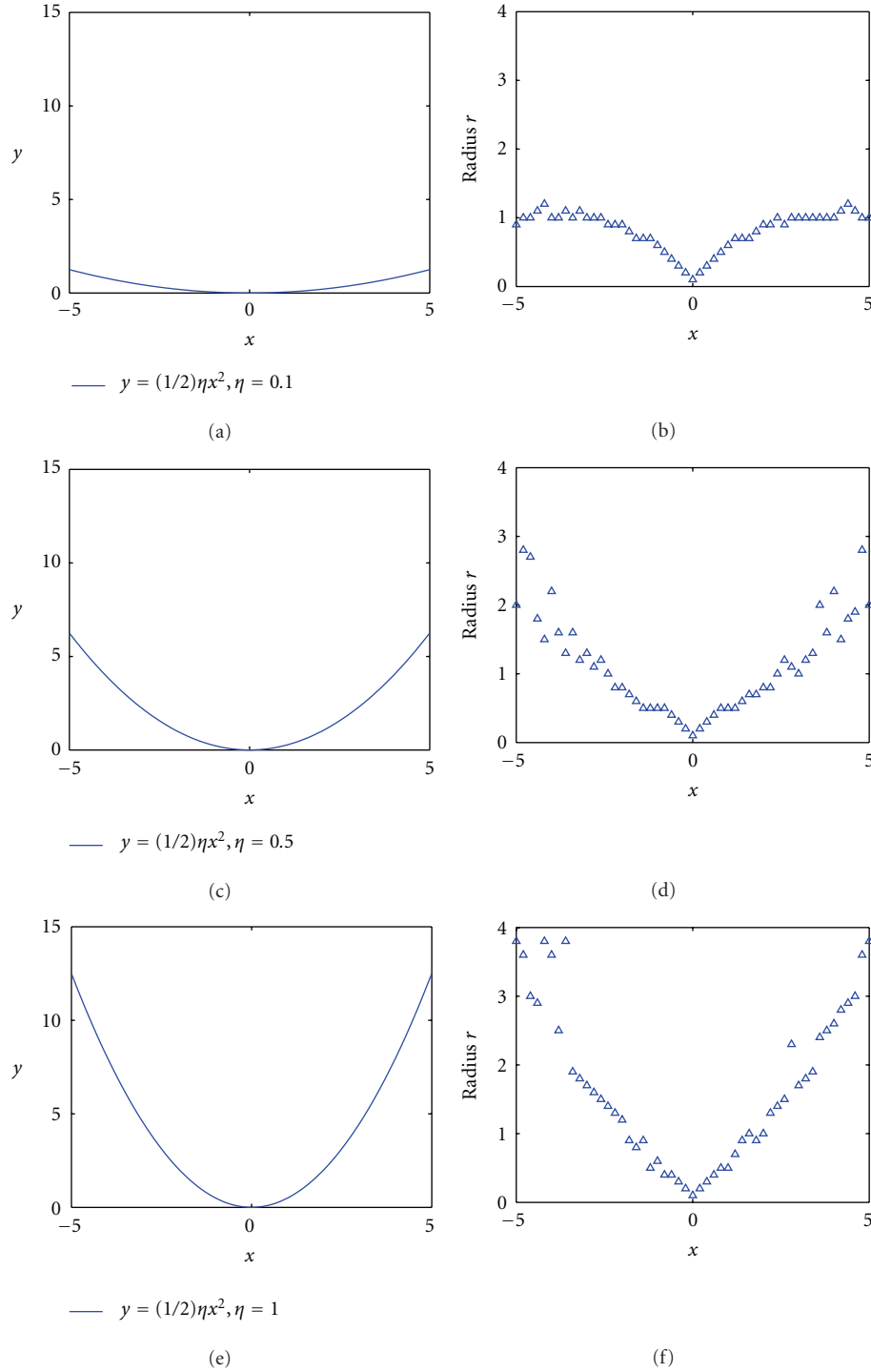


FIGURE 3: The proposed adaptive curvature estimator is applied to the curves depicted in (a), (c), and (e). The resultant radii of Ω_r are shown in (b), (d), and (f), respectively.

fundamental difficulty in curvature estimation, a window size must be determined adaptively depending on local characteristics.

A number of publications concerning the issue of adaptive window selection have appeared in the last two decades [33–37]. In the dominant point detection algorithms

[33, 35, 36], it is important to select a proper window for estimating curvature. Teh and Chin [33] use the ratio of perpendicular distance and the chord length to determine the size of a window. B. K. Ray and K. S. Ray [35] introduce a new measurement, namely, k -cosine, to decide a window adaptively based on some local properties of a curve. Wu [36]

