Data Hiding in Curves With Application to Fingerprinting Maps

Hongmei Gou, *Student Member, IEEE,* and Min Wu, *Member, IEEE*

***Abstract*—This paper presents a new data hiding method for curves. The proposed algorithm parameterizes a curve using the B-spline model and adds a spread spectrum sequence to the coordinates of the B-spline control points. In order to achieve robust fingerprint detection, an iterative alignment-minimization algorithm is proposed to perform curve registration and to deal with the nonuniqueness of B-spline control points. Through experiments, we demonstrate the robustness of the proposed data-hiding algorithm against various attacks, such as collusion, cropping, geometric transformations, vector/raster-raster/vector conversions, printing-and-scanning, and some of their combinations. We also show the feasibility of our method for fingerprinting topographic maps as well as writings and drawings.**

***Index Terms*—B-splines, collusion-resistant fingerprinting, data embedding, geospatial data protection, map watermarking, resilience to printing-and-scanning.**

I. Introduction

MAPS represent geospatial information ubiquitous in government, military, intelligence, and commercial operations. The traditional way to protect a map from unauthorized copying and distribution is to place deliberate errors in the map, such as spelling "Nelson Road" as "Nelsen Road," bending a road in a wrong way, and/or placing a nonexisting pond. If an unauthorized user has a map containing basically the same set of errors, this is a strong piece of evidence for piracy that can be presented in court. One of the classic lawsuits is the Rockford Map Pub. versus Dir. Service Co. of Colorado, 768 F.2d 145, 147 (7th Cir., 1985), where phony middle initials of names in a map spelled out "Rockford Map Inc." when read from the top of the map to the bottom, and thus, copyright infringement was found. However, the traditional protection methods alter the geospatial meanings conveyed by a map, which can cause serious problems in critical government, military, intelligence, and commercial operations that require high-fidelity geospatial information. Furthermore, in the situations where distinct errors serve as fingerprints to trace individual copies, such deliberately placed errors can be easily identified and removed by computer programs after multiple copies of a map are brought to the digital domain. All of these limitations of the traditional methods prompt the need for a modern way to map protection that can be more effective and less intrusive.

Curves are one of the major components appearing in maps as well as drawings and signatures. A huge amount of curve-based documents are being brought to the digital domain, owing to the popularity of scanning devices and pen-based devices (such as Tablet PCs). Digital maps and drawings are also generated directly by various computer programs, such as map-making software and computer-aided design systems. Having the capability of hiding digital watermarks or other secondary data in curves can facilitate digital rights management of important documents in government, military, and commercial operations. For example, trace-and-track capabilities can be provided by invisibly embedding a unique ID, which is referred to as a *digital fingerprint* to each copy of a document before distributing it to a user [3], [4]. In this paper, we present a new, robust data-hiding technique for curves and investigate its feasibility for fingerprinting maps.

As a forensic mechanism to deter information leakage and to trace traitors, digital fingerprints must be difficult to remove. For maps and other visual documents, the fingerprint has to be embedded in a robust way against common processing and malicious attacks. Some examples include collusion, where several users combine information from several copies, which have the same content but different fingerprints, to generate a new copy in which the original fingerprints are removed or attenuated [3]; various geometric transformations, such as rotation, scaling, and translation (RST); and D/A-A/D conversions, such as printing-and-scanning. On the other hand, the fingerprint must be embedded in a visually nonintrusive way without changing the geographical and/or visual meanings conveyed by the document. This is because the intrusive changes may have serious consequences in critical military and commercial operations, for example, when inaccurate data are given to troops or fed into navigation systems.

There are a very limited amount of existing works on watermarking maps [5], and few works exploit curve features or address fingerprinting issues. A text-based geometric normalization method was proposed in [6], whereby text labels are first used to normalize the orientation and scale of the map image, and conventional robust watermarking algorithms for grayscale images are then applied. As a map can be represented as a set of vectors, two related works on watermarking vector graphics perturb vertices through Fourier descriptors of polygonal lines [7] or spectral analysis of mesh models [8] to embed copyright

Manuscript received July 1, 2004; revised March 2, 2005. This work was supported in part by the U.S. National Science Foundation under CAREER Award CCR-0133704 and the U.S. Air Force Research Laboratory under Grant FA8750-05-1-0238. Preliminary results of this work were presented in IEEE International Conference on Image Processing, Singapore, 2004 [1] and the IEEE International Conference on Acousctics, Speech, and Signal Processing, Philadelphia, PA, 2005 [2]. The associate editor coordinating the review of this manuscript and approving it for publication was Guest Editor Prof. Pierre Moulin.

The authors are with the Department of Electrical and Computer Engineering and the Institute of Advanced Computing Studies, University of Maryland, College Park, MD 20742 USA (e-mail: hmgou@eng.umd.edu; minwu@eng.umd.edu).

Digital Object Identifier 10.1109/TSP.2005.855411

marks. The embedding in [7] introduces visible distortions, as shown by the experimental results in the paper. The watermarking approach in [8] has high complexity resulting from the mesh spectral analysis, and it cannot be easily applied to maps beyond urban areas, where curves become essential components in mapping a vast amount of land and underwater terrains. Since curve-based documents can also be represented as binary bitmap images (known as the raster representation), we expand the literature survey to data embedding works for general binary images. The data hiding algorithm in [9] enforces the ratio of black versus white pixels in a block to be larger or smaller than 1, and flippable pixels are defined and used in [10] and [11] to enforce a specific block-based relationship to embed data. The fragility of these embedding techniques and the dependence on precise sampling of pixels for correct decoding pose challenges in surviving geometric transformations, printing-and-scanning, and malicious removal in fingerprinting applications. A few other works embed information in dithered images by manipulating the dithering patterns, in fax images by manipulating the run-length [12], and in textual images by changing the line spacing and character spacing [13]. These works cannot be easily extended to robustly mark curve-based documents.

Several watermarking algorithms on graphic data explore compact representations of curves or surfaces for data embedding, such as through the nonuniform rational B-spline (NURBS) model. The work in [14] explains how to embed data in NURBS curves and surfaces without changing the shape or increasing the number of B-spline parameters. The approach demonstrated in the paper relies on reparameterizing a curve or surface using a rational linear function that has an offset determined by the bits to be embedded. The embedded data are fragile and can be removed by perturbing the NURBS parameters or another round of reparameterization. The work in [15] and [16] focuses on three-dimensional (3-D) surfaces and extracts NURBS features from a 3-D surface to form a few two-dimensional (2-D) arrays. Through DCT-domain embedding in these virtual images, a watermark is embedded into the 3-D NURBS surfaces. The work in [17] employs a different domain for 3-D surfaces through multiresolution mesh modeling and embeds a spread spectrum watermark by perturbing the mesh vertices along the direction of the surface normal. Registration techniques for 3-D NURBS surfaces, such as [18], may be employed to facilitate the alignment of the test surfaces with the original reference surface prior to watermark detection. These prior works provide enlightening analogies for watermarking 2-D curves in the B-spline feature domain. As most existing exploration either has limited robustness or targets mainly 3-D surfaces, there are few discussions on robust fingerprinting of curves. To the best of our knowledge, no existing watermarking work has demonstrated the robustness against curve format conversions and D/A-A/D conversions or addressed collusion resistance and traitor tracing issues for curves.

In this paper, we propose a robust curve watermarking method and apply it to fingerprinting maps without interfering with the geospatial meanings conveyed by the map. We select B-spline control points of curves as the feature domain and add mutually independent, noise-like sequences as digital fingerprints to the coordinates of the control points. A proper set of B-spline control points forms a compact collection of salient features representing the shape of the curve, which is analogous to the perceptually significant components in continuous-tone images [19]. The shape of curves is also invariant to such challenging attacks as printing-and-scanning and vector/raster-raster/vector conversions. The additive spread spectrum embedding and the corresponding correlation-based detection generally provide a good tradeoff between imperceptibility and robustness [19], especially when the original host signal is available to the detector, as in most of the fingerprinting applications [3]. To determine which fingerprint sequence(s) is(are) present in a test curve, registration with the original unmarked curve is an indispensable preprocessing step. B-splines have invariance to affine transformations in that the affine transformation of a curve is equivalent to the affine transformation of its control points. This affine invariance property of B-splines can facilitate automatic curve registration. Meanwhile, as a curve can be approximated by different sets of B-spline control points, we propose an iterative alignment-minimization (IAM) algorithm to simultaneously align the curves and identify the corresponding control points with high precision. Through the B-spline based data hiding as well as the IAM algorithm for robust fingerprint detection, our curve watermarking technique can sustain a number of challenging attacks, such as collusion, cropping, geometric transformations, vector/raster-raster/vector conversions, and printing-and-scanning and is therefore capable of building collusion-resistant fingerprinting for maps and other curve-based documents.

The paper is organized as follows: Section II discusses the feature domain in which data hiding is performed and presents the basic embedding and detection algorithms with experimental results on marking simple curves. Section III details the proposed iterative alignment-minimization algorithm for the fingerprint detection and analyzes its robustness. Experimental results on fingerprinting topographic maps are presented in Section IV to demonstrate the robustness of our method against a number of distortions and attacks. Finally, conclusions are drawn in Section V.

II. Basic Embedding and Detection

Our proposed algorithm employs B-spline control points of curves as the feature domain, and adopts spread spectrum embedding [19] for robustly watermarking the coordinates of the control points. The fingerprints for different users are approximately orthogonal and are generated by a pseudo-random number generator with different keys, and the detection is based on correlation statistics. In Sections II-A–B, we explain the main steps of the basic embedding and detection method in detail.

A. Feature Extraction

A number of approaches have been proposed for curve modeling, including using chain codes, Fourier descriptors, autoregressive models, and B-splines [20]. Among them, B-splines are particularly attractive and have been extensively used in computer-aided design and computer graphics. This is mainly be-

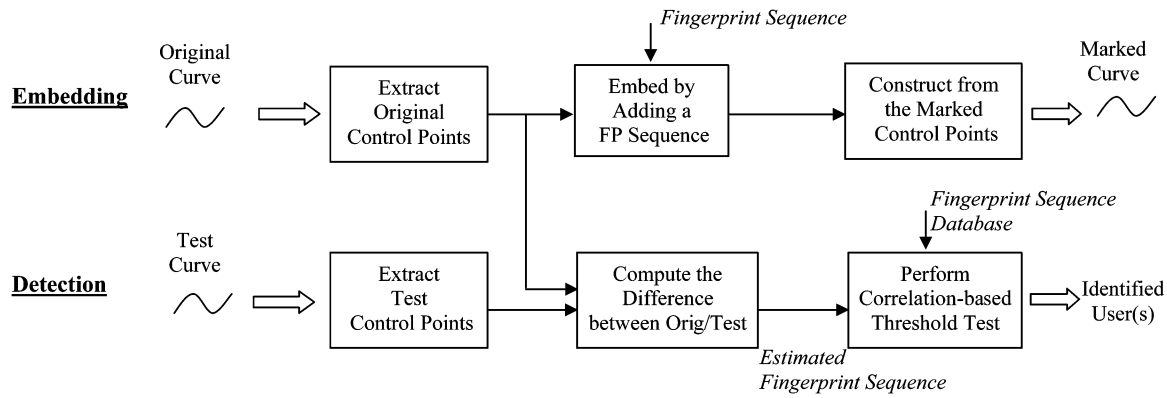

Fig. 1. Basic embedding and detection process of data hiding in curves.

cause the B-spline model provides a continuous approximation of a curve with excellent local shape control and is invariant to affine transformations [21]. These advantages also lead to our choosing B-splines as the feature domain for embedding data in curves.

B-splines are piecewise polynomial functions that provide local approximations of curves using a small number of parameters known as the *control points* [20]. Let $\{\mathbf{p}(t)\}$ denote a curve, where $\mathbf{p}(t) = (p_x(t), p_y(t))$ and t is a continuous indexing parameter. Its B-spline approximation $\{\mathbf{p}^{[B]}(t)\}$ can be written as

$$\mathbf{p}^{[B]}(t) = \sum_{i=0}^{n} \mathbf{c}_i B_{i,k}(t) \tag{1}$$

where t ranges from 0 to $n-1$, $\mathbf{c}_i = (c_{x_i}, c_{y_i})$ is the ith control point ($i = 0, 1, \ldots, n$), and $B_{i,k}(t)$ is the weight of the ith control point for the point $\mathbf{p}^{[B]}(t)$ and known as the kth order B-spline blending function. $B_{i,k}(t)$ is recursively defined as

$$\begin{aligned} B_{i,1}(t) &= \begin{cases} 1, & t_i \le t < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \\ B_{i,k}(t) &= \frac{(t-t_i)B_{i,k-1}(t)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-t)B_{i+1,k-1}(t)}{t_{i+k}-t_{i+1}} \\ & k = 2, 3, \ldots \end{aligned} \tag{2}$$

where $\{t_i\}$ are parameters known as *knots* and represent locations where the B-spline functions are tied together [20]. The placement of knots controls the form of B-spline functions and in turn the control points.

