# DATA HIDING IN CURVES FOR COLLUSION-RESISTANT DIGITAL FINGERPRINTING

*Hongmei Gou and Min Wu*

Electrical & Computer Engineering Department, University of Maryland, College Park, U.S.A.

**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 in the coordinates of the B-spline control points. We demonstrate through experiments the robustness of the proposed data hiding algorithm against printing-and-scanning and collusions, and show its feasibility for collusion-resistant fingerprinting of topographic maps as well as writings/drawings from pen-based input devices.*

## 1. INTRODUCTION

Curve is one of the major components appearing in maps, drawings, signatures, and other documents alike. A huge amount of such documents are being brought to the digital domain owing to the popularity of scanning devices and pen-based devices (such as the TabletPC). Hiding digital watermark or other secondary data in curves can facilitate digital rights management of important documents in government, intelligence, and commercial operations. For example, trace-and-track capabilities can be provided through embedding a unique ID, referred to as a *digital fingerprint*, to each copy of a document before distributing to users to deter the unauthorized leak of classified documents outside an allowed group [1].

The documents that contain mostly curves, such as maps and handwritten notes, can be represented as binary bitmap images (raster representation) or as a set of vectors. As the existing embedding techniques often flip pixels [2] or perturb the vertices [3][4] to hide data in these documents, the correct decoding of hidden data relies heavily on the correct sampling of pixels or vertices, posing challenges in surviving D/A-A/D conversion such as printing-and-scanning. In this paper, we present a new, robust data hiding technique for curves by identifying and manipulating curve parameters. In particular, we compute the control points in the B-spline representation of curves and embed spread spectrum signals in the coordinates of the control points. As we shall see, this embedding domain can sustain printing-and-scanning as well as multi-user collusions, thus can be used for fingerprinting purposes.

The paper is organized as follows. Section 2 discusses the feature domain in which data hiding is performed and presents the embedding and detection algorithms. Experimental results on marking simple curves as well as fingerprinting real topographic maps are given in Section 3. The paper is concluded with discussions on future work.

The authors can be reached at {hmgou, minwu}@eng.umd.edu.

## 2. THE PROPOSED DATA HIDING ALGORITHM

Our proposed algorithm employs B-spline control points as the feature domain, and adopts spread spectrum embedding [5] for robustly marking the coordinates of the control points. The detection is based on correlation statistics. In the following subsections, we explain the main steps in detail.

### 2.1 Feature Extraction

B-splines are piecewise polynomial functions that provide local approximations of curves using a small number of parameters known as the *control points* [6]. Let $\mathbf{p}(t) = (x(t), y(t))$ denote a curve, where $t$ is a continuous variable. The B-spline representation of the curve can be written as

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

where $\mathbf{c}_i$ is the $i^{\text{th}}$ control point ($i = 0, \ldots n$), and $B_{i,k}(t)$ is the B-spline blending function defined as:

$$B_{i,1}(t) = \begin{cases} 1, & t_i \le t < t_{i+1} \\ 0, & otherwise \end{cases} \qquad (2)$$

$$B_{i,k}(t) = \frac{(t - t_i) B_{i,k-1}(t)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - t) B_{i+1,k-1}(t)}{t_{i+k} - t_{i+1}}, \qquad k > 1$$

Given a set of properly chosen samples on the curve, we can obtain the control points of the corresponding B-spline approximation using the least square technique. The control points form a compact set of salient features for the original curve. We then apply embedding in this feature domain.

### 2.2 Data Embedding and Detection in Control Points

The spread spectrum embedding generally offers a good tradeoff between imperceptibility and robustness, especially when the original host signal is available to the detector [5] as in most of the fingerprinting applications. We use mutually orthogonal, noise-like sequences as digital fingerprints to represent different users/IDs for trace and track purposes [1]. As each of the $n$+1 control

points has two coordinate values $x$ and $y$, the fingerprint sequence has $2(n+1)$ elements. To apply spread spectrum embedding on a curve, we add a scaled version of the fingerprint sequence $\{\mathbf{w}_i\}$ to the coordinates of a set of control points obtained from the previous subsection. This results in a set of watermarked control points $\{\mathbf{c}_i'\}$ with $\mathbf{c}_i' = \mathbf{c}_i + \alpha\mathbf{w}_i$, where $\alpha$ is a scaling factor adjusting the fingerprint strength. Then a watermarked curve $\mathbf{p}'(t)$ can be constructed by the B-spline synthesis equation (1) using the watermarked control points $\{\mathbf{c}_i'\}$.