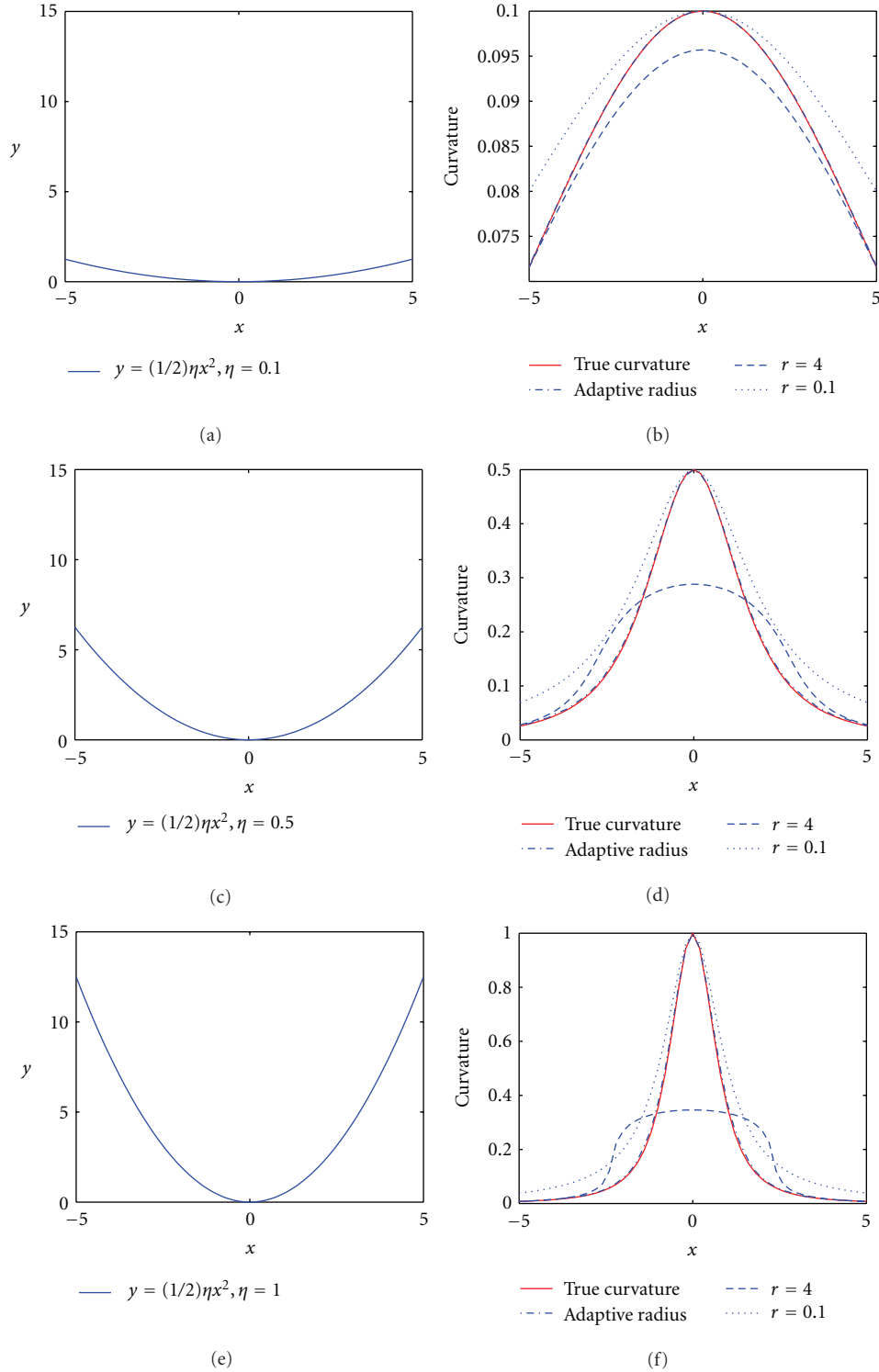


FIGURE 4: True curvatures and estimated curvatures of the curves in (a), (c), and (e) are shown in (b), (d), and (f), respectively. The curvature estimates are obtained by an adaptive radius and fixed radii.

proposes a simple measurement which utilizes an adaptive bending value to select the optimal window.

Recently, the bootstrap methods [38] have been applied with great success to a variety of adaptive window selection

problems. Foster and Zychaluk [37] present an algorithm for estimating biological transducer functions. They utilize a local fitting with bootstrap window selection to overcome the problems associated with traditional polynomial regression.

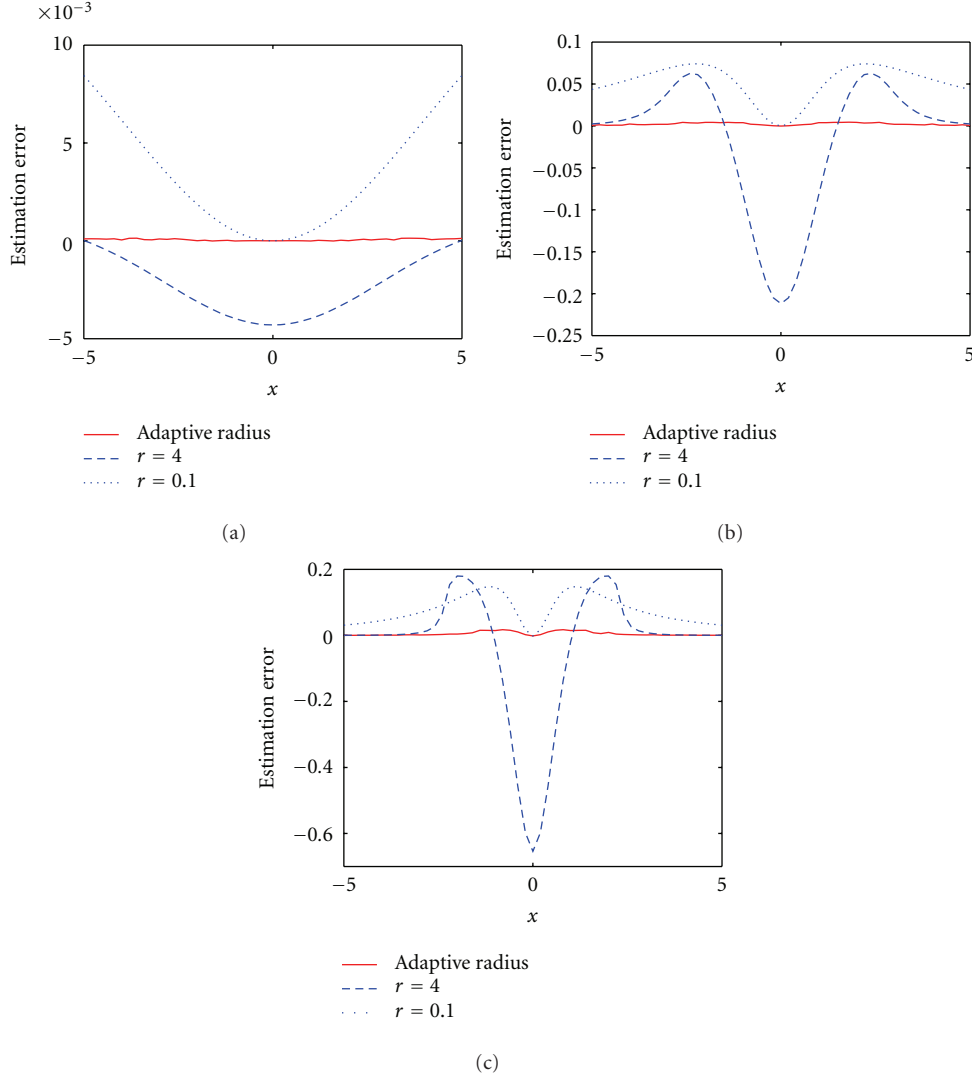


FIGURE 5: The estimation errors in Figures 4(b), 4(d), and 4(f) are shown in (a), (b), and (c), respectively.

Inspired by their work, we develop an adaptive curvature estimation algorithm based on the wild bootstrap method [15, 39]. We will elaborate the associated window selection algorithm in Section 4.

3. Curvature Estimation by Line Integrals

In this section, we introduce the approach for estimating curvature along a planar curve by using line integrals.

First, we briefly review some important results in differential geometry. Interested readers may refer to [40] for more details. Let $\tau \subset \mathbb{R}$ be an interval and $\alpha : \tau \rightarrow \mathbb{R}^2$ be a curve parameterized by arc length $s \in \tau$. To proceed with local analysis, it is necessary to add the assumption that the derivative $\alpha'(s)$ always exists. We interpret $\alpha(s)$ as the trajectory of a particle moving in a 2-dimensional space. The moving plane determined by the unit tangent and normal vectors, $\mathbf{t}(s)$ and $\mathbf{n}(s)$, is called the osculating plane at $\alpha(s)$.