As a compact representation, the number of B-spline control points necessary to represent a curve at a desired precision can be much smaller than the number of points that can be sampled from the curve. Thus, given a set of samples on the curve, finding a smaller set of control points for its B-spline approximation that minimizes the approximation error to the original curve can be formulated as a least-squares problem. Coordinates of the $m+1$ samples on the curve can be represented as an $(m+1)\times 2$ matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \cdots \\ \mathbf{p}_m \end{bmatrix} = \begin{bmatrix} p_{x_0} & p_{y_0} \\ p_{x_1} & p_{y_1} \\ \cdots & \cdots \\ p_{x_m} & p_{y_m} \end{bmatrix} \triangleq (\mathbf{p}_x, \mathbf{p}_y). \tag{3}$$

The indexing values of the B-spline blending functions corresponding to these $m+1$ samples are $t = s_0, s_1, s_2, \ldots, s_m$, where $s_0 < s_1 < s_2 < \cdots < s_m$. Further, let $\mathbf{C}$ represent a set of $n+1$ control points

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \cdots \\ \mathbf{c}_n \end{bmatrix} = \begin{bmatrix} c_{x_0} & c_{y_0} \\ c_{x_1} & c_{y_1} \\ \cdots & \cdots \\ c_{x_n} & c_{y_n} \end{bmatrix} \triangleq (\mathbf{c}_x, \mathbf{c}_y). \tag{4}$$

Then, we can write the least-squares problem with its solution as

$$\min_{\mathbf{C}} \|\mathbf{BC} - \mathbf{P}\|^2 \Longrightarrow \mathbf{C} = \left(\mathbf{B^T B}\right)^{-1} \mathbf{B^T P} = \mathbf{B}^{\dagger}\mathbf{P} \tag{5}$$

where $\{\mathbf{B}\}_{ji}$ is the value of the kth-order B-spline blending function $B_{i,k}(t)$ in (2) evaluated at $t = s_j$ for the ith control point and $\dagger$ denotes the pseudo inverse of a matrix. Because of the natural decoupling of the x and y coordinates in the B-spline representation, we can solve the problem separately along each of the two coordinates as

$$\begin{cases} \min_{\mathbf{c}_x} \|\mathbf{Bc}_x - \mathbf{p}_x\|^2 \\ \min_{\mathbf{c}_y} \|\mathbf{Bc}_y - \mathbf{p}_y\|^2 \end{cases} \Longrightarrow \begin{cases} \mathbf{c}_x = \mathbf{B}^{\dagger}\mathbf{p}_x \\ \mathbf{c}_y = \mathbf{B}^{\dagger}\mathbf{p}_y. \end{cases} \tag{6}$$

B. Basic Embedding and Detection Methods in the Control-Point Domain

Control points of a curve are analogous to perceptually significant components of a continuous-tone image [19] in that they form a compact set of salient features for curves. In such a feature domain, we apply spread spectrum embedding and correlation-based detection, as shown in Fig. 1.

In the embedding, we use mutually independent, noise-like sequences as digital fingerprints to represent different users/IDs for trace and track purposes. As each of the $n+1$ control points has two coordinate values x and y, the overall length of the fingerprint sequence is $2(n+1)$. To apply spread spectrum embedding on a curve, we add a scaled version of the fingerprint sequence $(\mathbf{w}_x, \mathbf{w}_y)$ to the coordinates of a set of control points obtained from the previous subsection. This results in a set of watermarked control points $\left(\mathbf{c}'_x, \mathbf{c}'_y\right)$ with

$$\begin{cases} \mathbf{c}'_x = \mathbf{c}_x + \alpha \mathbf{w}_x \\ \mathbf{c}'_y = \mathbf{c}_y + \alpha \mathbf{w}_y \end{cases} \tag{7}$$

where α is a scaling factor adjusting the fingerprint strength. A watermarked curve can then be constructed according to the B-spline synthesis equation (1) using these watermarked control points.

To determine which fingerprint sequence(s) is(are) present in a test curve, we first need to perform registration using as a reference the original unmarked curve that is commonly available to a detector in fingerprinting applications. After registration, control points $(\tilde{\mathbf{c}}_x, \tilde{\mathbf{c}}_y)$ are extracted from the test curve. The accurate registration and correct extraction of control points are crucial to the detection of fingerprints, which will be detailed in Section III. Assuming we have the set of sample points given by $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y) = (\mathbf{B}(\mathbf{c}_x + \alpha \mathbf{w}_x), \mathbf{B}(\mathbf{c}_y + \alpha \mathbf{w}_y))$, we can extract the test control points $(\tilde{\mathbf{c}}_x, \tilde{\mathbf{c}}_y)$ from $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y)$ using (6). After getting $(\tilde{\mathbf{c}}_x, \tilde{\mathbf{c}}_y)$, we compute the difference between the coordinates of the test and the original control points to arrive at an estimated fingerprint sequence

$$\begin{cases} \tilde{\mathbf{w}}_x = \frac{\tilde{\mathbf{c}}_x - \mathbf{c}_x}{\alpha} \\ \tilde{\mathbf{w}}_y = \frac{\tilde{\mathbf{c}}_y - \mathbf{c}_y}{\alpha}. \end{cases} \tag{8}$$

The estimated fingerprint sequence consists of one or several users' contributions as well as some noise coming from distortions or attacks. The problem of finding out which user(s) has(have) contributed to the estimated fingerprint can be formulated as hypothesis testing [22], which is commonly handled by evaluating the similarity between the estimated fingerprint sequence and each fingerprint sequence in the database through a correlation-based statistic. Various correlation-based statistics share a kernel term that measures the total correlation $\langle \tilde{\mathbf{w}}, \mathbf{w} \rangle$ (where $\tilde{\mathbf{w}} \triangleq [\tilde{\mathbf{w}}_x, \tilde{\mathbf{w}}_y]$ and $\mathbf{w} \triangleq [\mathbf{w}_x, \mathbf{w}_y]$) and differ in how they are normalized. To facilitate the evaluation of detection performance, we often normalize the detection statistic to make it have a unit variance and follow approximately a Gaussian distribution under distortions and attacks. There are several ways to do so [23], for example, to normalize using the product of the noise's standard deviation and the watermark's L_2 norm or through a logarithm-based transformation to be introduced next.

C. Z Statistic for Detection

A nonlinear function of the sample correlation coefficient from the mathematical statistics literature [24] was introduced to the watermarking community by Stone and colleagues of the NEC Research Institute [25]. This is often referred to as the *Fisher's Z statistic*. Among several correlation-based statistics analyzed and compared in [23], the Z statistic shows excellent robustness against different collusion attacks and does not require the explicit estimation of the noise's variance. These advantages make it attractive to handle our problem.

The Z statistic originated from the statistical problem of sampling a bivariate normal population [24], i.e., to obtain independent and identically distributed (i.i.d.) samples from a pair of random variables that are jointly Gaussian distributed, with correlation coefficient ρ unknown to the observers. Let r be the sample correlation coefficient computed from L pairs of sample data, and $-1 \le r \le 1$. The function of r

$$\frac{1}{2} \log \frac{1+r}{1-r}$$

has been shown [24] to asymptotically follow a normal distribution when $L \to \infty$, with the mean approximating $(1/2)\log((1+\rho)/(1-\rho))$ and the variance approximating $1/(L-3)$. Thus, the Z statistic defined below follows a unit-variance Gaussian distribution

$$Z \triangleq \frac{\sqrt{L-3}}{2} \log \frac{1+r}{1-r} \sim \mathcal{N}\left(\frac{\sqrt{L-3}}{2} \log \frac{1+\rho}{1-\rho}, 1 \right). \tag{9}$$

The approximation is found excellent with as few as ten pairs of samples.

In our fingerprint detection problem in the control-point domain, the effective number of samples for computing the statistic is $L = 2(n+1)$. Denoting the average values of the components in $\tilde{\mathbf{w}}$ and $\mathbf{w}$ as $\tilde{\mu}$ and μ, respectively, we compute the sample correlation coefficient r between $\tilde{\mathbf{w}}$ and $\mathbf{w}$ by

$$r = \frac{\sum_{i=1}^{L} (\tilde{w}_i - \tilde{\mu})(w_i - \mu)}{\sqrt{\sum_{j=1}^{L} (\tilde{w}_j - \tilde{\mu})^2 \sum_{k=1}^{L} (w_k - \mu)^2}}. \tag{10}$$

A simplified model considers that the corresponding components of an extracted fingerprint in question and a particular user Alice's fingerprint are i.i.d. samples from a bivariate normal population. When the extracted fingerprint does not have Alice's contribution, the expected correlation coefficient is zero, and the Z statistic will approximately follow a Gaussian distribution with a zero mean and a unit variance. When the fingerprint has Alice's contribution, the Z statistic will have a large positive mean determined by the correlation coefficient ρ. To derive the expression for ρ, we define a random variable $Y \triangleq W + N$ and the extracted fingerprint consists of i.i.d. samples from Y. Here, W is a zero-mean Gaussian random variable representing Alice's fingerprint, and N is a zero-mean Gaussian random variable representing noise. The correlation coefficient of this bivariate normal population (W, Y) is

$$\begin{aligned} \rho &= \frac{\mathrm{cov}(W,Y)}{\sqrt{\mathrm{Var}(W)\mathrm{Var}(Y)}} \\ &= \frac{\mathrm{Var}(W) + \mathrm{cov}(W,N)}{\sqrt{\mathrm{Var}(W)\left[\mathrm{Var}(W) + \mathrm{Var}(N) + 2\mathrm{cov}(W,N)\right]}}. \end{aligned} \tag{11}$$

When W and N are uncorrelated, the correlation coefficient becomes

$$\rho = \sqrt{\frac{\mathrm{Var}(W)}{\mathrm{Var}(W) + \mathrm{Var}(N)}} = \sqrt{\frac{1}{1 + \frac{1}{\mathrm{WNR}}}} \tag{12}$$

where the watermark-to-noise ratio $\mathrm{WNR} \triangleq (\mathrm{Var}(W)/\mathrm{Var}(N))$. Now knowing the distribution of the Z statistics under the presence and absence of Alice's fingerprint, we can compute the probabilities of detection P_d and false alarm P_{fa} for different decision thresholds. A threshold of 3 gives a false alarm probability on the order of 10^{-3}, while a threshold of 6 corresponds to the order of 10^{-9}.

In reality, as the noise introduced by attacks does not necessarily follow a Gaussian distribution and/or the noise samples may be mutually correlated, the Z statistics with true traitors may be different from the unit-variance Gaussian distribution.

The actual distributions of the Z statistics for the fingerprint presence/absence cases and the detection performance in terms of the detection probability and the false alarm probability will be presented in our experimental results.

D. Fidelity and Robustness Considerations

1) Fingerprint Construction: The collusion resistance requirement makes the fingerprinting problem more challenging than robustly embedding a meaningful ID label, as the simple encoding of IDs can be vulnerable to collusion (e.g., different users average their copies of the same content to remove the IDs). Designing a collusion-resistant code is one of the possible approaches and has been studied in [26]. Such a coded approach requires that each code symbol be reliably embedded, which consumes a nontrivial amount of markable features per embedded bit. The markable feature for our curve watermarking problem is the coordinates of control points. As the number of control points is limited and the changes have to be small, orthogonal modulation that uses (approximately) orthogonal signals to represent different users is more attractive than the coded modulation [27]. The general collusion resistance of orthogonal fingerprinting has been studied in [22], which shows the maximum number of colluders that the system can resist is a function of the watermark-to-noise ratio, the number of markable features, the total number of users, as well as the false positive and negative requirements.

While the orthogonal design of fingerprints is conceptually simple and easy to analyze, the practical implementation often employs a pseudo-random number generator to produce a sequence of independent random numbers as a fingerprint and uses different seeds for different users [23], [26], [28]. In this way, the actual fingerprints would be statistically uncorrelated, but they can have a nonzero correlation. This correlation can accommodate a larger number of fingerprint vectors than the vector's dimension, but it also affects the detection performance to some degree. Since the correlation is very low between the independent fingerprints, the impact is small. This can be seen from our experimental results of the detection performance in Sections II-E and IV.

2) Curvature-Based Sampling: The overall distortion introduced by the embedding process on a curve consists of two parts: one is from the watermark signals added to the control point coordinates, and the other is from the B-spline modeling. To make the B-spline synthesized curve as close to the original curve as possible and thus keep the modeling error low, the knots connecting adjacent segments of B-splines should be wisely placed, and the sample points should be properly chosen to feed into the least-squares estimator for the control points. Uniform sampling can be used when there are no abrupt changes in a curve segment, while nonuniform sampling is desirable for curve segments that exhibit substantial variations in curvature.

Inspired by [29] and [30], we employ a curvature-based method to select sample points from raster curves. Formally, the curvature [31] of a point $\mathbf{p}(t) = (p_x(t), p_y(t))$ on a curve $\{\mathbf{p}(t)\}$ is defined as

$$k(t) \triangleq \frac{p'_x p''_y - p'_y p''_x}{\left({p'_x}^2 + {p'_y}^2\right)^{3/2}} \tag{13}$$

where $p'_x = dp_x/dt$, $p''_x = d^2p_x/dt^2$, $p'_y = dp_y/dt$, and $p''_y = d^2p_y/dt^2$. In practical implementations, we approximate the curvature of each point on the curve by measuring the angular change in the tangent line at its location. Specifically, we perform a 1st-order polynomial curve fitting on an l-pixel interval before and after the curve point $\mathbf{p}(t)$ to get two slopes, k_1 and k_2. The approximate curvature $\hat{k}(t)$ is computed by $\hat{k}(t) = |\arctan(k_1) - \arctan(k_2)|$. Based on $\hat{k}(t)$, we select more sample points from higher-curvature segments and fewer from lower-curvature segments.

After selecting $m+1$ sample points and put them together into $(\mathbf{p}_x, \mathbf{p}_y)$ as defined in (3), we need to determine their indexing values $t = s_0, s_1, s_2, \ldots, s_m$, which will be used to evaluate their B-spline blending function values $B_{i,k}(t)$. In our tests, we employ the uniform nonperiodic B-spline blending function of order $k = 3$, and the knot parameters $\{t_i\}$ are determined as $[t_0, t_1, \ldots, t_{n+3}] = [0, 0, 0, 1, 2, 3, \ldots, n-2, n-1, n-1, n-1]$. One way of the indexing-value assignment is known as the chord-length method [32], which increases t values of the sample points in proportion to the chord length

$$s_j = s_{j-1} + \|\mathbf{p}_j - \mathbf{p}_{j-1}\| \frac{n-1}{\sum_{i=1}^{m} \|\mathbf{p}_i - \mathbf{p}_{i-1}\|} \qquad j = 1, 2, \ldots m \tag{14}$$

where $s_0 = 0$, and $\|\mathbf{p}_j - \mathbf{p}_{j-1}\|$ denotes the chord length between points $\mathbf{p}_j$ and $\mathbf{p}_{j-1}$.