To determine which fingerprint sequence(s) is present in a test curve, we first perform registration using the original unmarked curve that is commonly available to a detector in fingerprinting applications [1]. We then extract the control points $\{\mathbf{c}_i^{(T)}\}$ from the test curve, and compute the difference between the coordinates of the test and original control points to arrive at an estimated fingerprint sequence $\mathbf{w}_i^{(T)} = (\mathbf{c}_i^{(T)} - \mathbf{c}_i)/\alpha$. We evaluate the similarity between the estimated fingerprint sequence $\{\mathbf{w}_i^{(T)}\}$ and each fingerprint sequence in our database through a correlation-based statistic. In our work, we compute the correlation coefficient ρ and convert it to a Z-statistic by

$$Z = \log\left(\frac{1+\rho}{1-\rho}\right)\frac{\sqrt{2(n+1)-3}}{2} \qquad (3)$$

The Z-statistic has been shown to follow an approximate unit-variance Gaussian distribution with a large positive mean under the presence of a fingerprint, and a zero mean under the absence. Thus if the similarity is higher than a threshold (usually set between 3 to 6), with high probability the corresponding fingerprint sequence in the database is present in the test curve, allowing us to trace the test curve to a specific user [1].

### 2.3 Fidelity and Robustness Considerations

Estimating the control points requires a set of sample points from the original curve. When there is no abrupt change in a curve segment, uniform sampling can be used, while non-uniform sampling is needed for curve segments that exhibit substantial variations in curvature.

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 be lost, while using too many control points may lead to overfitting and bring artifacts after data embedding. 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, the longer the fingerprint sequence, and in turn the more robust the fingerprint against noise and attacks. In our work, the number of control points is about 5~8% of the total number of curve pixels. We will discuss more on the selecting of control points in the context of map fingerprinting in Section 3.2.

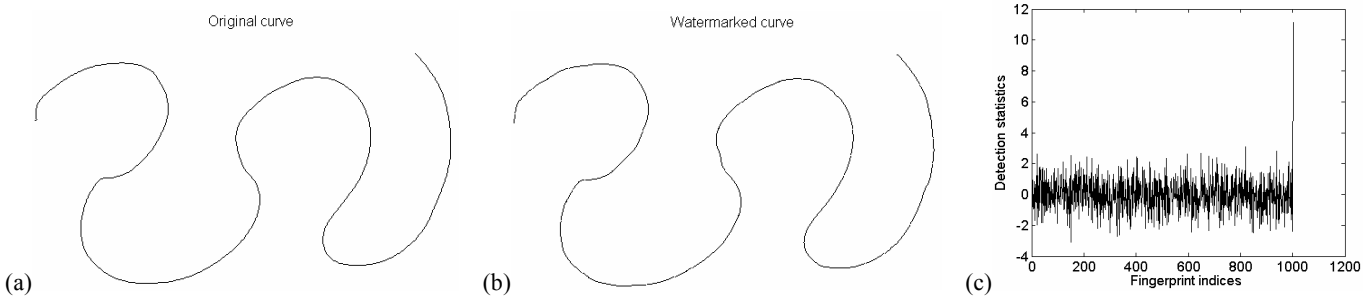


Figure 1: (a) Original curve; (b) Fingerprinted curve; (c) Detection statistics.