In analyzing the local properties of a point on a curve, it is convenient to work with the coordinate system associated with that point. Hence, one can write the equation of a curve, in the neighborhood of $\alpha(s_0)$, by using $\mathbf{t}(s_0)$ and $\mathbf{n}(s_0)$ as a coordinate frame. In particular, $\mathbf{t}(s_0)$ is the x -axis and $\mathbf{n}(s_0)$ is the y -axis. The Taylor series expansion of the curve in the neighborhood of $\alpha(s_0)$, denoted by $g(x)$, with respect to the local coordinate frame centered at $\alpha(s_0)$, is given by

$$y = g(x) = g(0) + xg'(0) + \frac{x^2}{2}g''(0) + \rho, \quad (1)$$

where ρ is the remainder. Since $g(0) = 0$, $g'(0) = 0$, and $g''(0)$ is the curvature at $\alpha(s_0)$, we obtain that $g(x) \approx (\kappa/2)x^2$, where κ denotes the curvature at $\alpha(s_0)$. For a point on a curve, let Ω_r denote a circle with center at that point and radius r . Then, we can perform the line integral of an arbitrary function f along C ,

$$I(f) = \int_C f(x, y) d\ell, \quad (2)$$

where $C = \{(x, y) \mid x^2 + y^2 = r^2, y \geq g(x)\}$ and $d\ell$ is the arc length element; in other words, C is the portion of the circle Ω_r that is above $g(x)$. An example of a circle and the corresponding integral region C is shown in Figure 1(a). The line integral $I(f)$ can be approximated by

$$I(f) \approx \hat{I}(f) = \int_{\Omega_r^+} f(x, y) d\ell - \int_0^{(1/2)\kappa r^2} f(r, y) dy - \int_0^{(1/2)\kappa r^2} f(-r, y) dy, \quad (3)$$

where Ω_r^+ denotes the upper half of Ω_r , that is, $\Omega_r^+ = \{(x, y) \mid x^2 + y^2 = r^2, y \geq 0\}$. In (3), we first perform line integral on the upper half of Ω_r (the first term) and then subtract the line integrals on the portions of Ω_r that are between $g(x)$ and x -axis (the second and third terms). We utilize two straight lines to approximate the portions of Ω_r bounded by $g(x)$ and x -axis.

Let $\mathbf{x} = [x \ y]^T$, the covariance matrix Σ of the region C is given by

$$\Sigma(C) = \int_C (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T d\ell = \int_C \mathbf{x}\mathbf{x}^T d\ell - L(C)\mathbf{m}\mathbf{m}^T, \quad (4)$$

where $L(C) = \int_C d\ell$ and $\mathbf{m} = (1/L) \int_C \mathbf{x} d\ell$ denote the length and the barycenter of C , respectively. Because the region Ω_r^+ is symmetric, the line integral $\hat{I}(f)$ is equal to zero for any odd function f . Hence, we have $I(x) \approx \hat{I}(x) = 0$ and $I(xy) \approx \hat{I}(xy) = 0$. By using (3), we can then obtain

$$\begin{aligned} I(x^2) &\approx \int_{\Omega_r^+} x^2 d\ell - 2 \int_0^{(1/2)\kappa r^2} r^2 dy = \frac{\pi}{2} r^3 - \kappa r^4, \\ I(y^2) &\approx \int_{\Omega_r^+} y^2 d\ell - 2 \int_0^{(1/2)\kappa r^2} y^2 dy = \frac{\pi}{2} r^3 - \frac{\kappa^3}{12} r^6, \\ I(y) &\approx \int_{\Omega_r^+} y d\ell - 2 \int_0^{(1/2)\kappa r^2} y dy = 2r^2 - \frac{\kappa^2}{4} r^4, \\ L = I(1) &\approx \int_{\Omega_r^+} d\ell - 2 \int_0^{(1/2)\kappa r^2} dy = \pi r - \kappa r^2. \end{aligned} \quad (5)$$

Therefore, the covariance matrix $\Sigma(C)$ can be approximated by

$$\begin{aligned} \Sigma(C) &\approx \begin{bmatrix} \frac{\pi}{2} r^3 - \kappa r^4 & 0 \\ 0 & \frac{\pi}{2} r^3 - \frac{\kappa^3}{12} r^6 \end{bmatrix} \\ &\quad - \frac{1}{\pi r - \kappa r^2} \begin{bmatrix} 0 & 0 \\ 0 & \left(2r^2 - \frac{\kappa^2}{4} r^4\right)^2 \end{bmatrix}. \end{aligned} \quad (6)$$

From (6), we can obtain the following relationship:

$$\Sigma_{1,1} \approx \frac{\pi}{2} r^3 - \kappa r^4 \Rightarrow \kappa \approx \frac{\pi}{2r} - \frac{\Sigma_{1,1}}{r^4}. \quad (7)$$

So, curvature κ can be estimated by performing the principal component analysis on the region C . In a real-world application, it does not matter which coordinate system is used for computing a covariance matrix. One can conduct the eigenvalue decomposition of $\Sigma(C)$ and then obtain a curvature estimate. The procedure for curvature estimation is as follows.

- (1) Let \mathbf{a} be a point on a curve. We draw a circle with radius r centered at \mathbf{a} . The intersections of the circle and the curve are denoted by \mathbf{b} and \mathbf{c} . The angle between the vector $\overrightarrow{\mathbf{ab}}$ and the x -axis is denoted by θ_0 . Similarly, θ_1 denotes the angle between the vector $\overrightarrow{\mathbf{ac}}$ and the x -axis. An example is shown in Figure 1(b).
- (2) Calculate the covariance matrix $\Sigma_a(C)$ associated with point \mathbf{a} . Following directly from (4), we have

$$\begin{aligned} \Sigma_a(C) &= \begin{bmatrix} I_a(x^2) & I_a(xy) \\ I_a(xy) & I_a(y^2) \end{bmatrix} \\ &\quad - \frac{1}{L_a(C)} \begin{bmatrix} I_a^2(x) & I_a(x)I_a(y) \\ I_a(x)I_a(y) & I_a^2(y) \end{bmatrix}. \end{aligned} \quad (8)$$

It is straightforward to show that the line integrals can be calculated as follows:

$$\begin{aligned} I_a(x^2) &= \frac{r^3}{2} [\theta_1 - \theta_0 + \sin \theta_1 \cos \theta_1 - \sin \theta_0 \cos \theta_0], \\ I_a(y^2) &= \frac{r^3}{2} [\theta_1 - \theta_0 - (\sin \theta_1 \cos \theta_1 - \sin \theta_0 \cos \theta_0)], \\ I_a(xy) &= \frac{r^3}{2} (\sin^2 \theta_1 - \sin^2 \theta_0), \\ I_a(x) &= r^2 (\sin \theta_1 - \sin \theta_0), \\ I_a(y) &= -r^2 (\cos \theta_1 - \cos \theta_0), \\ L_a(C) &= r(\theta_1 - \theta_0). \end{aligned} \quad (9)$$

- (3) The covariance matrix $\Sigma_a(C)$ can be factored as

$$\Sigma_a(C) = \mathbf{V} \mathbf{D} \mathbf{V}^T, \quad (10)$$

where $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2)$ contains the eigenvalues of $\Sigma_a(C)$ and $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2]$ contains the corresponding eigenvectors. Because $\Sigma_a(C)$ is real and symmetric, the eigenvectors \mathbf{v}_1 and \mathbf{v}_2 are orthogonal. Generally speaking, (10) shows the Singular Value Decomposition (SVD) and thus the diagonal elements of \mathbf{D} are also called the singular values of $\Sigma_a(C)$.