In the case of vector curves, we are generally given a set of discrete, nonuniformly spaced points for each curve. Since a vector curve can be rendered as a raster curve by interpolation, we can determine its sample points and their corresponding indexing values by rendering it to be a raster curve and then performing the curvature-based sampling and the chord-length method as described above. A simpler alternative is to directly use the given discrete points as sample points but assign their indexing values according to a curvature-based rule. Specifically, we approximate the curvature of each sample point $(p_x(s_j), p_y(s_j))$ by

$$\hat{k}(s_j) = \left| \arctan \frac{p_y(s_{j+1}) - p_y(s_j)}{p_x(s_{j+1}) - p_x(s_j)} - \arctan \frac{p_y(s_j) - p_y(s_{j-1})}{p_x(s_j) - p_x(s_{j-1})} \right| \tag{15}$$

and then increase t values of the sample points in inverse proportion to their curvature. The higher the curvature, the smaller the increase in the t value. This is equivalent to having more control points for higher-curvature segments. With more "resources" in terms of B-spline control points assigned to segments with more details (i.e., higher curvature segments), it allows for a better B-spline approximation.

3) Determining the Fingerprint Length and Strength: The number of control points is an important parameter for tuning. Depending on the shape of the curve, using too few control points could cause the details of the curve to be lost, while using too many control points may lead to over-fitting and bring artifacts even before data embedding. One simple method of determining the number of control points is to compute the ap-

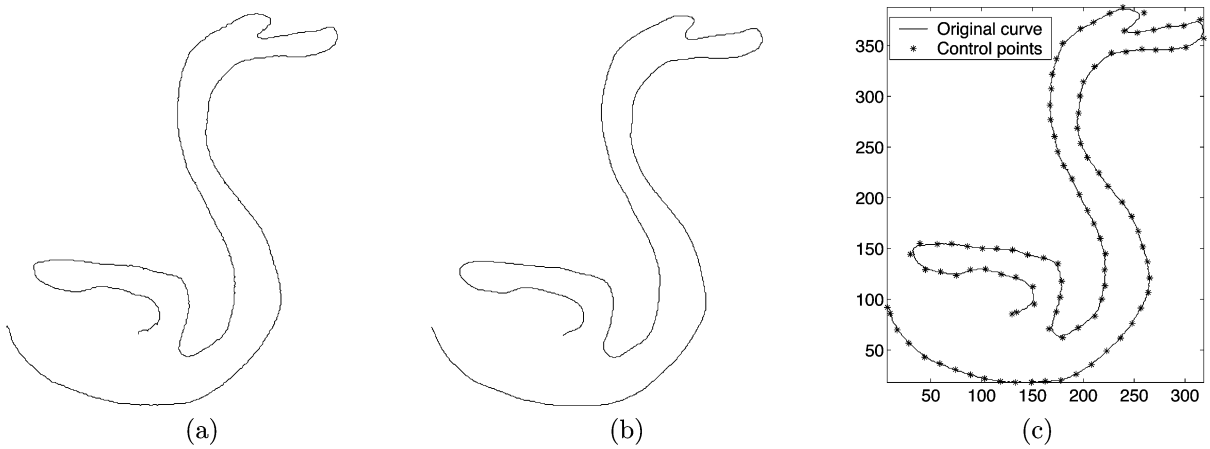

Fig. 2. Fingerprinting a hand-drawn "*Swan*" curve. (a) Original curve. (b) Fingerprinted curve. (c) Control points overlaid on the original curve.

proximate curvature of each sample point as in (15) and assign higher weights to points with higher curvature. We then determine the number of control points according to the total weights of all sample points on the curve. The number of control points not only affects the distortion introduced by the embedding but also determines the fingerprint's robustness against noise and attacks. The more the control points there are, the longer the fingerprint sequence is, and in turn, the more robust the fingerprint is against noise and attacks. Too many control points, however, may lead to over-fitting and incur visible distortions even before data embedding. In our tests, the number of control points is about 5–8% of the total number of curve pixels.

The scaling factor α also affects the invisibility and robustness of fingerprints. The larger the scaling factor is, the more robust the fingerprint is, but it results in a larger distortion. For cartographic applications, industrial standards provide guidelines on the maximum allowable changes [8]. Perturbation of two to three pixels is usually considered acceptable. We use random number sequences with a unit variance as fingerprints and set α to 0.5 in our tests. The difference between two curves can be quantified using such max-min metrics as the Hausdorff distance [33]. More specifically, let $d(a, b)$ be the distance between two points a and b, which are on two curves A and B, respectively. We further define the distance from point a to curve B as $d(a, B) \triangleq \inf_{b \in B} d(a, b)$, and the distance from curve A to curve B as $d_B(A) \triangleq \sup_{a \in A} d(a, B)$. Thus, the Hausdorff distance between curve A and B is

$$h(A, B) \triangleq d_B(A) + d_A(B). \tag{16}$$

4) Nonblind Detection: The basic fingerprint detection presented earlier makes use of the original unmarked copy and is known as *nonblind detection*. While blind detection is preferred for a few major data hiding applications (such as ownership verification, authentication, and annotation), nonblind detection is considered as a reasonable assumption for many fingerprinting applications [4], [26], [34], which is also the focus of this paper. The rationale for nonblind detection is that the fingerprint verification is usually handled by the content owner or by an authorized central server, who can have access to the original host signal and use it in the detection to answer the primary question of whose fingerprint is in the suspicious document. The availability of the original unmarked copy in the detection gives a high equivalent watermark-to-noise ratio, thus allowing for high resistance against noise and attacks. Additionally, using the original unmarked copy as a reference copy, the detector can register a test copy that suffers from geometric distortions, which enables the resilience to various geometric transformations as to be demonstrated later in this paper.

E. Experimental Results of Fingerprinting Simple Curves

To demonstrate our basic embedding and detection algorithms, we first present the fingerprinting results on two simple curves, the "Swan" curve in Fig. 2(a) and the "W" curve in Fig. 3(a). These two curves were hand-drawn on a Tablet PC and stored as binary images of size 329×392 and 521×288, respectively. We use the contour following algorithm in [30] to traverse the curve and obtain a set of ordered curve points. When fingerprinting these two curves, we perform uniform sampling on the curve points and determine the indexing values of the sample points using the chord-length method. We highlight the control points of the "Swan" curve in Fig. 2(c). The fingerprinted curves are shown in Figs. 2(b) and 3(b), where we have marked 101 control points for each curve. With the marked control points, we construct the fingerprinted curve with the same number of points as the original curve by evaluating the B-spline synthesis formula (1) at indexing values uniformly sampled between $t = 0$ and $t = n - 1$. As for the fidelity of the fingerprinted curves, the Hausdorff distance between the original and marked curves is 5.0 for the "Swan" curve, and 3.4 for the "W" curve. The differences are hardly visible to human eyes.

In the detection, we take a fingerprinted curve constructed above as a test curve and apply uniform sampling to it to obtain an approximation of the set of sample points $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y) = (\mathbf{B}(\mathbf{c}_x + \alpha \mathbf{w}_x), \mathbf{B}(\mathbf{c}_y + \alpha \mathbf{w}_y))$ assumed in Section II-B. We then estimate the test control points and perform the correlation-based detection. The detection results on the fingerprinted "W" curve are shown in Fig. 3(c), which illustrates a high Z value for the correct positive detection with the 1000th sequence corresponding to the true user, and very small Z statistics for the correct negative detection with other sequences of the innocent users. To quantify the detection performance, we generate

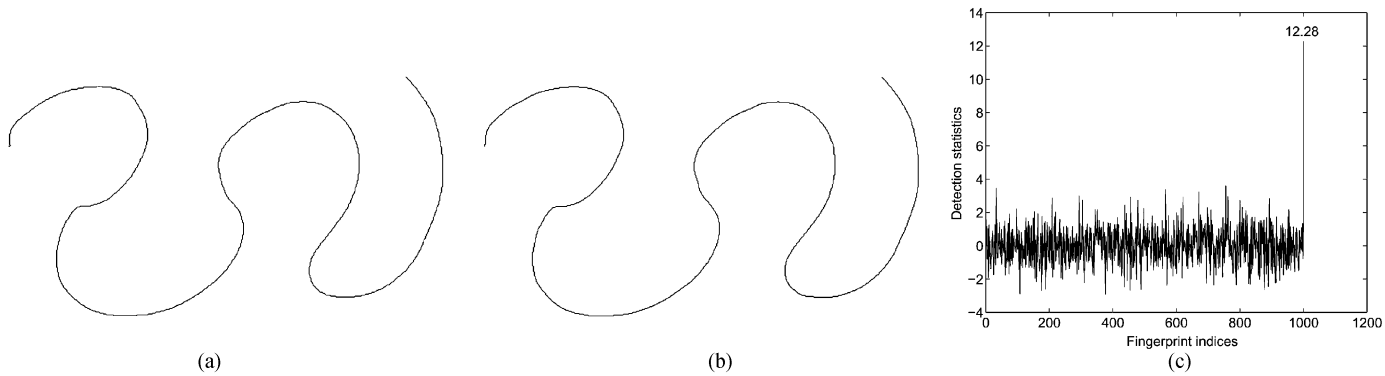

Fig. 3. Fingerprinting a hand-drawn "*W*" curve. (a) Original curve. (b) Fingerprinted curve. (c) Detection statistics.

1000 different sets of fingerprint sequences, and each set consists of 1000 independent fingerprint sequences that are approximately orthogonal to each other. For each set, we embed each of the 1000 fingerprint sequences into the original curve to form a fingerprinted curve, and then estimate a fingerprint sequence from the fingerprinted curve and compute the Z values with the 1000 fingerprint sequences. In this way, we collect a total of $1000 \times 1000 = 1 \times 10^6$ values for the fingerprint presence case and $1000 \times 1000 \times 999 \approx 1 \times 10^9$ values for the fingerprint absence case. Using these data, we plot in Fig. 4 the histogram of the Z values for both fingerprint presence and absence cases. For each of these two sets of data, we calculate its mean and variance and then plot the Gaussian distribution with the calculated mean and variance. As shown by the dashed curves in Fig. 4, the two Gaussian approximations $\mathcal{N}(11.22, 0.87)$ and $\mathcal{N}(-0.0009, 1.00)$ fit the observed Z values very well. We can see that we indeed get an approximate Gaussian distribution with a large positive mean under the presence of a fingerprint and a zero mean under the absence.

We further examine the measured and the Gaussian approximated probabilities of detection and false alarm for different thresholds on the Z statistic. We see from the results in Table I that the Gaussian approximation works well in regions close to the mean, whereas the measured values slightly deviate from the Gaussian approximation in regions farther from the mean. At the same time, we note that in the tail regions where the Gaussian approximation becomes loose, the approximated order of magnitude matches very well with the measured value from our experiments. The miss probability or false alarm probability in the tail regions is already very small (below 10^{-5}). Therefore, for many practical applications, either the probability may be deemed as zero, or an approximation on the order of magnitude would be sufficient. The table also shows that a threshold of 6 on the detection statistics gives a false alarm probability of 10^{-9}, which is sufficiently low for most applications. Thus, we choose 6 as the detection threshold in our tests. The detection result for the "Swan" curve is similar and will not be repeated here.

Furthermore, we examine the survivability of the fingerprints by using our basic scheme, which employs the coordinates of the B-spline control points as the embedding domain, and uses approximately orthogonal spread spectrum signals as fingerprints. We perform a printing-and-scanning test with manual registration between the scanned fingerprinted curve and the original unmarked curve. We print out the fingerprinted "W" curve using a HP laser printer, and scan it back as a 527×288 binary image, as shown in Fig. 5(a). In addition to manual registration, a thinning operation is performed to extract a one-pixel-wide skeleton from the scanned curve that is usually several pixels wide after high-resolution scanning. As we can see from the detection results in Fig. 5(b), despite the curve being simple and the number of control points being relatively small, the fingerprint survives the printing-and-scanning process and gives a detection statistic higher than the detection threshold. The issue of automating the registration process will be addressed in Section III.

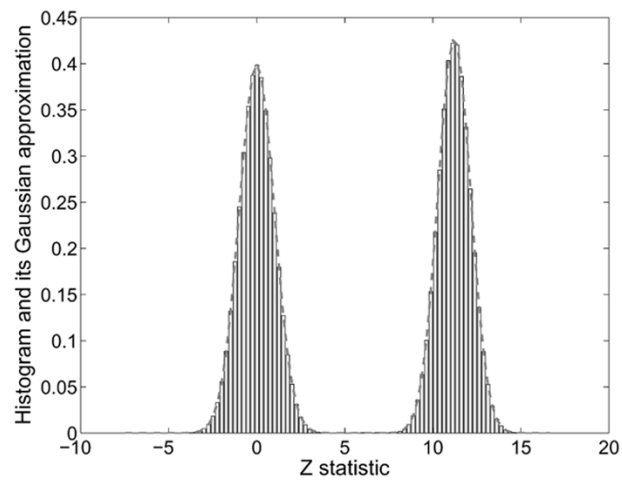

Fig. 4. Histogram and Gaussian approximation of the Z statistics for the "*W*" curve.