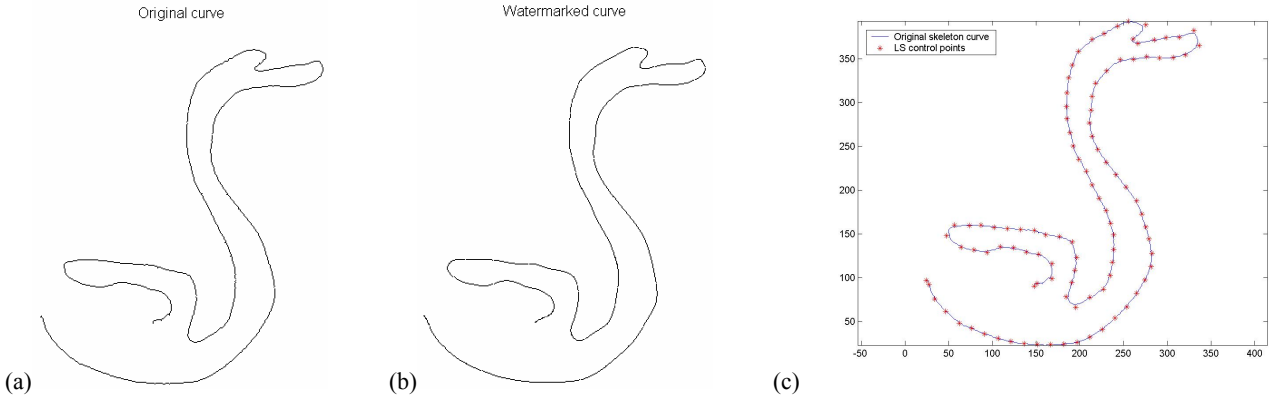


Figure 2: (a) Original curve; (b) Fingerprinted curve; (c) Control points overlaid on the original curve.

The scaling factor α also affects the robustness and fidelity of the fingerprinting. The larger the scaling factor, the more robust the fingerprint, but the larger the distortion resulted in. For cartographic applications, industrial standards provide guidelines on the maximum allowable changes [4]. Perturbation of 2 to 3 pixels is usually considered acceptable. We use random number sequences with unit variance as fingerprints and set α to 0.5 in our tests. The difference between two curves can be quantified using such metric as the *Hausdorff distance* [7] in a max-min sense. More specifically, let $d(a,b)$ be the distance between two points $a$ and $b$, and the distance from curve $A$ to curve $B$ is given by $d_B(A) = \sup_{a \in A} \inf_{b \in B} d(a,b)$.

The Hausdorff distance between the two curves is defined as $d_B(A) + d_A(B)$.

## 3. EXPERIMENTAL RESULTS

### 3.1 Fingerprinting Simple Curves

We first present the fingerprinting results on two simple curves, the "*W*" curve in Figure 1(a) and the "Swan" curve in Figure 2(a), which were hand-drawn on a TabletPC and stored as binary images of size 288x521 and 392x329, respectively. We use the contour-following algorithm in [6] to traverse the curve and obtain the vector representation $\mathbf{p}(t_i)$ indexed by $t_i$. Uniform sampling of curve points and the quadratic B-spline blending function (order $k$ = 3) are employed for fingerprinting these two curves. The fingerprinting results are shown in Figure 1(b) and Figure 2(b), where we have marked 100 control points in each curve. The Hausdorff distance between the original and marked is 3.4 for the "*W*" curve and 5.0 for the "Swan" curve, and the differences are hardly visible to human eyes. The detection results on the fingerprinted "*W*" curve are shown in Figure 1(c), which illustrates the correct positive detection with the 1000th sequence, along with the very small Z statistics for the correct negative detection with other sequences. For the "Swan" curve, we highlight the control points in Figure 2(c).

We then print out the fingerprinted "*W*" curve using a HP laser printer and scan back as a 324x324 binary image shown in Figure 3(a). For the proof-of-concept purpose, we apply manual registration between the scanned curve and the original unmarked curve. Other preprocessing before detection includes a thinning operation 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 Figure 3(b), despite that the curve is simple and the number of control points is relatively small, the fingerprint survives the printing-and-scanning process and gives a detection statistic higher than the detection threshold.

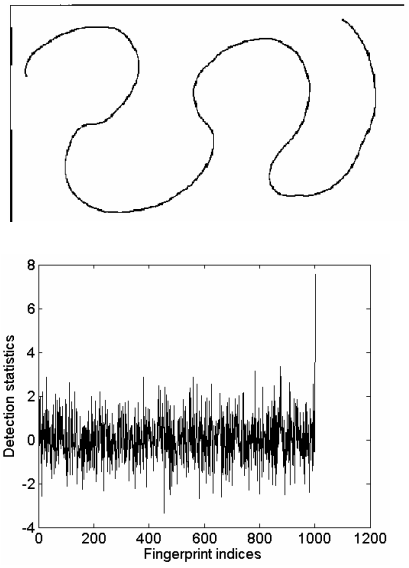


Figure 3: (a) Fingerprinted curve after printing and scanning; (b) Detection statistics.