- (4) The unit tangent at \mathbf{a} , denoted by $\mathbf{t}(\mathbf{a})$, must be parallel to either \mathbf{v}_1 or \mathbf{v}_2 . If the eigenvector parallel to $\mathbf{t}(\mathbf{a})$ were identified, one could compute curvature by using the corresponding eigenvalue (see (7)). Here, we choose the eigenvalue by comparing signs of inner products $\overrightarrow{\mathbf{ab}} \cdot \mathbf{v}_i$ and $\overrightarrow{\mathbf{ac}} \cdot \mathbf{v}_i$. If \mathbf{v}_i were parallel to $\mathbf{t}(\mathbf{a})$, the signs of $\overrightarrow{\mathbf{ab}} \cdot \mathbf{v}_i$ and $\overrightarrow{\mathbf{ac}} \cdot \mathbf{v}_i$ must be different.

One can use either \mathbf{v}_1 or \mathbf{v}_2 . Pseudocode for computing curvature utilizing \mathbf{v}_1 is shown below.

$$\begin{aligned} & \text{if } \text{sign}(\vec{ab} \cdot \mathbf{v}_1) \neq \text{sign}(\vec{ac} \cdot \mathbf{v}_1) \\ & \text{else } \kappa \approx \frac{\pi}{2r} - \frac{\lambda_1}{r^4} \\ & \quad \kappa \approx \frac{\pi}{2r} - \frac{\lambda_2}{r^4}. \end{aligned} \quad (11)$$

Note that the numerical integration is typically computed by convolution in the previous work [13, 14]. For example, when evaluating the area integral invariant [13] of a particular point on a curve, the standard convolution algorithm has a quadratic computational complexity. With the help of the convolution theorem and the Fast Fourier Transform (FFT), the complexity of convolution can be significantly reduced [14]. However, the running time required by the FFT is $\mathcal{O}(N^2 \log N)$, where N^2 equals the number of sampling points in an integral region. Compared with the earlier methods [13, 14], the complexities of the integrals in (8) are constant and hence our method is computationally more efficient.

4. Adaptive Radius Selection

A critical issue in curvature estimation by line integrals lies in selecting an appropriate circle. The circle Ω_r must be large enough to include enough data points for reliable estimation, but small enough to avoid the effect of oversmoothing. For this reason, the radius of a circle must be selected adaptively, based on local shapes of a curve. In this section, we will first formulate the problem of selecting an optimal radius and then present an adaptive radius selection algorithm.

Intuitively, an optimal radius can be obtained by minimizing the difference between the estimated curvature $\hat{\kappa}_r$, based on the data within radius r , to its true value κ . A common way to quantify the difference between $\hat{\kappa}_r$ and κ is to compute the Mean Squared Error (MSE) as a function of r , that is,

$$\text{MSE}(r) = E[(\hat{\kappa}_r - \kappa)^2], \quad (12)$$

where E is the expectation (the value that could be obtained if the distribution of $\hat{\kappa}_r$ were available). However, the minimizer of $\text{MSE}(r)$ cannot be found in practice since it involves an unknown value κ .

The bootstrap method [38], which has been extensively analyzed in the literature, provides an effective means for overcoming such a difficulty. In (12), one can simply replace the unknown value κ with the estimate obtained from a given dataset, then replace the original estimate $\hat{\kappa}_r$ with the estimates computed from bootstrap datasets. Therefore, the optimal radius can be determined by

$$r_{\text{opt}} = \arg \min_r \text{MSE}^*(r) = \arg \min_r E^*[(\hat{\kappa}_r^* - \hat{\kappa}_r)^2], \quad (13)$$

where the asterisks denote that the statistics are obtained from bootstrap samples.

The conceptual block diagram of the radius selection algorithm using bootstrap method is shown in Figure 2 and the detailed steps are described below.

- (1) Given a point (x_0, y_0) on a curve, we draw an initial circle of radius r .
- (2) By using the estimator described in Section 3, the estimate $\hat{\kappa}_r$ is calculated from the neighboring points of (x_0, y_0) within radius r . In the rest of this paper, we will use $\mathcal{D} = \{(x_i, y_i) \mid i = 1, 2, \dots, N\}$ to denote the neighboring points of (x_0, y_0) within radius r .
- (3) The local shape around (x_0, y_0) can be modeled by

$$y_i = \frac{\hat{\kappa}_r}{2} x_i^2 + \varepsilon_i, \quad i = 1, 2, \dots, N, \quad (14)$$

where ε_i is called a modeling error or residual. Note that we use the moving plane described in Section 3 as our local coordinate system.

- (4) Generate wild bootstrap residuals ε_i^* from a two-point distribution [15]:

$$\varepsilon_i^* = \varepsilon_i \left(\frac{V_i}{\sqrt{2}} + \frac{V_i^2 - 1}{2} \right), \quad i = 1, 2, \dots, N, \quad (15)$$

where the V_i 's are independent standard normal random variables.

- (5) The wild bootstrap samples (x_i, y_i^*) are constructed by adding the bootstrap residuals ε_i^* :

$$y_i^* = \frac{\hat{\kappa}_r}{2} x_i^2 + \varepsilon_i^*. \quad (16)$$

We use $\mathcal{D}^* = \{(x_i, y_i^*) \mid i = 1, 2, \dots, N\}$ to denote a wild bootstrap dataset.

- (6) By repeating the third to the fifth steps, we can generate many wild bootstrap datasets, that is, $\mathcal{D}^{*1}, \mathcal{D}^{*2}, \dots, \mathcal{D}^{*B}$. The larger the number of wild bootstrap datasets, the more satisfactory the estimate of a statistic will be.
- (7) We can then obtain bootstrap estimates $\hat{\kappa}_r^{*1}, \hat{\kappa}_r^{*2}, \dots, \hat{\kappa}_r^{*B}$ from the wild bootstrap datasets $\mathcal{D}^{*1}, \mathcal{D}^{*2}, \dots, \mathcal{D}^{*B}$. The bootstrap estimate of the $\text{MSE}(r)$ is given by

$$\text{MSE}^*(r) = \frac{1}{B} \sum_{b=1}^B \left[(\hat{\kappa}_r^{*b} - \hat{\kappa}_r)^2 \right]. \quad (17)$$

- (8) The optimal radius is defined as the minimizer of (17), that is,

$$r_{\text{opt}} = \arg \min_r \frac{1}{B} \sum_{b=1}^B \left[(\hat{\kappa}_r^{*b} - \hat{\kappa}_r)^2 \right]. \quad (18)$$