III. Iterative Alignment-Minimization Algorithm for Robust Fingerprint Detection

The set of test sample points $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y)$ assumed in Section II-B is not always available to a detector, especially when a test curve undergoes vector-raster conversion, geometric transformations (such as rotation, translation, and scaling), and/or is scanned from a printed hard copy. A pre-processing step preceding the basic fingerprint detection module is needed to align the test curve with the original one. While manual registration between the test curve and the original unmarked curve shown in Section II-E is a possible way to overcome simple geometric distortions, auto-

TABLE I
MEASURED DETECTION PERFORMANCE AND ITS GAUSSIAN APPROXIMATION FOR THE "W" CURVE

Threshold on Z		*3*	*4.5*	*6*	*7.5*	*9*
$1\text{-}P_d$	Measured	0	0	0	4.1×10^{-5}	8.7×10^{-3}
	Gaussian approx.	6.6×10^{-19}	3.1×10^{-13}	1.1×10^{-8}	3.4×10^{-5}	8.7×10^{-3}
P_{fa}	Measured	1.4×10^{-3}	4.1×10^{-6}	2.0×10^{-9}	0	0
	Gaussian approx.	1.3×10^{-3}	3.4×10^{-6}	0.97×10^{-9}	3.1×10^{-14}	1.1×10^{-19}

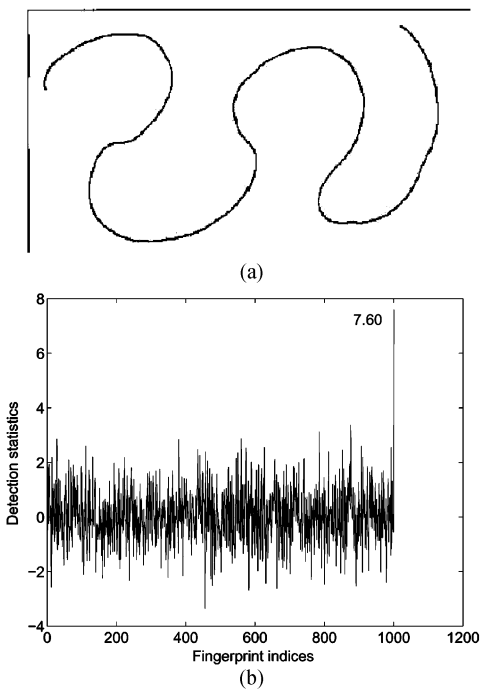

Fig. 5. Printing-and-scanning test for the "W" curve. (a) Fingerprinted curve after printing-and-scanning. (b) Detection statistics.

mated registration is more desirable to improve the accuracy and efficiency of this indispensable preprocessing step. Note that the test curve should be registered with the original unmarked curve, and any "clean/undistorted" fingerprinted copies known to the detector should not be used as a reference for registering the test curve. This is not only because which fingerprints are present in the test curve still remains to be determined but also because using a fingerprinted copy as a reference for registration may increase the false alarm probability in determining the presence or absence of the corresponding fingerprint.

With the affine invariance property, B-splines have been used in a few existing curve alignment works. In the moment-based approach of [29], two affine-related curves are fitted by two separate B-splines, and the transform parameters are estimated by using weighted B-spline curve moments. This method requires taking integration as well as the second-order curve derivatives to obtain the moments. In a recent method employing a *super-curve* [35], two affine-related curves are superimposed together in the same frame, and then this combined super-curve is fitted by a single B-spline. Through minimizing the B-spline fitting error, both transform parameters and control points of the fitting B-spline can be estimated simultaneously. Since neither integration nor differentiation is needed, this method is robust to noise and will serve as a building block in our work.

Another problem related to the test sample points assumed earlier is the inherent nonuniqueness of B-spline control points, which refers to the fact that a curve can be effectively approximated by different sets of B-spline control points. We have seen from Section II-A that B-spline control points are estimated from a set of sample points from the curve. With a different set of sample points or a different indexing-value assignment, we may induce a quite different set of control points that can still well describe the same curve. It is possible for the difference between two sets of unmarked control points to be much larger than the embedded fingerprint sequence, as demonstrated by the example in Fig. 6. Therefore, if we cannot find from a test curve a set of control points corresponding to the one used in the embedding, we may not be able to detect the fingerprint sequence. Considering the one-to-one relationship between sample points (including their indexing values $\{s_j\}$) and control points, we try to find the set of sample points from a test curve that corresponds to the set of sample points used in the embedding. We shall refer to this problem as the *point correspondence problem*. As we shall see, the nonuniqueness issue of B-spline control points can be addressed through finding the point correspondence.

In Sections III-A–C, we first formulate the curve registration and point correspondence problem in the context of fingerprint detection. We then take the curve alignment method introduced in [35] as a building block and propose an Iterative Alignment-Minimization (IAM) algorithm that can perform curve registration and solve the point correspondence problem simultaneously. Finally, we present a detection example for a single curve using the IAM algorithm and discuss the robustness issues.

A. Problem Formulation

We use "View-I" to refer to the geometric setup of the original unmarked curve and "View-II" to refer to the setup of the test curve. Thus, we can register the two curves by transforming the test curve from "View-II" to "View-I" or transforming the original curve from "View-I" to "View-II." We focus on registration under affine transformations, which can represent combinations of scaling, rotation, translation, reflection, and shearing. These are common geometric transformations and can effectively model common scenarios in the printing-and-scanning process.

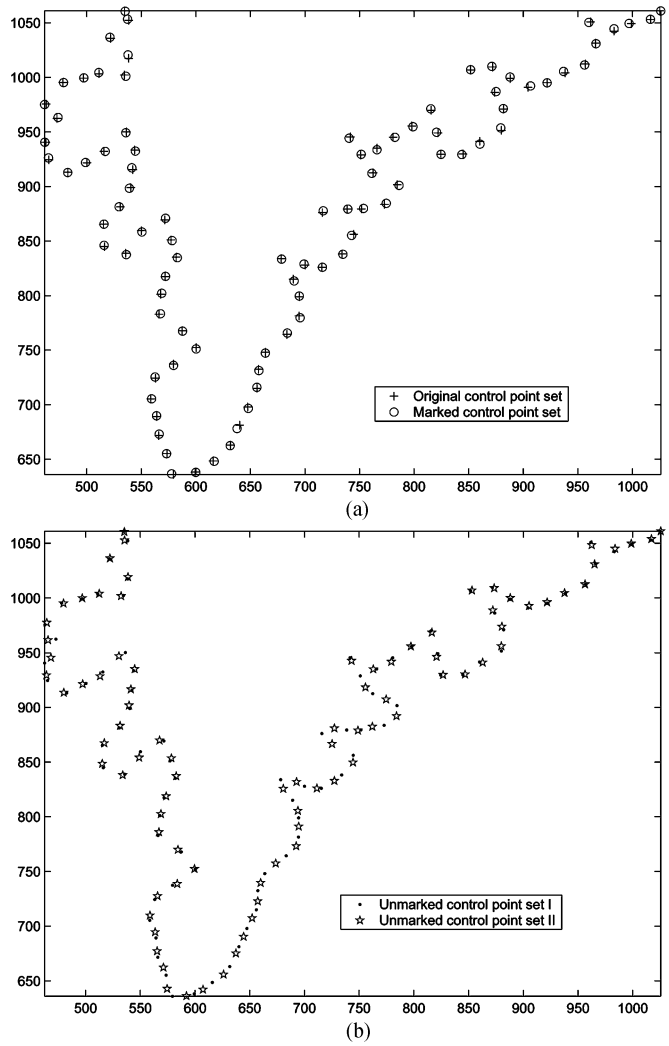

Fig. 6. Nonuniqueness of B-spline control points. (a) Set of control points for the original unmarked curve and its fingerprinted version. (b) Two different sets of control points for modeling the same unmarked curve.

Under affine transformations, each point (x, y) on one curve is transformed to a corresponding point $(\tilde{x}, \tilde{y})$ on another curve via

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \tag{17}$$

where $\{a_{ij}\}$ are parameters representing the collective effect of scaling, rotation, translation, reflection and shearing. The transform parameters can also be represented in a homogeneous coordinate by two column vectors $\mathbf{a}_x$ and $\mathbf{a}_y$, or by a single matrix $\mathbf{A}$:

$$\begin{aligned} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} & \mathbf{a}_x^T & \\ & \mathbf{a}_y^T & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \end{aligned} \tag{18}$$

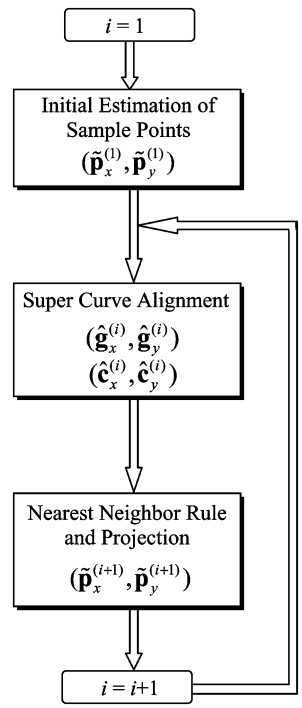

Fig. 7. Basic flow and main modules of the proposed Iterative Alignment-Minimization (IAM) algorithm.

Similarly, the inverse transform can be represented by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} \triangleq \begin{bmatrix} & \mathbf{g}_x^T & \\ & \mathbf{g}_y^T & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix}. \tag{19}$$

The original curve available to the detector in fingerprinting applications can be a raster curve or a vector curve. The detector also knows the original set of sample points $(\mathbf{p}_x, \mathbf{p}_y) \approx (\mathbf{B}\mathbf{c}_x, \mathbf{B}\mathbf{c}_y)$ that is used for estimating the set of control points upon which spread spectrum embedding is applied. The test curve can be a vector curve with sampled curve points $(\tilde{\mathbf{v}}_x, \tilde{\mathbf{v}}_y)$ or a raster curve with pixel coordinates $(\tilde{\mathbf{r}}_x, \tilde{\mathbf{r}}_y)$. A relatively simple case is that the set of discrete points in a test vector curve corresponds to the set of sample points used in the embedding, except with possible affine transformations and noise addition. In this case, the test vector points $(\tilde{\mathbf{v}}_x, \tilde{\mathbf{v}}_y)$ and the original set of control points $(\mathbf{c}_x, \mathbf{c}_y)$ are related by

$$\begin{cases} \tilde{\mathbf{v}}_x = [\,\mathbf{B}(\mathbf{c}_x + \alpha \mathbf{w}_x) \quad \mathbf{B}(\mathbf{c}_y + \alpha \mathbf{w}_y) \quad \mathbf{1}\,]\,\mathbf{a}_x + \mathbf{n}_x \\ \tilde{\mathbf{v}}_y = [\,\mathbf{B}(\mathbf{c}_x + \alpha \mathbf{w}_x) \quad \mathbf{B}(\mathbf{c}_y + \alpha \mathbf{w}_y) \quad \mathbf{1}\,]\,\mathbf{a}_y + \mathbf{n}_y \end{cases} \tag{20}$$

where $(\mathbf{n}_x, \mathbf{n}_y)$ represents additional noise applied to the transformed fingerprinted vector points, and $\mathbf{1}$ is a column vector with all 1s. With the point correspondence available, the only issue is curve alignment, and it can be solved by directly applying the curve alignment method in [35]. However, in addition to possible affine transformations between the original and the test curves, the correct point correspondence information may not always be available. This is especially the case after a fingerprinted curve undergoes vector-raster conversions and/or printing-and-scanning. Under these situations, not only transform parameters for the curve alignment but also the point correspondence must be estimated in order to locate the fingerprinted control points successfully. We consider that both the

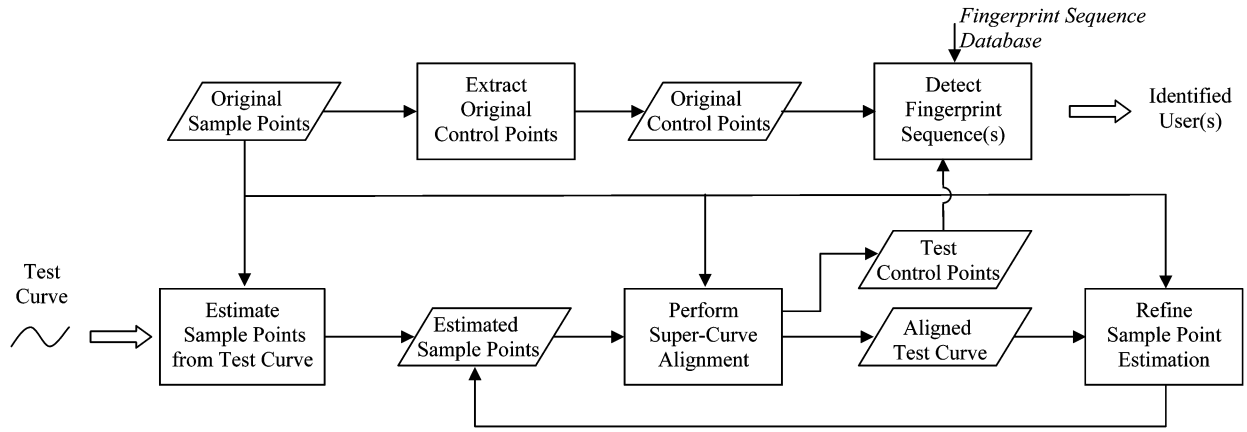

Fig. 8. Block diagram of curve registration and fingerprint detection using the proposed IAM algorithm.