### 3.2 Fingerprinting Topographic Maps

Topographic maps are two-dimensional representations of the earth's three-dimensional surface. Vertical elevation is shown with contour lines (also known as level lines), which represent the earth's surface that are of equal altitude. Contour lines in topographic maps often exhibit a considerable amount of variations and irregularity, prompting the need of non-uniform sampling of curve points in the parametric modeling of the contours. For each contour, we measure the angle changes in the tangent line at different locations and assign higher weights to locations with larger changes. More specifically, this can be quantified by the singularity [8] of a curve point $\mathbf{p}(t_k) = (x(t_k), y(t_k))$ defined as:

$$sg(t_k) = \left| \arctan\left( \frac{y(t_{k+1}) - y(t_k)}{x(t_{k+1}) - x(t_k)} \right) - \arctan\left( \frac{y(t_k) - y(t_{k-1})}{x(t_k) - x(t_{k-1})} \right) \right| \quad (4)$$

Figure 4 shows the fingerprinting results for a 1100x1100 topographic vector map (Figure 4(a)) obtained from http://www.ablesw.com. Only curves with sufficient size are chosen for marking. In this particular example, we marked 9 curves that have more than 128 vector points each and a total of 1331 control points are used to carry the fingerprint. We overlay in Figure 4(b) these 9 original and marked curves using blue lines and red dots, respectively. To help illustrate the fidelity of our method, we enlarge a portion of the overlaid image in Figure 4(c).

To demonstrate the resistance of the proposed method against collusion, we present in Figure 5 the detection statistics under three different types of collusion attacks. Figure 5(a) shows the collusion results where the control

points for each curve are taken from two differently fingerprinted maps in an alternating fashion. The collusion attack for Figure 5(b) is known as random interleaving, where control points are equiprobably picked from two maps. Averaging collusion among five differently fingerprinted maps is employed for Figure 5(c), by averaging the corresponding control points from the five maps. As we can see, the embedded fingerprints from all contributing sequences/users survive the collusion attacks and are identified with high confidence.

## 4. CONCLUSIONS AND FURTURE WORK

In this paper, we have presented a new data hiding algorithm for curves by parameterizing a curve using B-spline model and adding spread spectrum sequences in curve parameters. We have demonstrated the feasibility of our proposed algorithm in collusion-resistant fingerprinting applications for writings/drawings from pen-based input as well as topographic maps.

For future work, the manual registration used now can be replaced by automated registration [9]. This will facilitate the fingerprint detection from scanned maps as well as dealing with geometric attacks. Printing-and-scanning tests for large scale maps are also worth further investigations.

**Acknowledgement** This work was supported in part by NSF Grant CCR0133704. The authors thank Prof. B. Liu and Dr. M. Xia of Princeton University for a preprint of Ref.[9] and the enlightening discussions that inspired this research.

## REFERENCES

[1] M. Wu, W. Trappe, Z. Wang, and K.J.R. Liu: "Collusion Resistant Fingerprinting for Multimedia", IEEE Signal Processing Magazine, vol.21(2), pp. 15-27, March 2004.

[2] M. Wu, E. Tang, and B. Liu: "Data Hiding in Digital Binary Image", Proc. of IEEE ICME, pp.393-396, 2000.

[3] V. Solachidis, N. Nikolaidis, and I. Pitas: "Watermarking Polygonal Lines Using Fourier Descriptors", IEEE 2000 ICASSP, pp. 1955-1958, 2000.

[4] R. Ohbuchi, H. Ueda, S. Endoh: "Watermarking 2D Vector Maps in the Mesh-Spectral Domain," Proc. of the Shape Modeling International (SMI), 2003.

[5] I. Cox, J. Kilian, F. Leighton, and T. Shamoon: "Secure spread spectrum watermarking for multimedia," IEEE Tran. on Image Proc., vol. 6(12), pp.1673-1687, 1997.