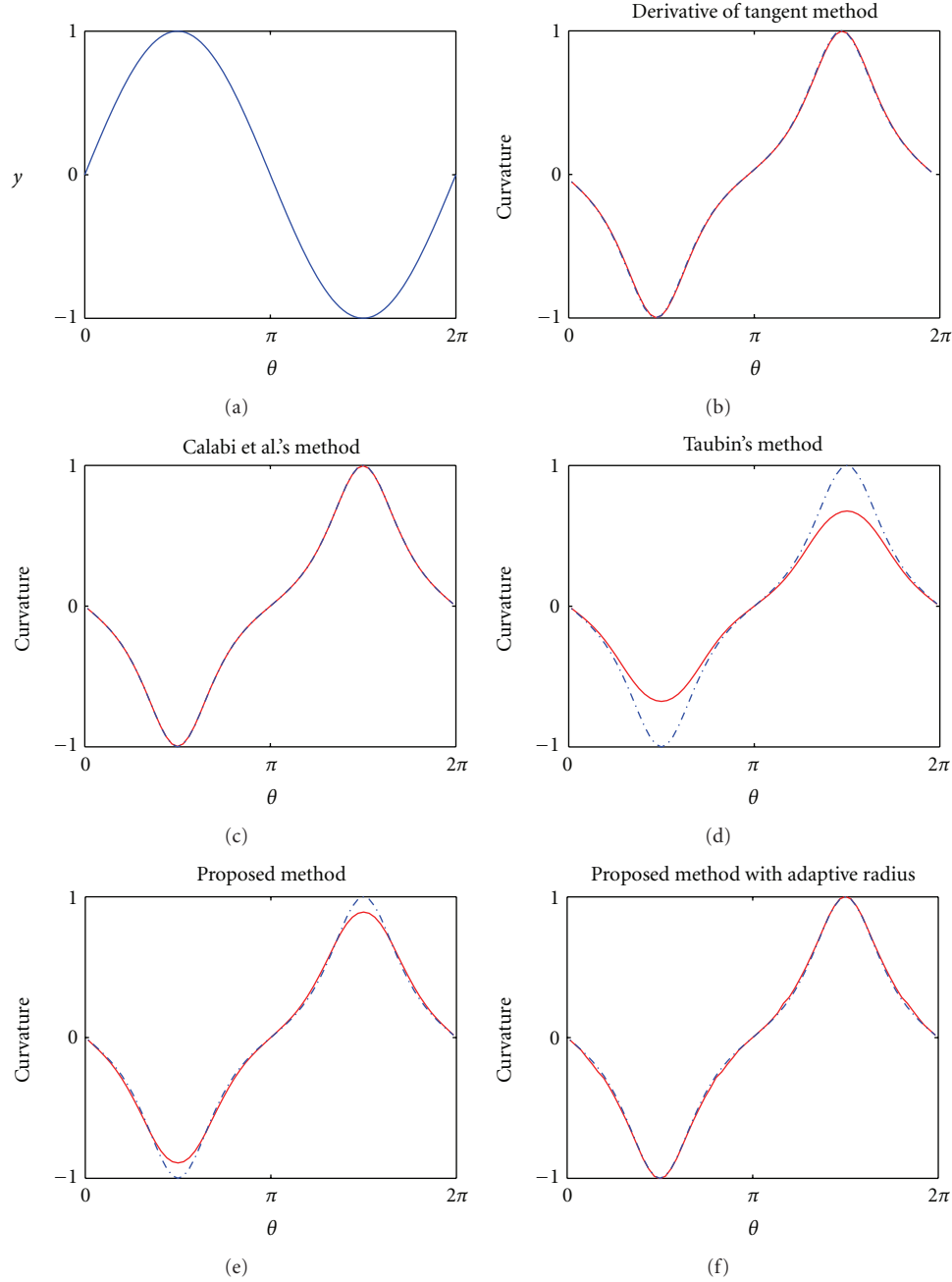


FIGURE 6: (a) A sinusoidal waveform, (b) curvature estimate obtained by derivative of tangent, (c) curvature estimate obtained by Calabi et al.'s algorithm, (d) curvature estimate obtained by Taubin's algorithm, (e) curvature estimate obtained by line integrals, and (f) curvature estimate obtained by line integrals with adaptive radius. Notice that a dashed blue line denotes the true curvature.

5. Experiments and Results

We conduct several experiments to evaluate the performance of the proposed adaptive curvature estimator. In Section 5.1, we demonstrate how the radius of the estimator changes with respect to local contour geometry. In Section 5.2, the experiments are conducted to verify whether the adaptivity provides an improved estimation accuracy. And, the robustness of the proposed method is experimentally validated in Section 5.3.

5.1. Qualitative Experiments. These experiments are intended to qualitatively verify the behavior of selecting an optimal radius. The curves $\{y = (1/2)\eta x^2 \mid x \in [-5, 5], \eta = 0.1, 0.5, 1\}$ are utilized as test subjects in the experiments. The sampling points along a curve are generated by performing sampling uniformly along the x -axis. The radius of the proposed adaptive curvature estimator ranges from 0.1 to 4 with the step size of 0.1 . Figure 3 shows the adaptively varying radii obtained by our method. We can see that the radius is relatively small near the point at $x = 0$ and become