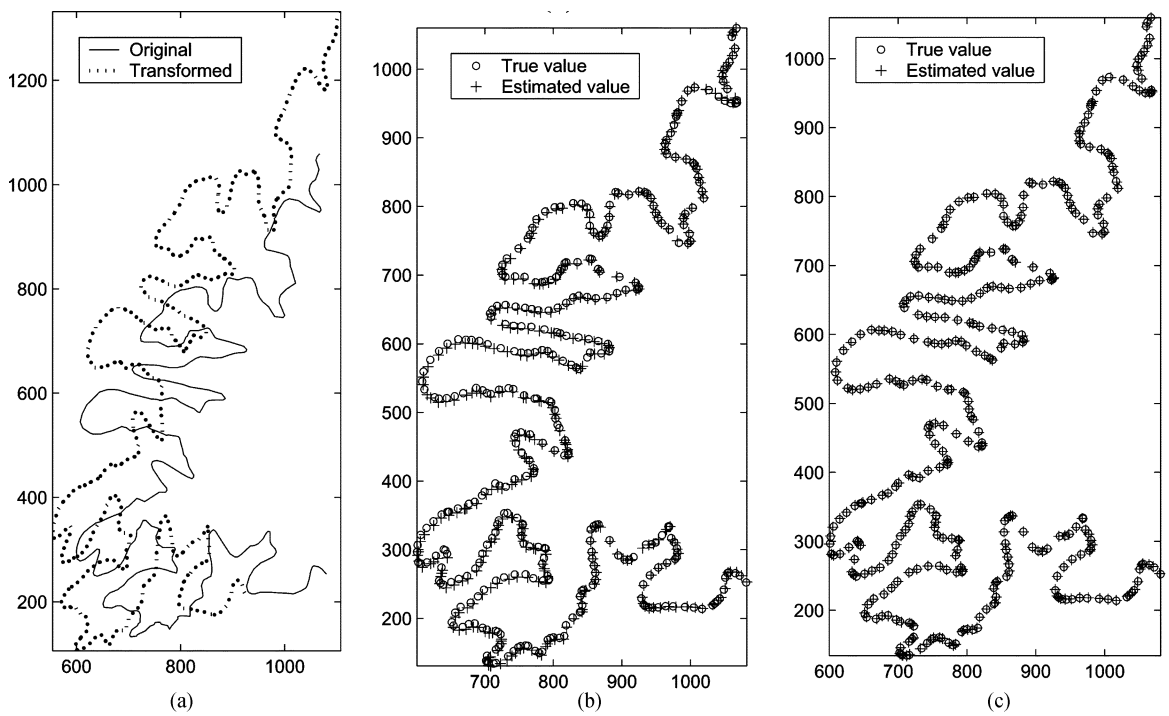

Fig. 9. Register a curve using the proposed IAM algorithm. (a) Original curve and a fingerprinted curve undergone vector-raster conversion and affine transformations. (b) Estimated sample points after one iteration. (c) Estimated sample points after 15 iterations.

original and the test curves are represented in raster format since a vector curve can be rendered as a raster curve by interpolation and that the sample points used in fingerprinting the original curve are known to the detector. The problem can be formulated as follows:

Given an original raster curve with a set of sample points $(\mathbf{p}_x, \mathbf{p}_y)$ and a test raster curve $(\tilde{\mathbf{r}}_x, \tilde{\mathbf{r}}_y)$, we register the test curve with the original curve and extract the control points of the test curve. Both transform parameters $(\mathbf{a}_x, \mathbf{a}_y)$ (or equivalently $(\mathbf{g}_x, \mathbf{g}_y)$) and a set of sample points $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y)$ corresponding to the one used in the fingerprint embedding must be found from the test curve.

B. Iterative Alignment-Minimization (IAM) Algorithm

To align the test curves with the original curves and in the mean time identify the point correspondence of the sample points, we develop an IAM algorithm. As shown in Fig. 7, the IAM algorithm consists of three main steps and the latter two steps will be executed iteratively. We first obtain an initial estimation of the test sample points. With the estimated point correspondence, we then perform super-curve alignment to estimate both the transform parameters and the control points of the test curve. With the estimated transform parameters, we refine the estimation of point correspondence through a nearest-neighbor rule. A detailed block diagram of the proposed IAM-based fingerprint detection is shown in Fig. 8.

1) Step 1—Initial Estimation of Sample Points on the Test Curve: We initialize the sample points $\left(\tilde{\mathbf{p}}_x^{(1)}, \tilde{\mathbf{p}}_y^{(1)}\right)$ on the test curve using the following simple estimator. Let M and $\tilde{M}$ be the number of points on the original and the test raster curves, respectively. From the known indices $\mathbf{J} = [j_0, j_1, j_2, \ldots, j_m]$ of the original curve's $m+1$ sample points, where $j_0 < j_1 <$

$j_2 < \cdots < j_m$ are integers ranging from 0 to $M-1$, we estimate the indices of the test curve's $m+1$ sample points by $\tilde{\mathbf{J}} = \text{round}\left((\tilde{M}-1/M-1)\cdot \mathbf{J}\right)$. Using this estimated index vector $\tilde{\mathbf{J}}$, we can identify their corresponding sample points from the test curve and take them as the initial estimate.

2) Step 2—Curve Alignment With the Estimated Sample Points: Given the estimated point correspondence with sample points $\left(\tilde{\mathbf{p}}_x^{(i)}, \tilde{\mathbf{p}}_y^{(i)}\right)$ for the test curve in the ith iteration, we apply the curve alignment method in [35] to estimate the transform parameters and the control points of the test curve. More specifically, let the transform parameters from View-I (the original curve) to View-II (the test curve) be $\left(\mathbf{a}_x^{(i)}, \mathbf{a}_y^{(i)}\right)$. The sample points on the test curve can be transformed back to View-I by $\left(\mathbf{g}_x^{(i)}, \mathbf{g}_y^{(i)}\right)$. We then fit these transformed test sample points as well as the original sample points with a single B-spline curve (referred to as a super-curve in [35]), and we search for both the transform parameters $\left(\hat{\mathbf{g}}_x^{(i)}, \hat{\mathbf{g}}_y^{(i)}\right)$ and the B-spline control points $\left(\hat{\mathbf{c}}_x^{(i)}, \hat{\mathbf{c}}_y^{(i)}\right)$ to minimize the fitting error

$$f\left(\hat{\mathbf{c}}_x^{(i)}, \hat{\mathbf{c}}_y^{(i)}, \hat{\mathbf{g}}_x^{(i)}, \hat{\mathbf{g}}_y^{(i)}\right) = \left\| \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix} \hat{\mathbf{c}}_x^{(i)} - \begin{bmatrix} \mathbf{p}_x \\ \tilde{\mathbf{P}}^{(i)} \hat{\mathbf{g}}_x^{(i)} \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix} \hat{\mathbf{c}}_y^{(i)} - \begin{bmatrix} \mathbf{p}_y \\ \tilde{\mathbf{P}}^{(i)} \hat{\mathbf{g}}_y^{(i)} \end{bmatrix} \right\|^2 \quad (21)$$

where $\tilde{\mathbf{P}}^{(i)} \triangleq \begin{bmatrix} \tilde{\mathbf{p}}_x^{(i)} & \tilde{\mathbf{p}}_y^{(i)} & \mathbf{1} \end{bmatrix}$ and $\mathbf{1}$ is a column vector with all 1s. The partial derivatives of the fitting error function with respect to $\hat{\mathbf{g}}_x^{(i)}$, $\hat{\mathbf{g}}_y^{(i)}$, $\hat{\mathbf{c}}_x^{(i)}$, and $\hat{\mathbf{c}}_y^{(i)}$ being zero is the necessary condition for the solution to this optimization problem. Thus, we obtain an estimate of the transform parameters and the B-spline control points as

$$\begin{cases} \hat{\mathbf{g}}_x^{(i)} = \mathbf{C}^{(i)}\mathbf{D}^{(i)}\mathbf{p}_x, \quad \hat{\mathbf{g}}_y^{(i)} = \mathbf{C}^{(i)}\mathbf{D}^{(i)}\mathbf{p}_y \\ \hat{\mathbf{c}}_x^{(i)} = \mathbf{D}^{(i)}\mathbf{p}_x, \quad \hat{\mathbf{c}}_y^{(i)} = \mathbf{D}^{(i)}\mathbf{p}_y \end{cases}$$
$$\text{where} \begin{cases} \mathbf{C}^{(i)} \triangleq \left(\tilde{\mathbf{P}}^{(i)T}\tilde{\mathbf{P}}^{(i)}\right)^{\dagger} \tilde{\mathbf{P}}^{(i)T}\mathbf{B} \\ \mathbf{D}^{(i)} \triangleq \left(2\mathbf{B}^T\mathbf{B} - \mathbf{B}^T\tilde{\mathbf{P}}^{(i)}\mathbf{C}^{(i)}\right)^{\dagger}\mathbf{B}^T \end{cases}. \quad (22)$$

The estimated control points $\left(\hat{\mathbf{c}}_x^{(i)}, \hat{\mathbf{c}}_y^{(i)}\right)$ can then be used to estimate the fingerprint sequence and further compute the detection statistic $Z^{(i)}$, as described in Section II.

3) Step 3—Refinement of Sample Point Estimation on the Test Curve: Given the estimated transform parameters $\left(\hat{\mathbf{g}}_x^{(i)}, \hat{\mathbf{g}}_y^{(i)}\right)$, we align the test raster curve $(\tilde{\mathbf{r}}_x, \tilde{\mathbf{r}}_y)$ with the original curve by transforming it to View-I

$$\begin{cases} \tilde{\mathbf{r}}_{x,I}^{(i)} = [\tilde{\mathbf{r}}_x \quad \tilde{\mathbf{r}}_y \quad \mathbf{1}]\,\hat{\mathbf{g}}_x^{(i)} \\ \tilde{\mathbf{r}}_{y,I}^{(i)} = [\tilde{\mathbf{r}}_x \quad \tilde{\mathbf{r}}_y \quad \mathbf{1}]\,\hat{\mathbf{g}}_y^{(i)} \end{cases}. \quad (23)$$

As the fingerprinted sample points $(\mathbf{B}(\mathbf{c}_x + \alpha\mathbf{w}_x), \mathbf{B}(\mathbf{c}_y + \alpha\mathbf{w}_y))$ are located at the neighborhood of their corresponding unmarked version $(\mathbf{B}\mathbf{c}_x, \mathbf{B}\mathbf{c}_y)$, we apply a *nearest-neighbor* rule to refine the estimation of the test curve's sample points. More specifically, for each point of $(\mathbf{B}\mathbf{c}_x, \mathbf{B}\mathbf{c}_y)$, we find its closest point from the aligned test raster curve $(\tilde{\mathbf{r}}_{x,I}^{(i)}, \tilde{\mathbf{r}}_{y,I}^{(i)})$ and then denote the collection of these closest points as $(\tilde{\mathbf{p}}_{x,i}^{(i+1)}, \tilde{\mathbf{p}}_{y,i}^{(i+1)})$. These nearest neighbors form a refined estimate of the test sample points in View-I and are then transformed with parameters $(\hat{\mathbf{a}}_x^{(i)}, \hat{\mathbf{a}}_y^{(i)})$ back to View-II as a new estimate of the test sample points

$$\begin{cases} \tilde{\mathbf{p}}_x^{(i+1)} = \begin{bmatrix} \tilde{\mathbf{p}}_{x,I}^{(i+1)} & \tilde{\mathbf{p}}_{y,I}^{(i+1)} & \mathbf{1} \end{bmatrix} \hat{\mathbf{a}}_x^{(i)} \\ \tilde{\mathbf{p}}_y^{(i+1)} = \begin{bmatrix} \tilde{\mathbf{p}}_{x,I}^{(i+1)} & \tilde{\mathbf{p}}_{y,I}^{(i+1)} & \mathbf{1} \end{bmatrix} \hat{\mathbf{a}}_y^{(i)}. \end{cases} \quad (24)$$

After this update, we increase i and go back to Step 2. The iteration will continue until convergence or for an empirically determined number of times. A total of 15 rounds of iterations are used in our experiments.

C. Detection Example and Discussions

We present a detection example employing the proposed IAM algorithm on a curve taken from a topographic map. The original curve and a fingerprinted curve having undergone vector-raster conversion and some geometric transformations are shown in Fig. 9(a). The original curve consists of 367 vector points, which are used as sample points to estimate a set of 200 control points for data embedding. Then, a fingerprinted curve with 367 vector points is generated, rendered to be a raster curve, and affinely transformed. After the vector-raster conversion, the point correspondence is no longer directly available from the raster curve representation. We apply the IAM algorithm to align the test curve with the original one and to estimate the correspondence between sample points. The estimated sample points for the test curve after one iteration and 15 iterations are shown in Fig. 9(b) and (c), respectively. We can see that initially the estimated values deviate from the true values by a nontrivial amount, whereas after 15 iterations, the estimated values converge to the true values. We plot the six estimated transform parameters for each iteration in Fig. 10(a), which shows an accurate registration by the proposed IAM algorithm after half of a dozen iterations. Upon convergence, we use the estimated control points $\left(\hat{\mathbf{c}}_x^{(i)}, \hat{\mathbf{c}}_y^{(i)}\right)$ to perform detection with the fingerprint involved. The high detection statistic value shown in Fig. 10(b) suggests the positive identification of the correct fingerprint by using the proposed IAM algorithm.

The computation time for this experiment is as follows. The IAM algorithm is implemented in Matlab 6.5 and tested on a Pentium-4 2.0 GHz PC with 512M RAM. Each iteration of the algorithm takes about 0.5 of a second, and the total 15 iterations plus the initialization take 7.61 seconds. Together with the 0.56 sec required for computing Z statistics with 1000 fingerprint sequences, the total duration of the detection process is 8.17 sec.

