Fingerprinting Digital Elevation Maps *

Hongmei Gou and Min Wu †

ECE Department, University of Maryland, College Park, U.S.A

ABSTRACT

Digital elevation maps (DEMs) provide a digital representation of 3-D terrain information. In civilian applications, high-precision DEMs carry a high commercial value owing to the large amount of effort in acquiring them; and in military applications, DEMs are often used to represent critical geospatial information in sensitive operations. These call for new technologies to prevent unauthorized distribution and to trace traitors in the event of information leak related to DEMs. In this paper, we propose a new digital fingerprinting technique to protect DEM data from illegal re-distribution. The proposed method enables reliable detection of fingerprints from both 3-D DEM data set and its 2-D rendering, whichever format that is available to a detector. Our method starts with extracting from a DEM a set of critical contours either corresponding to important topographic features of the terrain or having application-dependent importance. Fingerprints are then embedded into these critical contours by employing parametric curve modeling and spread spectrum embedding. Finally, a fingerprinted DEM is constructed to incorporate the marked 2-D contours. Through experimental results, we demonstrate the robustness of the proposed method against a number of challenging attacks applied to either DEMs or their contour representations.

Keywords: Data embedding, digital fingerprinting, geospatial data protection, digital elevation maps, B-splines

1. INTRODUCTION

Digital elevation maps (DEMs) provides a digital representation of surface terrains, by defining a set of points (x, y, z) in the three-dimensional space. The coordinates (x, y) indicate the spatial location of the point, while the z coordinate represents its elevation. An example DEM for the Monterey Bay region is shown in Figure 1(a), where the ocean-floor depth data are obtained from the U.S. National Geophysical Data Center (NGDC) [1]. In recent years, acquisition systems used to obtain DEM data have been improved to achieve better resolutions in both the spatial plane and the elevation coordinate. With such improvement, DEMs have become more widely used in military and commercial operations, such as navigation, landing, petroleum exploration, and land use planning. Because of the large amount of efforts put in acquiring DEM data as well as their critical role in practical applications, DEMs have high commercial values and should be protected from unauthorized copying and re-distribution. Furthermore, DEM data used in military applications contain sensitive information, and prompt the need for preventing information leakage as well as tracing the source of leak.

Data embedding techniques are promising building blocks to construct digital forensic mechanisms that protect geospatial data from illegal distribution. Before distributing copies of a DEM, the authority embeds into each DEM copy a unique ID, which is referred to as a *digital fingerprint* and represents a recipient's identity [2]. When some recipients leak their copies and these leaked copies are acquired by the authority, the sources of the leak can be identified by examining what IDs are contained in the suspicious DEMs. In order to trace individual copies successfully, the embedded fingerprints must be difficult to remove under a variety of attacks, which calls for a robust embedding method. In the mean time, the fingerprinted DEM should preserve the geographical information to be conveyed with high precision. Otherwise, there may be serious consequences, for example, when inaccurate DEM data are given to troops or fed into navigation systems.

The data points of DEMs are generally acquired and archived using a rectangular grid. If we take the spatial location (x, y) as the image coordinates, and the elevation value z as the image intensity, a 3-D DEM can be

* This work was supported in part by the U.S. Air Force Research Laboratory under Grant FA8750-05-1-0238.

† The authors can be reached at hmgou@eng.umd.edu and minwu@eng.umd.edu.

Figure 1. Digital representation of Monterey Bay, California: (a) the 3-D DEM; (b) the corresponding representation in 2-D gray-scaled image; (c) a 2-D topographic map of the 3-D DEM.

represented by a 2-D gray-scaled image, as shown in Figure 1(b). A DEM generally contains much redundant information in that neighboring points in many areas have gradually changing elevations. Therefore, applications of the DEM often focus on elevation contours extracted from the DEM, as opposed to the DEM itself [3]. A contour line represents the earth’s surfaces that have an equal altitude. From a given DEM, a set of contours corresponding to certain elevations can be extracted, forming a 2-D topographic map as shown in Figure 1(c). Topographic maps can convey typical geographic features of a surface terrain. Therefore, they provide partial but important information of DEMs.

Since DEMs can be represented as 2-D gray-scaled images, approaches developed for robustly watermarking images may be employed to fingerprint DEMs. One candidate is spread spectrum embedding in the DCT (Discrete Cosine Transform) or DFT (Discrete Fourier Transform) domain [4]. However, such image watermarking methods do not explicitly take the geographic meanings of DEMs into consideration. As contours represent typical geographic features of a DEM, an aggressive attacker may preserve some meaningful contours but remove all the other information. Under this type of attacks that reduces 3-D digital elevation maps to 2-D contours, the fingerprint information embedded in the DCT or DFT coefficients would be destroyed along with the dimension reduction.