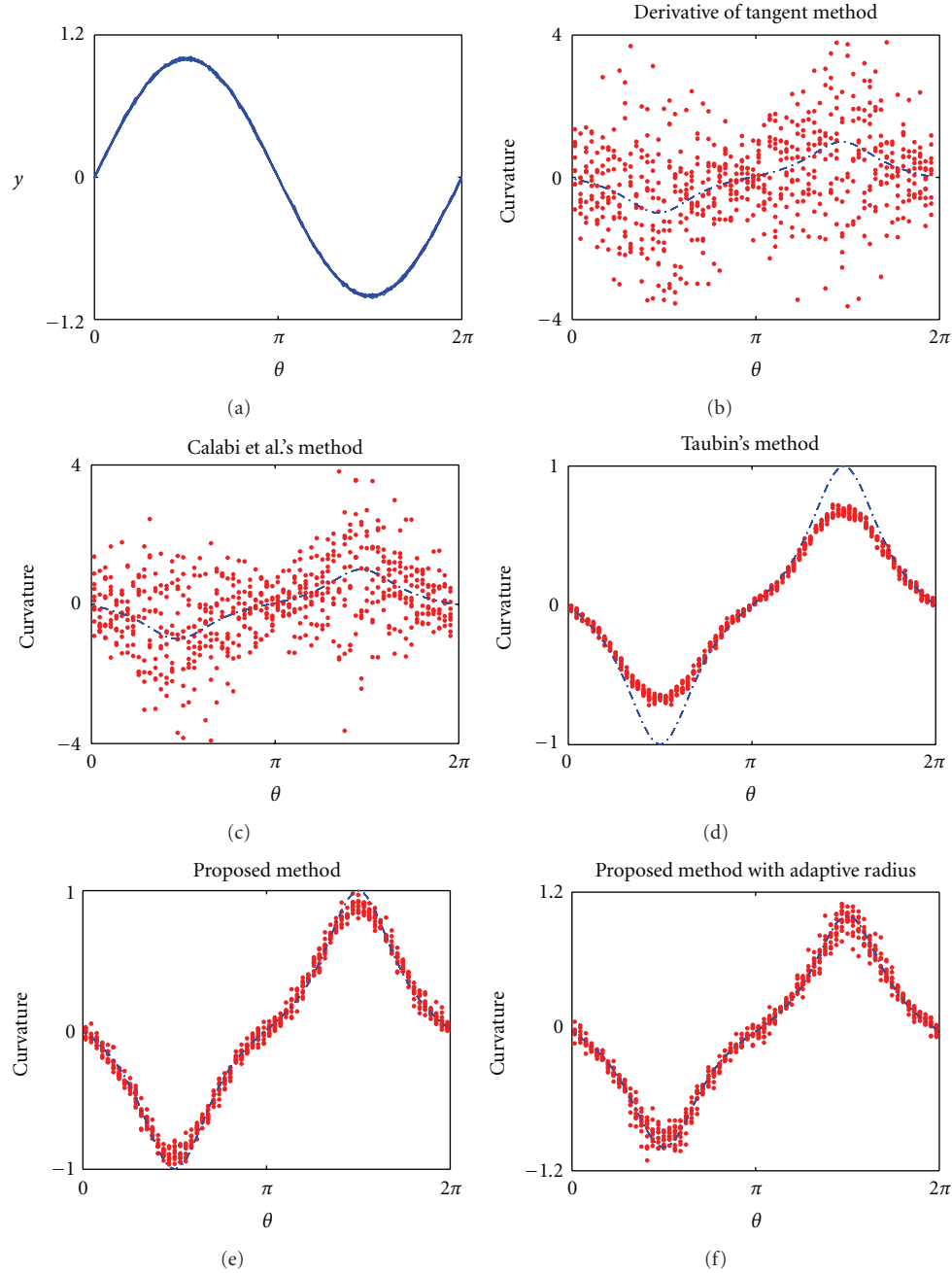


FIGURE 7: Trial-to-trial variability in curvature estimates. The data consist of 10 trials. (a) sinusoidal waveforms with additive Gaussian noise, (b) curvature estimate obtained by derivative of tangent, (c) curvature estimate obtained by Calabi et al.'s algorithm, (d) curvature estimate obtained by Taubin's algorithm, (e) curvature estimate obtained by line integrals, and (f) curvature estimate obtained by line integrals with adaptive radius. Notice that these figures have different ranges in vertical coordinate because some methods yield noisy results. The true curvature is denoted by a dashed blue line.

larger as $|x|$ is increasing. This phenomenon corresponds to our expectation that a smaller radius should be chosen at a point with high curvature so that smoothing effect can be reduced. In a low-curvature area, a larger radius should be selected so that a more reliable estimate can be obtained. Since the behavior is in accordance with the favorable expectation, the remaining issue is whether the adaptively selected radius indeed improves estimation accuracy. In

the following section, we will perform an experimental validation on this issue.

5.2. Quantitative Experiments. In the quantitative analysis, the curvature estimate obtained by adaptive radius is compared against the true curvature, and the estimate obtained by fixed radii. In Figure 4, it can be seen that the curvature estimator with a fixed undersize radius will be accurate at

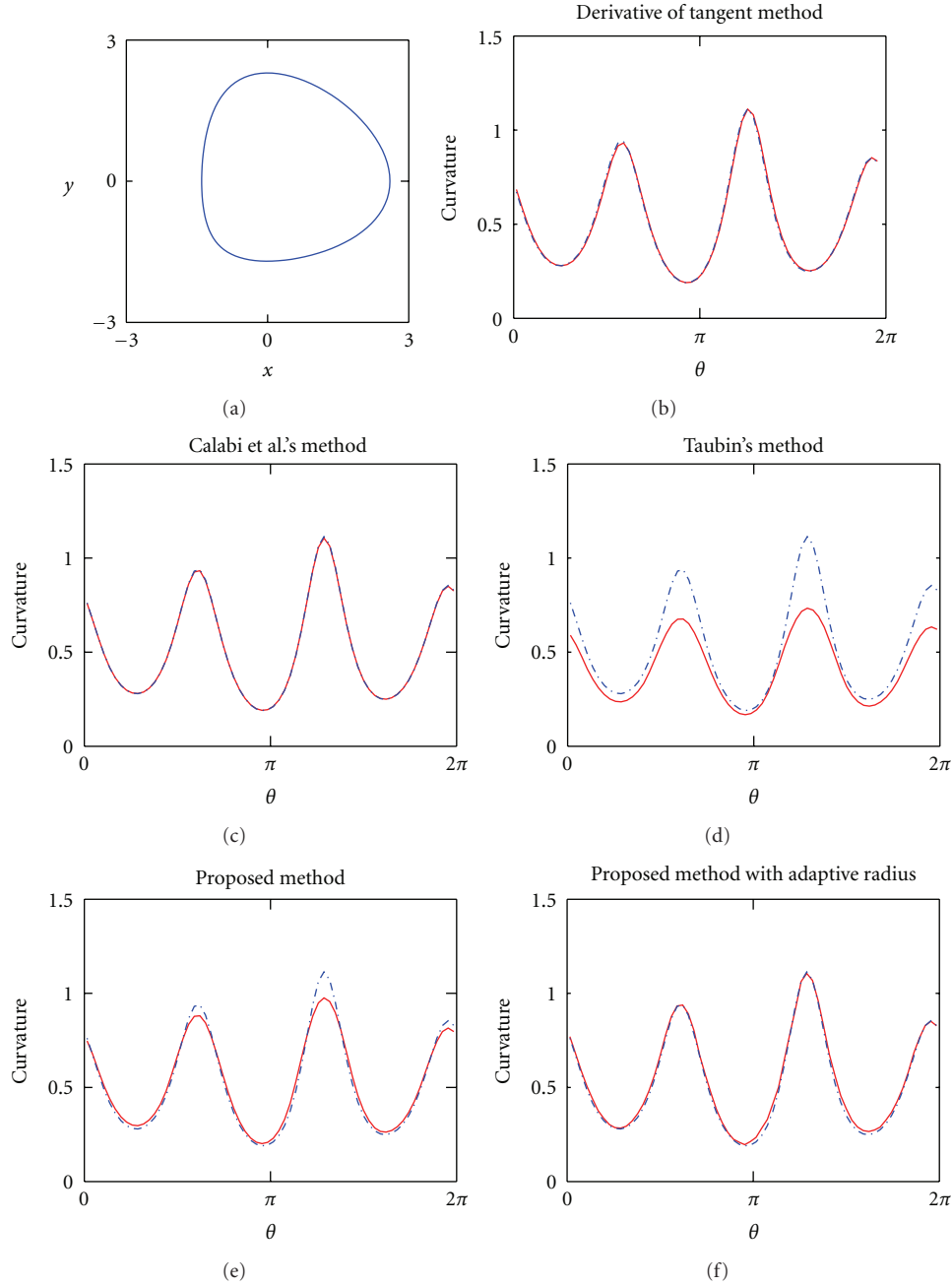


FIGURE 8: (a) The closed curve $\{x = 2 \cos \theta + (3/5)\cos^2\theta, y = 2 \sin \theta + (3/10)\sin^2\theta \mid \theta \in [0, 2\pi]\}$, (b) curvature estimate obtained by derivative of tangent, (c) curvature estimate obtained by Calabi et al.'s algorithm, (d) curvature estimate obtained by Taubin's algorithm, (e) curvature estimate obtained by line integrals, and (f) curvature estimate obtained by line integrals with adaptive radius. Notice that a dashed blue line denotes the true curvature.

the peak but inaccurate in the flat regions. On the other hand, the curvature estimator with a fixed oversize radius will lead to over smoothing and hence be inaccurate at the peak. Therefore, it is desirable that the curvature estimator could adapt according to the input data. By adjusting the radius adaptively, we observe that the precision of curvature estimation is significantly improved. Figure 5 depicts the estimation errors of using fixed radii and adaptive radius.

It is obvious that the estimation errors of the adaptive radius algorithm are much smaller than those of the fixed radius algorithm. In short, we have demonstrated that the estimation accuracy depends largely on the radius of Ω_r and it can be significantly improved by using adaptive radius.