The above example shows that through the IAM algorithm, we can register the test curve with the original unmarked curve and extract the fingerprinted control points with high accuracy. With good estimation of affine transform parameters, our data embedding method for curves is resilient to combinations of scaling, rotation, translation, and shearing. The explicit estimation of point correspondence also provides resilience to the vector-raster and vector-raster-vector conversions. In the vector-raster conversion case, a fingerprinted curve stored in vector format is rendered as a raster curve, and thus the point correspondence is no longer directly available from the raster curve representation. In the vector-raster-vector conversion case, the intermediate raster curve is converted to a vector curve with a new set of vector points that are likely to be different from the

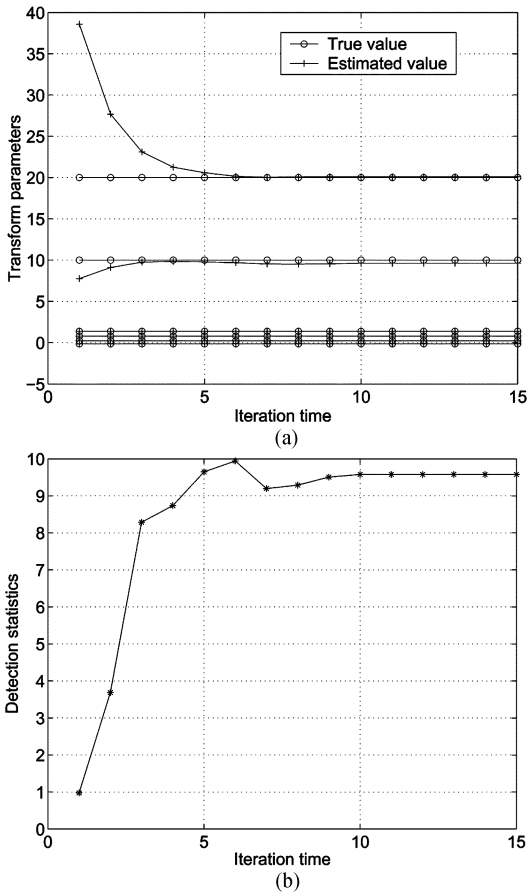

Fig. 10. Convergence results on estimated transform parameters and detection statistics for the example curve in Fig. 9 using the proposed IAM algorithm. (a) Estimated transform parameters for each iteration. (b) Fingerprint detection statistic for each iteration.

initial vector points prior to the conversion, even though there is little visual difference between these two vector curves. Again, the point correspondence is likely to get corrupted by this conversion, and accurate estimation of point correspondence is a necessary step for the successful detection of the fingerprint. With robustness resulting from the spread spectrum embedding in B-spline control points and the IAM algorithm, our curve fingerprinting approach can resist a number of challenging attacks and distortions. For example, the distortion from printing-and-scanning involves both vector-raster rendering and a certain amount of rotation, scaling, and translation; a fingerprinted curve in vector format may be rendered as a raster image and then affinely transformed before reaching the detector; in the collusion scenario, colluders may construct a colluded copy, print it out, and then distribute it out of the allowed domain. In Section IV, we use our curve-based data hiding approach to fingerprint topographic maps and demonstrate the robustness of our approach against various attacks and distortions.

IV. Experimental Results for Map Fingerprinting

We now present experimental results of the proposed curve fingerprinting algorithm in the context of tracing and tracking topographic maps. A topographic map provides a two-dimensional representation of the earth's 3-D surface. Vertical elevation is shown with contour lines (also known as level lines) to represent the earth's surfaces that are of equal altitude. Contour lines in topographic maps often exhibit a considerable amount of variations and irregularities, prompting the need for nonuniform sampling of curve points in the parametric modeling of the contours. We will first examine the fidelity of the fingerprinted map, and then evaluate the robustness of the fingerprints against collusion, cropping, geometric transformations, format conversions, point deletion, curve smoothing, printing-and-scanning, and some combinations of these distortions.

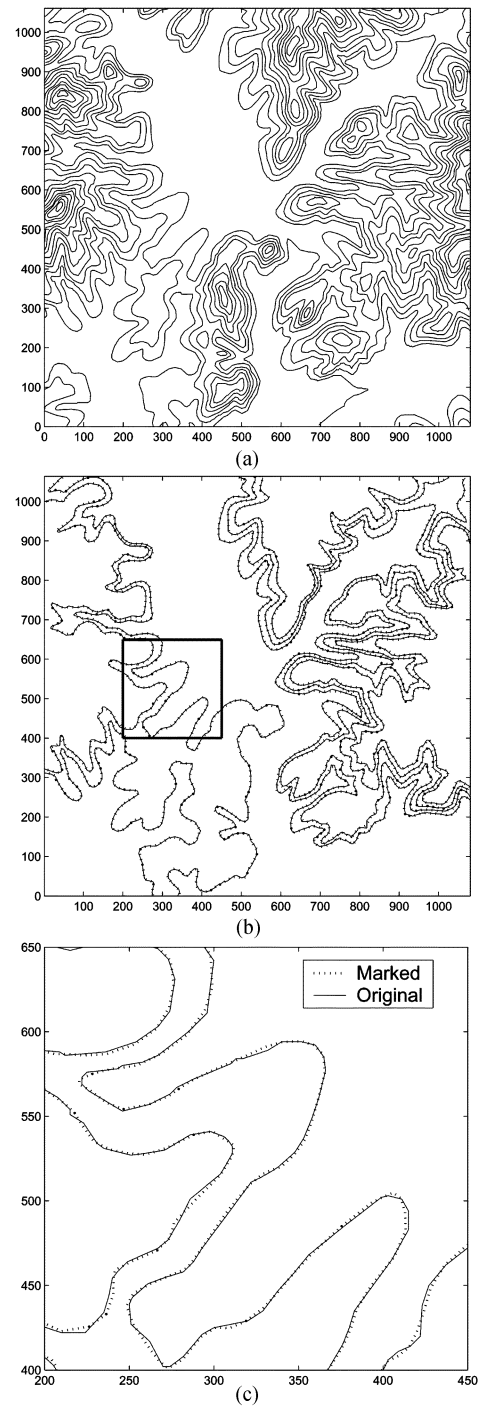

Fig. 11. Fingerprinting topographic maps. (a) Original map. (b) Original and fingerprinted curves overlaid with each other. (c) Zoomed-in view of (b).

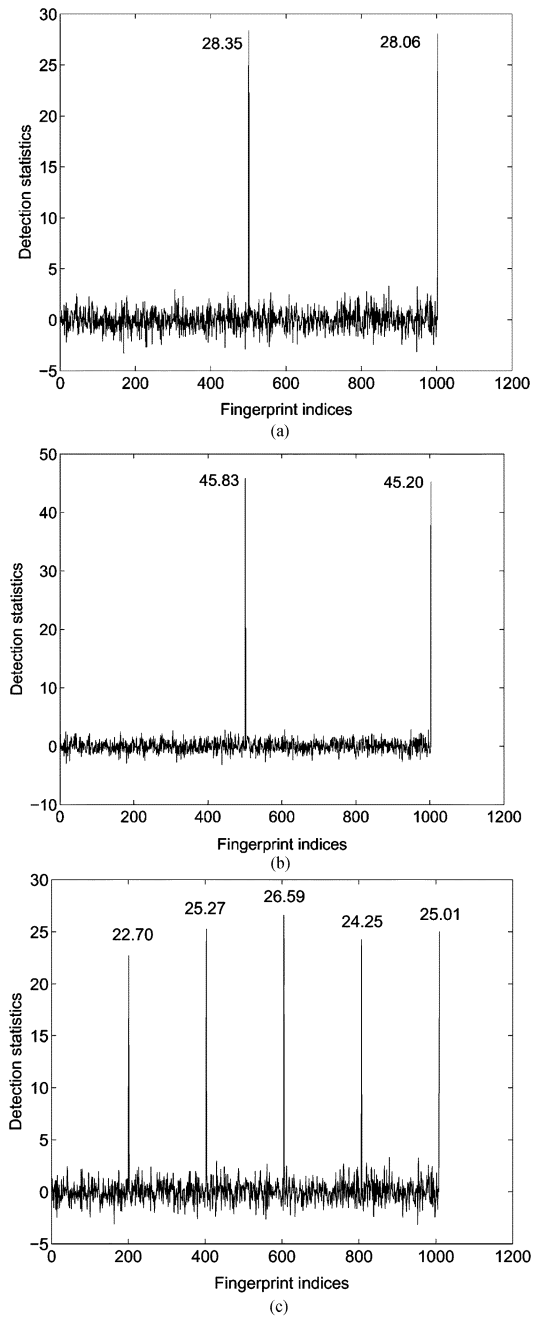

Fig. 12. Collusion test on fingerprinted vector maps. (a) Two-user random interleaving collusion. (b) Two-user averaging collusion. (c) Five-user averaging collusion.

1) Fingerprinted Topographic Maps: A 1100×1100 topographic vector map obtained from http://www. ablesw.com is used in our experiment. Starting with the original map shown in Fig. 11(a), we mark nine curves that are sufficiently long. For each of these nine curves, a set of nonuniformly spaced vector points is given by the original dataset. We directly use these points as sample points and determine their indexing values according to the curvature-based rule presented in Section II-D. A total of 1331 control points are used to carry the fingerprint. In Fig. 11(b), we overlay the nine original curves and the corresponding marked curves using solid lines and dotted lines, respectively. To help illustrate the fidelity of our method, we enlarge a portion of the overlaid image in Fig. 11(c). We can see that the fingerprinted map preserves the geospatial information in the original map with high precision. The perturbation can be adapted to be compliant with cartographic industry standards and/or the need of specific applications.

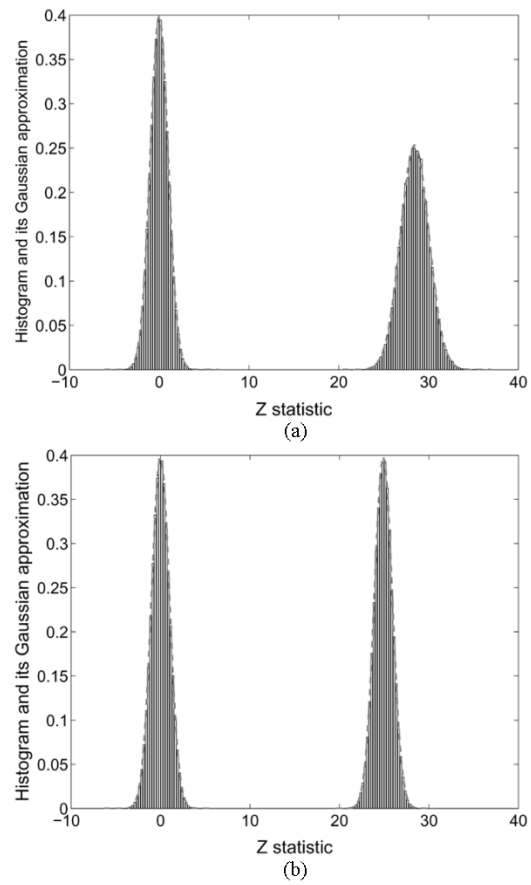

Fig. 13. Histogram and Gaussian approximation of the Z detection statistics for collusion attacks. (a) Two-user random interleaving collusion with Gaussian approximations $\mathcal{N}(0, 1.00)$ and $\mathcal{N}(28.45, 2.48)$. (b) Five-user averaging collusion with Gaussian approximations $\mathcal{N}(0, 1.00)$ and $\mathcal{N}(24.90, 1.00)$.

2) Resilience to Collusion: To demonstrate the resistance of the proposed method against collusion, we present in Fig. 12 the detection statistics under two different types of collusion attacks. Fig. 12(a) shows the collusion results under a random interleaving attack, where the control points for each curve are equiprobably taken from two differently fingerprinted maps. The collusion attack for Fig. 12(b) and (c) is known as averaging, where the coordinates of the corresponding control points from two and five differently fingerprinted maps are averaged, respectively. We assume the correct point correspondence is available in this test, and the cases with unknown point correspondence will be addressed in the later subsections. As we can see from the detection statistics, the embedded fingerprints from all contributing users survive the collusion attacks and are identified with high confidence. For both two-user random interleaving collusion and five-user averaging collusion, we use 20 different sets of fingerprint sequences to evaluate the detection performance, using a similar experiment setup to the one discussed in Section II-E. The histograms and the Gaussian approximations in Fig. 13 show

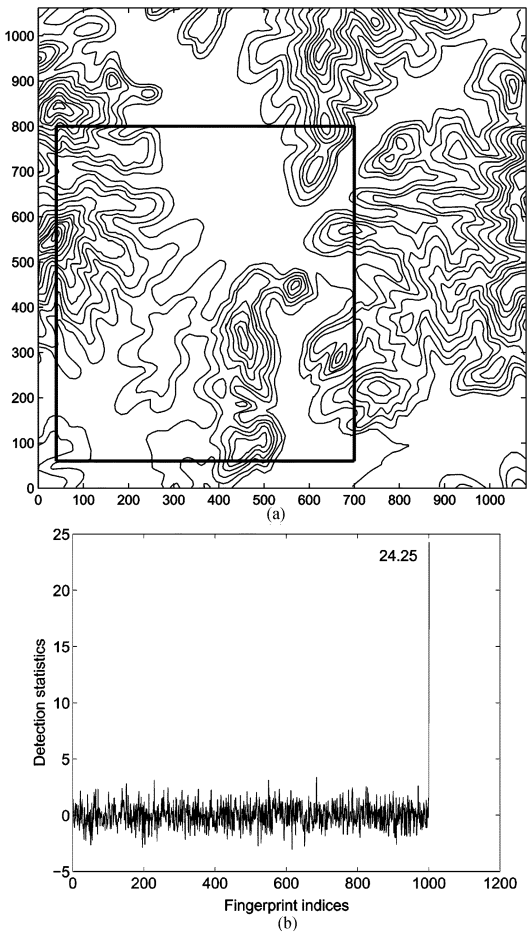

Fig. 14. Cropping test on fingerprinted vector maps. (a) Fingerprinted map with the cropping area. (b) Z statistics.

a very good separation between the Z values for the presence and absence of true fingerprints. Compared with the two-user averaging collusion, the two-user random interleaving leads to more substantial coordinate changes in control points when a detector performs B-spline parametrization, and such coordinate changes may follow a distribution different from i.i.d. Gaussian. This is reflected by a reduced mean and a nonunit variance for the fingerprint presence case.