[6] A. K. Jain: *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[7] E. Belogay, C. Cabrelliay, U. Molter, R. Shonkwiler: "Calculating the Hausdorff distance between curves," Information Processing Letters 64, pp.17-22, 1997.

[8] C.A. Cabrelli and U.M. Molter: "Automatic Representation of Binary Images", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.12(12), pp.1190-1196, 1990.

[9] M. Xia and B. Liu: "Image Registration by 'super-curves' ", IEEE Tran. on Image Proc., pp. 720-732, vol.13(5), 2004.

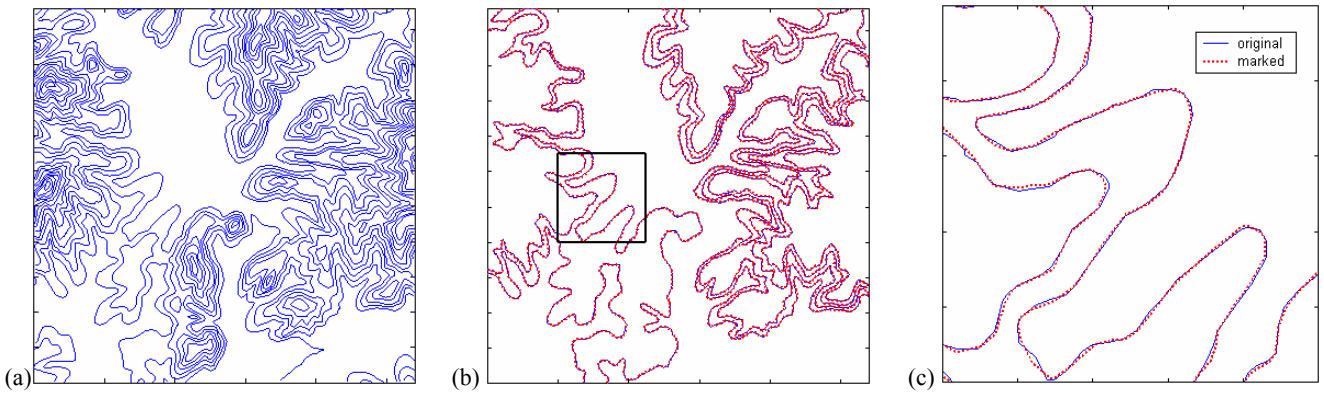


Figure 4: (a) Original map; (b) Original and fingerprinted curves overlaid with each other; (c) Enlarged difference.

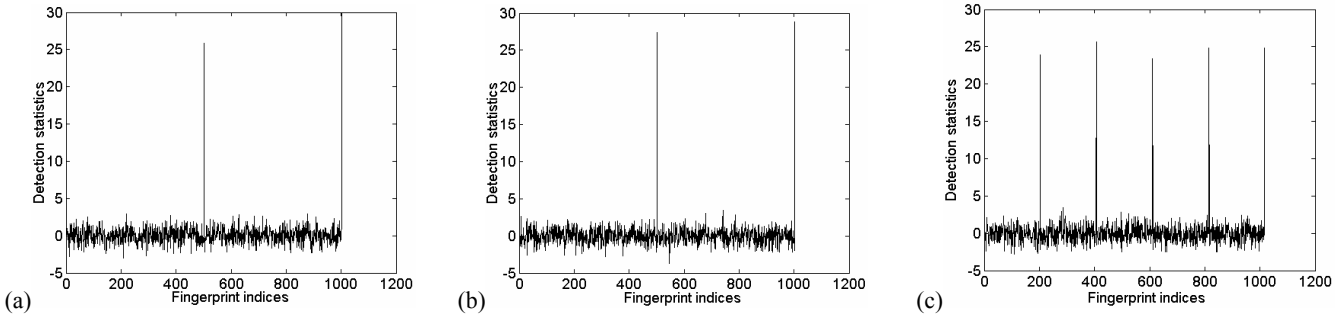


Figure 5: Detection results after collusion attacks on fingerprinted topographic maps of Figure 4: (a) 2-user alternating interleaving collusion; (b) 2-user random interleaving collusion; (c) 5-user averaging collusion.