5.3. Sensitivity to Perturbations. In Section 3, the concept of curvature estimation via line integrals has been developed

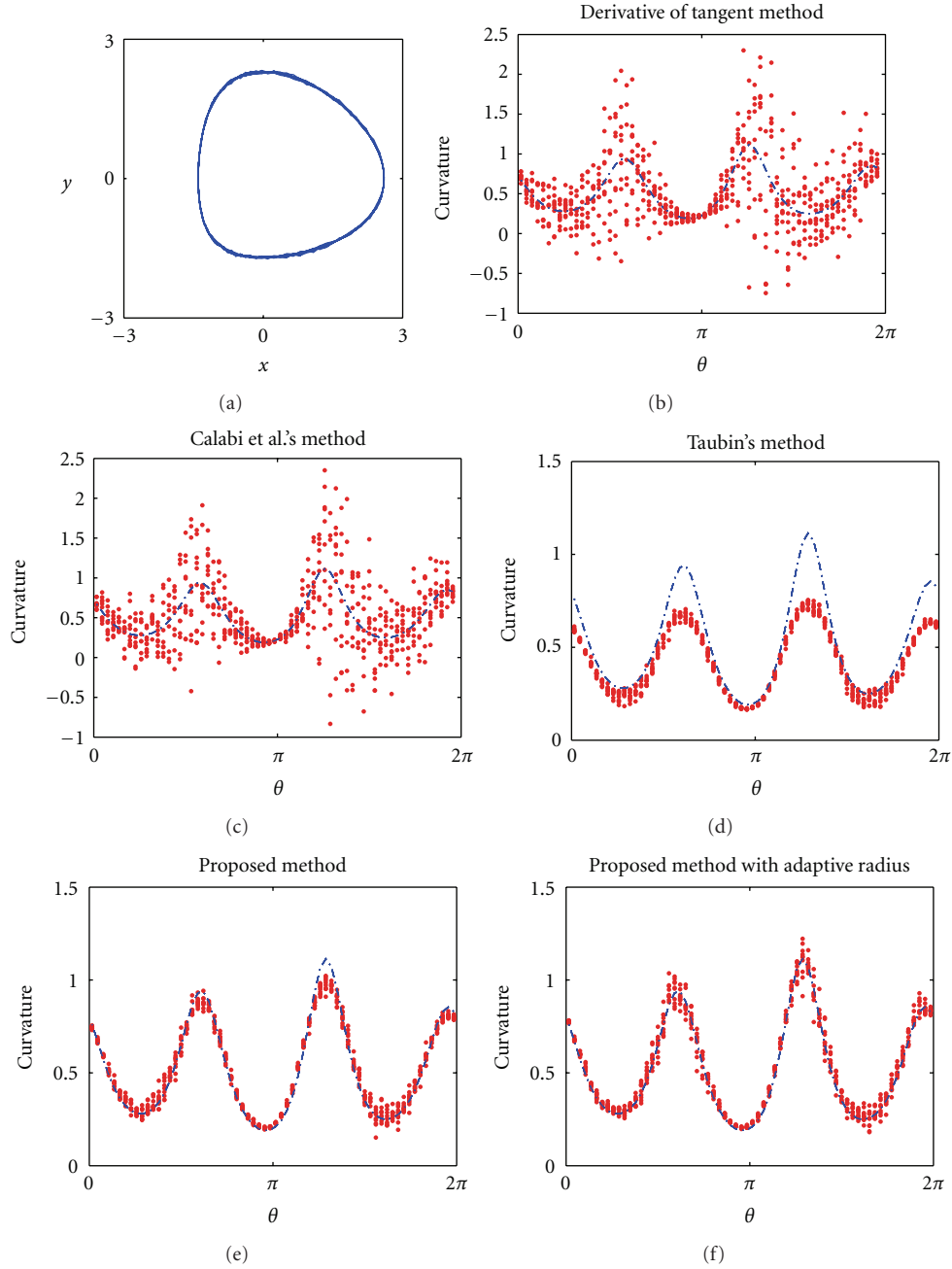


FIGURE 9: Trial-to-trial variability in curvature estimates. The data consist of 10 trials. (a) The closed curves $\{x = 2 \cos \theta + (3/5)\cos^2 \theta, y = 2 \sin \theta + (3/10)\sin^2 \theta \mid \theta \in [0, 2\pi]\}$ with additive Gaussian noise, (b) curvature estimate obtained by derivative of tangent, (c) curvature estimate obtained by Calabi et al.'s algorithm, (d) curvature estimate obtained by Taubin's algorithm, (e) curvature estimate obtained by line integrals, and (f) curvature estimate obtained by line integrals with adaptive radius. Notice that these figures have different ranges in vertical coordinate because some methods yield noisy results. The true curvature is denoted by a dashed blue line.

under an error-free assumption. However, in practical applications, perturbations in the input data can arise from many sources, such as roundoff errors or sensor noise. In this section, we will evaluate the robustness of the proposed curvature estimator and will compare it with existing approaches.

Before dealing with noisy data, we first consider noise-free cases. Two curves are utilized in our experiments. One is

a sinusoidal curve which contains both positive and negative curvatures (see Figure 6(a)). The other one is a closed curve given by $\{x = 2 \cos \theta + (3/5)\cos^2 \theta, y = 2 \sin \theta + (3/10)\sin^2 \theta \mid \theta \in [0, 2\pi]\}$ (see Figure 8(a)). These continuous curves have been discretized by uniformly sampling in the angular variable and curvature is estimated at each sampling point.

Figures 6(b) and 8(b) give the curvature estimate obtained by the derivative of tangent [7]. The results

obtained by Calabi et al.'s method [41] are shown in Figures 6(c) and 8(c). Both of these methods calculate a curvature estimate from three successive sampling points and thus can yield an excellent accuracy under a noise-free condition. However, this kind of approach cannot have reliable results under practical situations where perturbations are inevitable. By using another integral-based method [8], the curvature estimates calculated from the noisy shapes are shown in Figures 7(d) and 9(d). Similar to the proposed method with a fixed radius, this method can obtain reliable results from noisy data but has large estimation errors in high curvature regions.

Compared with the above mentioned methods, the proposed curvature estimator with a fixed radius has larger estimation errors, especially in sharp regions (Figures 6(e) and 8(e)). This is an expected result because the circle Ω_r provides a smoothing effect on sharp corners. As one can see in Figures 6(f) and 8(f), the estimation accuracy is significantly improved once the radius of Ω_r at each point is adaptively determined.

In the following experiments, we add noise to the original shapes and then perform curvature estimation on the noisy shapes. Figure 7 depicts the results of 10 trials. It is obvious that the proposed method, with or without adaptive radius, is able to obtain reasonable estimates while the derivative of tangent [7] and Calabi et al.'s algorithm [41] have significantly larger trial-to-trial variabilities. The same phenomenon is also observed in Figure 9. The results strongly suggest that sensitivity to perturbations is significantly reduced by using the proposed method. The proposed approach for curvature estimation is clearly a better choice if the input data are likely to be noisy.