3) Resilience to Cropping: As shown in Fig. 14(a), we crop an area of a fingerprinted vector map and use it as the test map. Among the nine curves used for carrying the fingerprint, only two curves are retained with sufficiently large size. Using the original map as a reference, we perform detection on these two retained segments and obtain the detection result shown in Fig. 14(b). As we can see, the detection statistic with the correct fingerprint is still high enough so that its corresponding user can be identified with high confidence.

4) Resilience to Affine Transformations on Vector Maps: To demonstrate the resilience of our approach to a substantial amount of affine transformations, we take a fingerprinted vector map and apply a combination of rotation, scaling, and translation. More specifically, we rotate it by -30°, then scale it by 120% or 80% in the X and Y directions, respectively, followed by 100- and 200-pixel translation in the X and Y directions, respectively. The resulting vector map is rendered in Fig. 15(a).

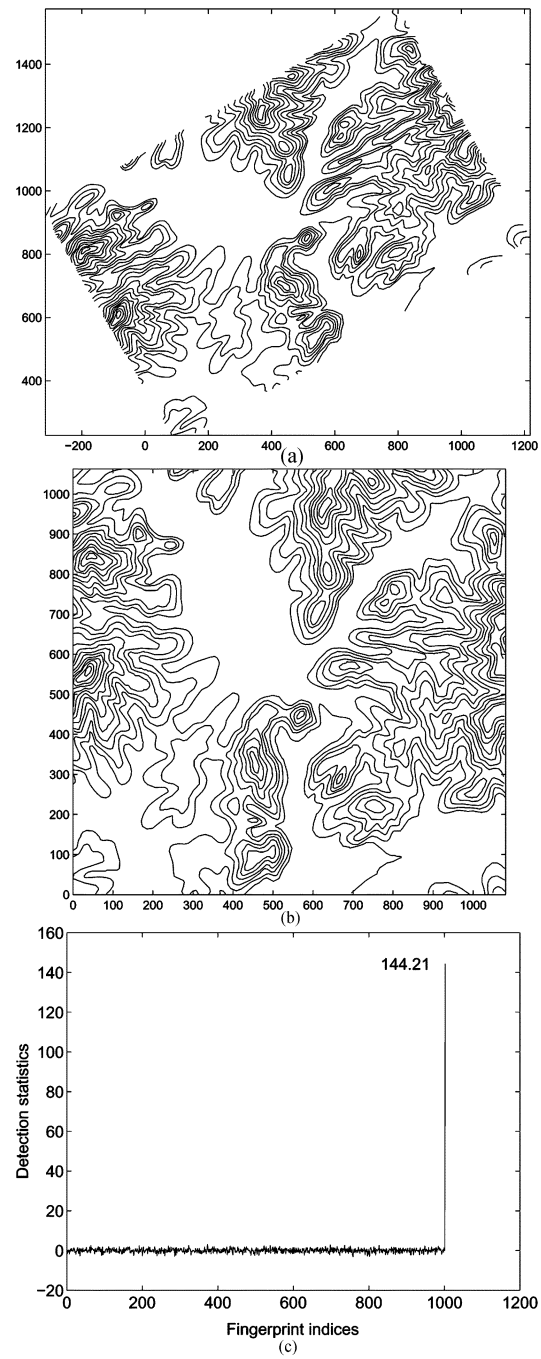

Fig. 15. Affine transformation test on fingerprinted vector maps. (a) Test map. (b) Aligned map. (c) Z statistics.

In this test, we assume that correct point correspondence is available. Thus, the super-curve alignment method can be directly applied to register the original and the test curves and to extract the control points. The registered vector map is shown in Fig. 15(b), and Fig. 15(c) shows the detection statistics. We can see that the embedded fingerprint can survive affine transformations, and the detection statistic with the correct fingerprint is high after the registration.

5) Resilience to Vector-Raster Conversion: We now examine the resilience to vector-raster conversion coupled with possible affine transformations. A fingerprinted vector map

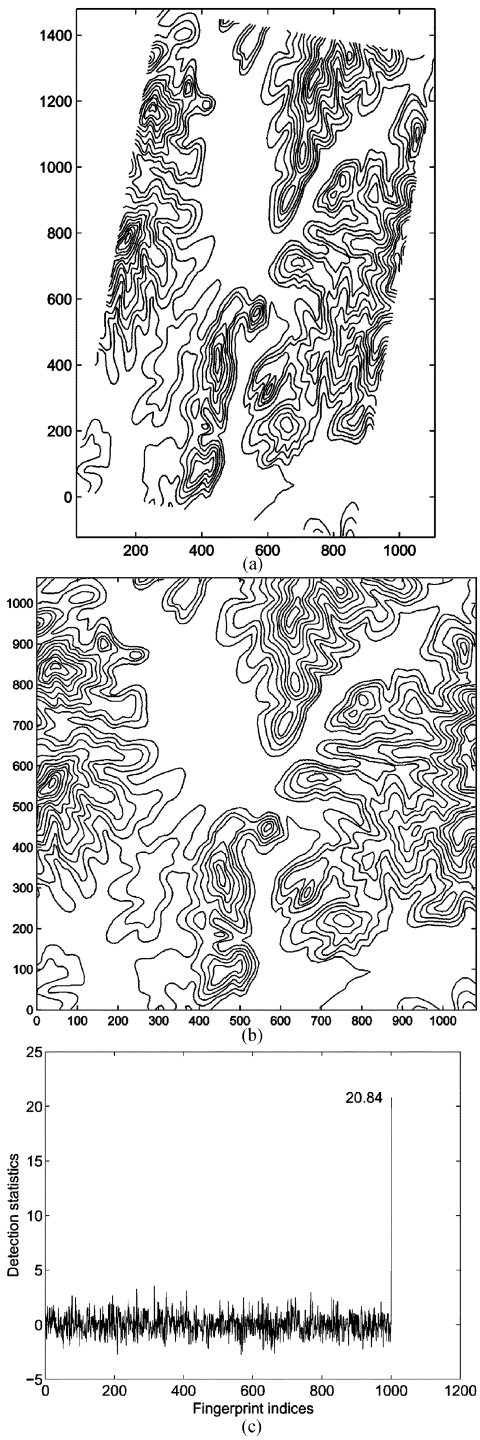

Fig. 16. Affine transformation test on fingerprinted raster maps. (a) Test map. (b) Aligned map. (c) Z statistics.

after raster rendering as a 1100×1100 image and affine transformations is shown in Fig. 16(a). The affine transformation consists of 10° rotation, 80% and 140% scaling in the X and Y directions, respectively, and 10- and 20-pixel translation in the X and Y directions, respectively. As the point correspondence is no longer directly available after the vector-raster conversion, we apply the proposed IAM algorithm to estimate the transform parameters and to locate the sample points on test curves corresponding to those used in the embedding. After

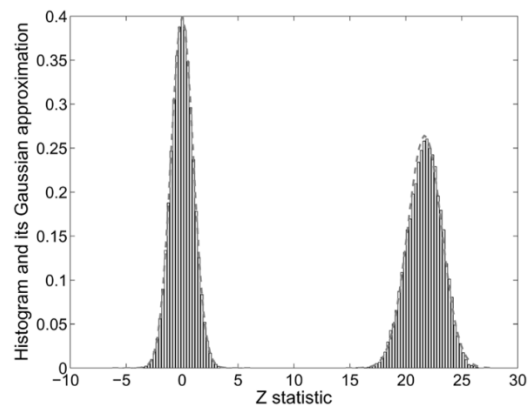

Fig. 17. Histogram and Gaussian approximation $\mathcal{N}(0, 1.00)$ and $\mathcal{N}(21.69, 2.28)$ of the Z detection statistics for the combined vector-raster conversion and geometric transformations.

15 iterations, we get the registered raster map as shown in Fig. 16(b) and the detection statistics as shown in Fig. 16(c). The detection statistic results suggest that the embedded fingerprint is identified with high confidence. Using the same settings on the computing system as in Section III-C, we measure the detection time required for this transformed raster map. The total duration of the detection process is 42.43 sec, including 40.05 sec for curve registration and fingerprint extraction from nine curves and 2.38 sec for evaluating Z statistics with 1000 fingerprint sequences.

Similar to the collusion test, we plot in Fig. 17 the histogram and the Gaussian approximation of the Z statistics for this vector-raster conversion test combined with geometric transformations. The transform parameters are randomly selected from the following ranges with a uniform distribution: $-20 \sim +20$ degrees of rotation, 60%–140% scaling in the X and Y directions, and 20–40 pixels' translation in the X and Y directions. As the errors from registration and resampling are not always i.i.d. Gaussian distributed, we observe a variance larger than 1 for the Z values in the fingerprint presence case.

6) Resilience to Vector-Raster-Vector Conversion: To demonstrate the resilience of our method to vector-raster-vector conversion, we first render a fingerprinted vector map as a raster map. Then, we uniformly sample it to obtain a new vector map, which has the same number of vector points as prior to the conversion but at different sampling locations. In the detection process, we first render this "new" vector map as a raster map by linear interpolation and then apply our IAM algorithm. From the high detection statistic shown in Fig. 18(a), we can see that our approach is robust against the vector-raster-vector attack.

7) Resilience to Point Deletion in Vector and Raster Maps: As we have seen throughout the paper, traitor tracing applications usually involve adversaries who have strong incentives to remove fingerprints. Attackers may delete a certain number of points from a fingerprinted vector/raster map while keeping similar shapes of its contour lines. Detection statistics after point deletion in a fingerprinted vector and raster map, respectively, are shown in Fig. 18(b) and (c). For the vector map, 20% of points are randomly chosen and removed from each fingerprinted curve, whereas in the raster map 70% of black

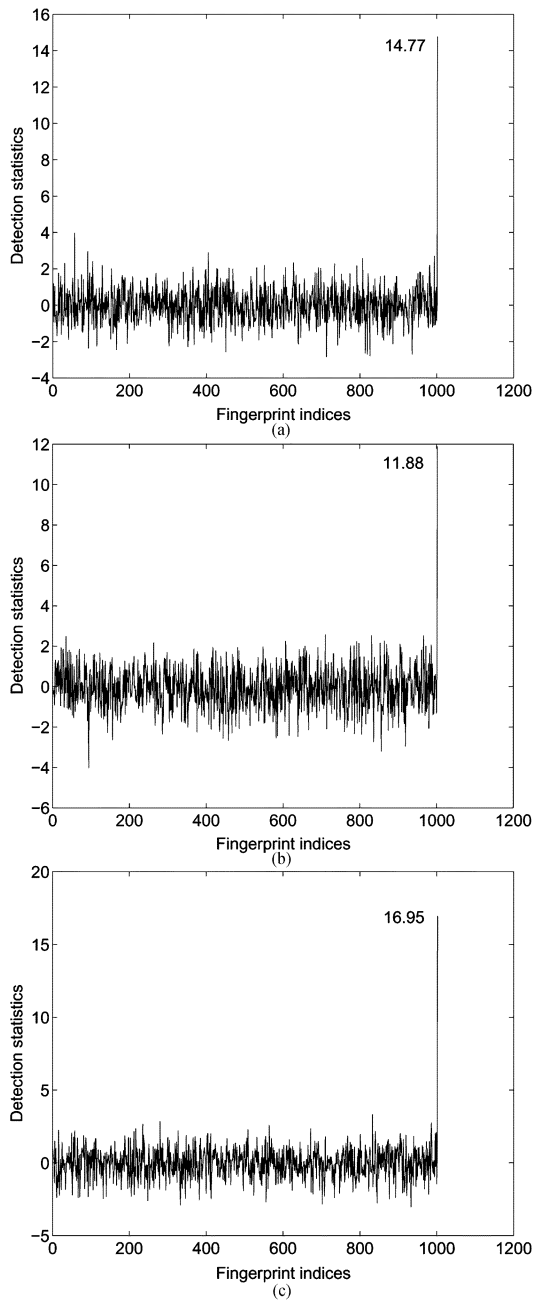

Fig. 18. Detection results after various attacks. (a) Z statistics for vector-raster-vector conversion. (b) Z statistics for point deletion in vector maps. (c) Z statistics for point deletion in raster maps.

pixels on the curve are randomly chosen and removed. We can see that the embedded fingerprints can survive point deletion applied to both vector maps and raster maps. Similar to the vector-raster-vector conversion test, linear interpolation and the IAM algorithm are used in fingerprint detection.

8) Resilience to Curve Smoothing: Similar to lowpass filtering for images, curve smoothing can be applied to topographic maps as an attempt to remove the embedded fingerprint. In order to demonstrate the resilience of the proposed method to curve smoothing, we traverse each marked curve and apply a moving average filter to it. A curve point with coordinates

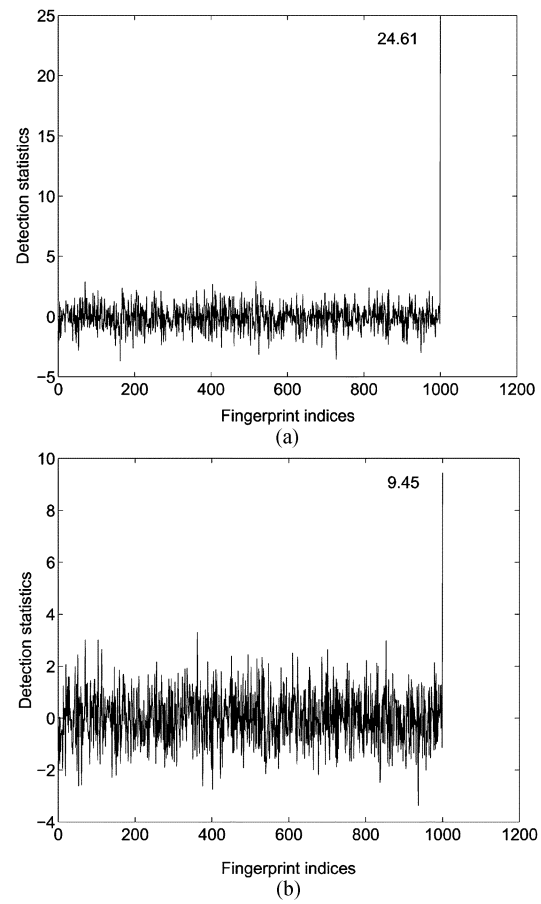

Fig. 19. Curve smoothing test on fingerprinted raster maps. (a) Z statistics for filter window size 5. (b) Z statistics for filter window size 21.