6. Conclusions and Future Work

A novel curvature estimator, which achieves robustness to noise without sacrificing estimation accuracy, is presented in this paper. The novelty of the proposed method lies in performing line integrals on a circle. Because of performing integrations, we can avoid numerical differentiation which is a notoriously unstable process. Furthermore, instead of choosing a fixed radius for the circle, an optimal radius is determined at each point to minimize estimation error. We present extensive simulation results that demonstrate the effectiveness of our approach as compared with the recently proposed approaches [7, 8, 41]. Notice that we choose [8] for comparison because it is frequently used as a baseline algorithm in the literature. Although the estimator introduced by Taubin [8] is aiming for principal curvatures on surfaces, it can be trivially changed to compute curvature on curves.

An important issue for future research is to generalize the proposed framework to the estimation of principal curvature. To position our approach among others, we would also like to conduct a comparative study of more curvature estimation methods.

Acknowledgment

The authors have been partially supported by the National Science Council, Taiwan (Grant no. 98-2221-E-194-039-MY3).

References

- [1] P. J. Besl and R. C. Jain, "Invariant surface characteristics for 3D object recognition in range images," *Computer Vision, Graphics and Image Processing*, vol. 33, no. 1, pp. 33–80, 1986.
- [2] U. M. Landau, "Estimation of a circular arc center and its radius," *Computer Vision, Graphics and Image Processing*, vol. 38, no. 3, pp. 317–326, 1987.
- [3] S. M. Thomas and Y. T. Chan, "A simple approach for the estimation of circular arc center and its radius," *Computer Vision, Graphics and Image Processing*, vol. 45, no. 3, pp. 362–370, 1989.
- [4] P. J. Flynn and A. K. Jain, "On reliable curvature estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 110–116, 1989.
- [5] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," *Computer Aided Geometric Design*, vol. 22, no. 2, pp. 121–146, 2005.
- [6] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 2–14, 1986.
- [7] M. Worring and A. W. M. Smeulders, "Digital curvature estimation," *CVGIP: Image Understanding*, vol. 58, no. 3, pp. 366–382, 1993.
- [8] G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation," in *Proceedings of the 5th International Conference on Computer Vision*, pp. 902–907, June 1995.
- [9] D. L. Page, A. Koschan, Y. Sun, J. Paik, and M. A. Abidi, "Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 162–167, December 2001.
- [10] C.-K. Tang and G. Medioni, "Curvature-augmented tensor voting for shape inference from noisy 3D data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 858–864, 2002.
- [11] W.-S. Tong and C.-K. Tang, "Robust estimation of adaptive tensors of curvature by tensor voting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 434–449, 2005.
- [12] C. E. Hann and M. S. Hickman, "Projective curvature and integral invariants," *Acta Applicandae Mathematicae*, vol. 74, no. 2, pp. 177–193, 2002.
- [13] S. Manay, D. Cremers, B.-W. Hong, A. J. Yezzi Jr., and S. Soatto, "Integral invariants for shape matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1602–1617, 2006.
- [14] H. Pottmann, J. Wallner, Y.-L. Yang, Y.-K. Lai, and S.-M. Hu, "Principal curvatures from the integral invariant viewpoint," *Computer Aided Geometric Design*, vol. 24, no. 8-9, pp. 428–442, 2007.
- [15] E. Mammen, "Bootstrap and wild bootstrap for high dimensional linear models," *The Annals of Statistics*, vol. 21, no. 1, pp. 255–285, 1993.

- [16] E. Trucco and R. B. Fisher, "Experiments in curvature-based segmentation of range data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 177–182, 1995.
- [17] E. Magid, O. Soldea, and E. Rivlin, "A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data," *Computer Vision and Image Understanding*, vol. 107, no. 3, pp. 139–159, 2007.
- [18] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, Orlando, Fla, USA, 1982.
- [19] I. M. Anderson and J. C. Bezdek, "Curvature and tangential 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 27–40, 1984.
- [20] S. Hermann and R. Klette, "Multigrid analysis of curvature estimators," in *Proceedings of the Image Vision Computing New Zealand*, pp. 108–112, 2003.
- [21] D. Coeurjolly, S. Miguët, L. Tougne, and E. Laboratoire, "Discrete curvature based on osculating circle estimation," in *Workshop on Visual Form*, pp. 303–312, Springer, London, UK, 2001.
- [22] E. M. Stokely and S. Y. Wu, "Surface parametrization and curvature measurement of arbitrary 3-D objects: five practical methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 833–840, 1992.
- [23] B. Hamann, "Curvature approximation for triangulated surfaces," in *Geometric Modelling*, pp. 139–153, Springer, Berlin, Germany, 1993.
- [24] P. Krsek, T. Pajdla, and V. Hlaváč, "Estimation of differential parameters on triangulated surface," in *Proceedings of the 21st Workshop of the Austrian Association for Pattern Recognition*, 1997.
- [25] D. S. Meek and D. J. Walton, "On surface normal and Gaussian curvature approximations given data sampled from a smooth surface," *Computer Aided Geometric Design*, vol. 17, no. 6, pp. 521–543, 2000.
- [26] I. Douras and B. Buxton, "Three-dimensional surface curvature estimation using quadric surface patches," in *Proceedings of the Scanning*, Paris, France, May 2002.
- [27] G. Xu, "Discrete Laplace-Beltrami operators and their convergence," *Computer Aided Geometric Design*, vol. 21, no. 8, pp. 767–784, 2004.
- [28] F. K. H. Quek, R. W. I. Yarger, and C. Kirbas, "Surface parameterization in volumetric images for curvature-based feature classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 33, no. 5, pp. 758–765, 2003.
- [29] X. Chen and F. Schmitt, "Intrinsic surface properties from surface triangulation," in *Proceedings of the the 2nd European Conference on Computer Vision*, pp. 739–743, 1992.
- [30] R. Martin, "Estimation of principal curvatures from range data," *International Journal of Shape Modeling*, vol. 4, no. 1, pp. 99–109, 1998.
- [31] A. M. McIvor and R. J. Valkenburg, "A comparison of local surface geometry estimation methods," *Machine Vision and Applications*, vol. 10, no. 1, pp. 17–26, 1997.
- [32] C. Tang and G. Medioni, "Robust estimation of curvature information from noisy 3D data for shape description," in *Proceedings of the 17th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 1, pp. 426–433, September 1999.
- [33] C. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 859–872, 1989.
- [34] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1088–1095, April 1991.
- [35] B. K. Ray and K. S. Ray, "Detection of significant points and polygonal approximation of digitized curves," *Pattern Recognition Letters*, vol. 13, no. 6, pp. 443–452, 1992.
- [36] W.-Y. Wu, "Dominant point detection using adaptive bending value," *Image and Vision Computing*, vol. 21, no. 6, pp. 517–525, 2003.
- [37] D. H. Foster and K. Zychaluk, "Nonparametric estimates of biological transducer functions," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 49–58, 2007.
- [38] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall, New York, NY, USA, 1993.
- [39] W. Hardle and J. S. Marron, "Bootstrap simultaneous error bars for nonparametric regression," *The Annals of Statistics*, vol. 19, no. 2, pp. 778–796, 1991.
- [40] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1976.
- [41] E. Calabi, P. J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker, "Differential and numerically invariant signature curves applied to object recognition," *International Journal of Computer Vision*, vol. 26, no. 2, pp. 107–135, 1998.