(r_{x_i}, r_{y_i}) will be replaced by a new point, whose coordinates $\left(r_{x_i}^{(s)}, r_{y_i}^{(s)}\right)$ are obtained by

$$\begin{cases} r_{x_i}^{(s)} = \frac{1}{2S+1} \sum_{j=-S}^{S} r_{x_{i+j}} \\ r_{y_i}^{(s)} = \frac{1}{2S+1} \sum_{j=-S}^{S} r_{y_{i+j}} \end{cases} \tag{25}$$

where $2S+1$ is the filter length. Finally, we apply the proposed IAM algorithm to these smoothed curves and compute the detection statistics. Two different filter lengths (5 and 21) are used in our experiments. As shown in Fig. 19, the detection statistic with the correct fingerprint under five -point averaging is 24.61 and that under 21-point averaging is 9.45. Indeed, the fingerprint is weakened by the smoothing operation, but the detection statistics are still well above the threshold for a correct positive detection. In the latter case, when a long filter is used in the smoothing attack, some visual details have been lost from the curves. This study shows that the proposed method is robust against curve smoothing, provided that the smoothing does not severely change the shape of the curve and that the fingerprint sequence is sufficiently long to help the detector collect information for a positive detection.

9) Resilience to Printing-and-Scanning: To show the robustness of our approach against the printing-and-scanning

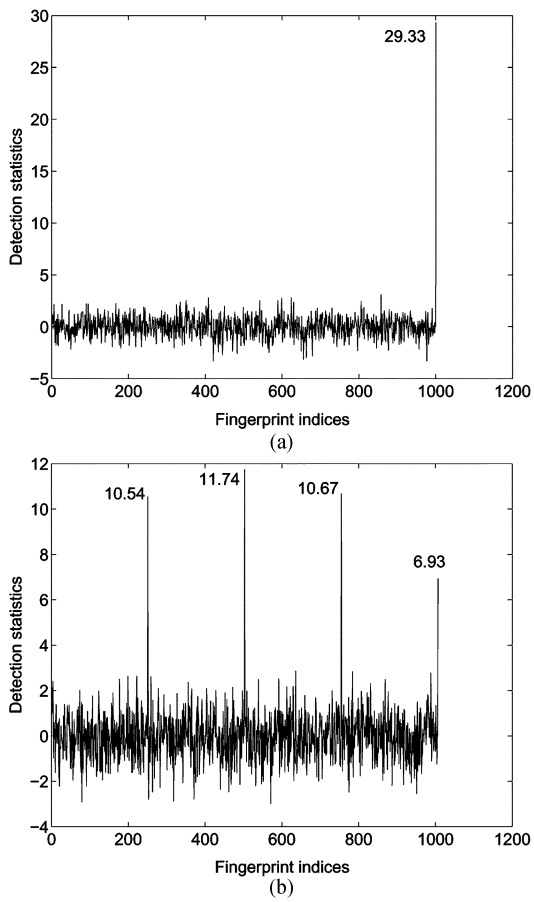

Fig. 20. Printing-and-scanning test. (a) Z statistics for printing-and-scanning. (b) Z statistics for four-user averaging collusion combined with printing-and-scanning.

attack, we render a fingerprinted vector map by taking a screen shot of its display in Matlab, print out the image using a HP laser printer, and then scan it back as a binary image using a Canon scanner with 360 dpi resolution. Preprocessing before detection includes a thinning operation to extract one-pixel wide skeletons from the scanned curves that are usually several-pixel wide after high-resolution scanning. As we can see from the detection results in Fig. 20(a), the fingerprint survives the printing-and-scanning test and gives reliable positive detection with the detection statistic much higher than the detection threshold.

To further examine our approach under the combined attack of collusion and printing-and-scanning, we first generate a colluded map by averaging coordinates of the corresponding control points from four users' fingerprinted maps, then render and print it out, and scan it back as a binary image. From the detection statistics in Fig. 20(b), we can see that the embedded fingerprints from all the four colluders can be correctly identified after this combination of attacks involving collusion, vector-raster conversion, filtering, and affine transformations.

V. Conclusions

In this paper, we have presented a new data hiding algorithm for curves by parameterizing a curve using the B-spline model and adding spread spectrum sequences to curve parameters. In conjunction with the basic embedding and detection techniques, we have proposed an iterative alignment-minimization algorithm to allow for robust fingerprint detection under unknown geometric transformations and in the absence of explicit point correspondence. We have demonstrated the fidelity of our method as well as its robustness against collusion, cropping, affine transformations, vector-raster and vector-raster-vector conversions, point deletion, curve smoothing, printing-and-scanning, and their combinations. Our work has shown the feasibility of the proposed algorithm in fingerprinting applications for tracing and tracking topographic maps as well as writings/drawings from pen-based inputs. The protection of such documents has increasing importance to the emerging digital operations in government, military, and commercial agencies.

Acknowledgment

The work in this paper was inspired by discussions with Prof. B. Liu of Princeton University and Dr. M. Xia of Microsoft Corporation, who also kindly provided a preprint of the super-curve registration paper [35]. D. Cardy of the University of Maryland, College Park, provided helpful input on the curvature-based sampling method in Section II-D. In addition, the authors would like to thank the Guest Editor Prof. P. Moulin and the anonymous reviewers for their comments and suggestions that have helped improve the quality and the clarity of the paper.

References

[1] H. Gou and M. Wu, "Data hiding in curves for collusion-resistant digital fingerprinting," in *Proc. IEEE ICIP*, 2004.
[2] ——, "Robust digital fingerprinting for curves," in *Proc. IEEE ICASSP*, vol. II, 2005, pp. 529–532.
[3] M. Wu, W. Trappe, Z. J. Wang, and K. J. R. Liu, "Collusion resistant fingerprinting for multimedia," *IEEE Signal Process. Mag.*, vol. 21, no. 2, pp. 15–27, Mar. 2004.
[4] I. Cox, J. Bloom, and M. Miller, *Digital Watermarking*. San Franicsco, CA: Morgan Kaufmann, 2001.
[5] H. Chang, T. Chen, and K. Kan, "Watermarking 2D/3D graphics for copyright protection," in *Proc. IEEE ICASSP*, 2003, pp. 720–723.
[6] M. Barni, F. Bartolini, A. Piva, and F. Salucco, "Robust watermarking of cartographic images," *EURASIP J. Applied Signal Process.*, vol. 2, pp. 197–208, 2002.
[7] V. Solachidis and I. Pitas, "Watermarking polygonal lines using Fourier descriptors," *IEEE Computer Graphics Appl.*, vol. 24, no. 3, pp. 44–51, May/Jun. 2004.
[8] R. Ohbuchi, H. Ueda, and S. Endoh, "Watermarking 2D vector maps in the mesh-spectral domain," *Proc. Shape Modeling Int. Conf.*, 2003.
[9] E. Koch and J. Zhao, "Embedding robust labels into images for copyright protection," in *Proc. Int. Congr. Intellectural Property Rights Specialized Inf., Knowledge New Technol.*, 1995.
[10] M. Wu, E. Tang, and B. Liu, "Data hiding in digital binary image," in *Proc. IEEE ICME*, 2000, pp. 393–396.
[11] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 528–538, Aug. 2004.
[12] K. Matsui and K. Tanaka, "Video-steganography: How to secretly embed a signature in a picture," *Proc. IMA Intellectural Property Project*, vol. 1, no. 1, 1994.
[13] N. F. Maxemchuk and S. Low, "Making text documents," in *Proc. IEEE ICIP*, 1997.
[14] R. Ohbuchi, H. Masuda, and M. Aono, "A shape-preserving data embedding algorithm for NURBS curves and surfaces," in *Proc. CGI*, 1999, pp. 180–187.

[15] J. J. Lee, N. I. Cho, and J. W. Kim, "Watermarking for 3D NURBS graphic data," *Proc. IEEE MMSP*, pp. 304–307, 2002.
[16] J. J. Lee, N. I. Cho, and S. U. Lee, "Watermarking algorithms for 3D NURBS graphic data," *EURASIP J. Appl. Signal Process.*, no. 14, pp. 2142–2152, Oct. 2004.
[17] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," in *Proc. Comput. Graphics*, 1999, pp. 49–56.
[18] K. H. Ko, T. Maekawa, N. M. Patrikalakis, H. Masuda, and F.-E. Wolter, "Shape intrinsic fingerprints for free-form object matching," in *Proc. 8th ACM Symp. Solid Modeling Appl.*, 2003, pp. 196–207.
[19] I. Cox, J. Killian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.
[20] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[21] G. E. Farin, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, Fourth ed. New York: Academic, 1997.
[22] Z. J. Wang, M. Wu, H. Zhao, W. Trappe, and K.J. R. Liu, "Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation," *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 804–821, Jun. 2005.
[23] H. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu, "Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting," *IEEE Trans. Image Process.*, vol. 14, no. 5, pp. 646–661, May 2005.
[24] H. D. Brunk, *An Introduction to Mathematical Statistics*. Boston, MA: Ginn, 1960.
[25] H. Stone, "Analysis of Attacks on Image Watermarks With Randomized Coefficients," NEC Res. Inst., Tech. Rep. 96-045, 1996.
[26] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1069–1087, Apr. 2003.
[27] M. Wu and B. Liu, "Data hiding in image and video: Part-I fundamental issues and solutions," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 685–695, Jun. 2003.
[28] D. Kirovski, H. S. Malvar, and Y. Yacobi, "Multimedia content screening using a dual watermarking and fingerprinting system," in *Proc. ACM Multimedia*, 2002.
[29] Z. Huang and F. Cohen, "Affine-invariant B-spline moments for curve matching," *IEEE Trans. Image Process.*, vol. 5, no. 10, pp. 1473–1480, Oct. 1996.
[30] C. Cabrelli and U. Molter, "Automatic representation of binary images," *IEEE Trans. Pattern Anal.Machine Intell.*, vol. 12, no. 12, pp. 1190–1196, Dec. 1990.
[31] H. S. M. Coxeter, *Introduction to Geometry*, 2nd ed. New York: Wiley, 1969.
[32] F. S. Cohen and J. Y. Wang, "Modeling image curves using invariant 3-D object curve models—A path to 3-D recognition and shape estimation from image contours," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 1, pp. 1–12, Jan. 1994.
[33] E. Belogay, C. Cabrelliay, U. Molter, and R. Shonkwiler, "Calculating the Hausdorff distance between curves," *Inf. Process. Lett.*, vol. 64, pp. 17–22, 1997.
[34] J. Lubin, J. A. Bloom, and H. Cheng, "Robust, content-dependent, high-fidelity watermark for tracking in digital cinema," in *Proc. SPIE*, vol. 5020, 2003, pp. 536–545.
[35] M. Xia and B. Liu, "Image registration by 'super-curves'," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 720–732, May 2004.

Hongmei Gou (S'05) received the B.E. and D.E. degrees in automatic control and industrial engineering from Tsinghua University, Beijing, China, in 1997 and 2002, respectively. She is currently pursuing the Ph.D. degree in signal processing and communications with the Electrical and Computer Engineering Department, University of Maryland, College Park.

Previously, she was a research intern at the IBM China Research Lab, Beijing, in 2001 and with the R&D Department, Pitney Bowes, Shelton, CT, in 2005. Her research interests include information security, multimedia signal processing, and biomedical signal processing.

Ms. Gou received the Outstanding College Graduate Award from Tsinghua University in 1997 and the Graduate School Fellowship from University of Maryland for the years 2002 to 2004.

Min Wu (S'95–M'01) received the B.E. degree in electrical engineering and the B.A. degree in economics (both with the highest honors) from Tsinghua University, Beijing, China, in 1996 and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1998 and 2001, respectively.

Since 2001, she has been an Assistant Professor with the Department of Electrical and Computer Engineering and the Institute of Advanced Computer Studies, University of Maryland, College Park. Previously, she was with NEC Research Institute and Signafy, Inc., Princeton, in 1998 and with Panasonic Information and Networking Laboratories, Princeton, in 1999. She served as a Guest Editor of the Special Issue on Media Security and Rights Management published in October 2004 by the *EURASIP Journal on Applied Signal Processing*. She co-authored a book *Multimedia Data Hiding* (New York: Springer-Verlag, 2003) and holds four U.S. patents on multimedia security. Her research interests include information security and forensics, multimedia signal processing, and multimedia communications.

Dr. Wu received a CAREER award from the U.S. National Science Foundation in 2002, a George Corcoran Education Award from University of Maryland in 2003, a TR100 Young Innovator Award from the Massachusetts Institute of Technology's *Technology Review Magazine* in 2004, and a Young Investigator Award from the U.S. Office of Naval Research in 2005. She is a member of the IEEE Technical Committees on Multimedia Signal Processing and on Multimedia Systems and Applications for the years 2002 to 2005 and was Publicity Chair of 2003 IEEE International Conference on Multimedia and Expo.