In order to combat such attack of reducing 3-D DEMs to 2-D contours as well as other attacks and distortions to either DEMs or their contours, we propose to fingerprint DEMs through hiding data in their 2-D contours. This approach has the potential to make the fingerprint detectable from both the DEMs and the contours, whichever available to a fingerprint detector. We have recently developed a robust data hiding method for 2-D curves by identifying and manipulating curve parameters obtained from B-spline modeling of curves [5–7]. Taking mutually independent, noise-like sequences as digital fingerprints, we perform spread spectrum embedding in the coordinates of B-spline control points. By adjusting the strength of the fingerprint sequence, the marked curve can maintain fidelity with respect to the original curve. For the fingerprint detection, we propose an iterative alignment-minimization algorithm coupled with correlation-based detection to achieve robustness against various distortions and attacks, such as collusion, geometric transformations, and D/A-A/D conversions. In this paper, we extend our curve embedding method to fingerprinting DEMs. We carry out proper transformations between a 3-D DEM and its 2-D contours, and hide fingerprints in the 2-D contours of the DEM data.

The paper is organized as follows. Section 2 provides a framework of fingerprinting a DEM through hiding data in its contours, as well as a detailed description of the embedding and detection algorithms. In Section 3, we discuss the tradeoff between fingerprint robustness and imperceptibility, and address a special attack named contour replacement. Experimental results are presented in Section 4 to demonstrate the effectiveness of the proposed approach. Finally, conclusions are drawn in Section 5.

2. EMBEDDING AND DETECTION ALGORITHMS

When fingerprinting a DEM through hiding data in its 2-D contours, the first step is to determine which contours to select for hiding fingerprints. If we select some contours to embed the fingerprint but adversaries are likely to

be interested in other elevations, they may carefully re-render the DEM data into a different set of 2-D contours to avoid the marked elevations. Such an attack can remove a considerable amount of fingerprint information. The concept of “critical contours” [8] can help mitigate such attacks. Critical contours of a DEM are those equal-elevation lines that correspond to either interesting topographic features of a terrain, such as the maxima, the minima, and the saddle points, or some objects with application-dependent importance, such as oil wells. Analogous to perceptually significant components in natural images, these critical contours are essential to a DEM and can be used to carry the fingerprint. As attackers are often not willing to completely remove or seriously distort these critical contours due to their critical role in conveying valuable geospatial information, the fingerprint embedded into them will be preserved to some degree as well.

After identifying the contours to carry the hidden data, we fingerprint a DEM through proper transformations between the 3-D DEM and its 2-D contours, and leveraging a curve fingerprinting method developed in our recent work. The main steps of the proposed approach are illustrated in Figure 2. In the following Sections 2.1-2.2, we explain the embedding and detection algorithms in detail.

Figure 2. The block diagram of fingerprinting 3-D DEMs through hiding data in 2-D contours.

2.1. Data Embedding in Digital Elevation Maps

In the embedding process, we first extract from the given DEM a set of critical 2-D contours. Then, we embed the fingerprint sequence into these contours by employing B-spline based parametric curve modeling and spread spectrum embedding. Finally, we construct a fingerprinted DEM to incorporate the marked 2-D contours.

2.1.1. Extracting Critical 2-D Contours from a DEM

Critical contours of a DEM either identify certain topographic features of the terrain or depend on its usage in particular applications. Correspondingly, elevations of critical contours, or *critical elevations* in short, can be obtained from analyzing terrain features, or have been specified by the applications. In the literature, the Morse theory has been used to identify critical elevations corresponding to such topographical features as maxima, minima, and saddle points [8].

Formulate a DEM as a function $u : D \rightarrow \mathbb{R}$, where $D \subset \mathbb{R}^2$ represents (x, y) locations, and the function u specifies the elevation value of each location. The upper level set of an elevation λ , denoted as $[u \geq \lambda]$, consists of locations where the height is greater than or equal to λ , i.e., $[u \geq \lambda] \triangleq \{(x, y) \in D : u(x, y) \geq \lambda\}$. Similarly, the lower level set of an elevation λ is defined as $[u \leq \lambda] \triangleq \{(x, y) \in D : u(x, y) \leq \lambda\}$. More generally, for $\lambda, \mu \in \mathbb{R}$, and $\lambda \leq \mu$, we can define a set $[\lambda \leq u \leq \mu] \triangleq \{(x, y) \in D : \lambda \leq u(x, y) \leq \mu\}$. Further, the contour set of an elevation λ is defined as $[u = \lambda] = [\lambda \leq u \leq \lambda]$.

Shown in Figure 3 are two major types of terrain, where maxima appear in the Type-I terrain, and minima appear in the Type-II terrain. There are also two kinds of saddle points, one with each type of terrain. To examine the properties of these two types of terrain, we consider a DEM with elevation values in the range of

$[a, b], a \leq b$, i.e., $u : D \rightarrow [a, b]$. We examine the contour set $[u = \lambda]$ as λ continually decreases from b to a . For the Type-I terrain shown in Figure 3(a), each time when we come across a maximum, a small curve appears in the contour set, as the situation for elevation λ_1 . As we move toward lower elevation, if there is a saddle point, as for elevation λ_2 , two relatively short curves merge into a single curve. For the Type-II terrain shown in Figure 3(b), a small curve disappears at each minimum, e.g. at elevation λ_3 , while a single curve splits into two short ones at a saddle point, e.g., at elevation λ_4 . Through such examination of the moving contour sets, critical elevations corresponding to maxima, minima, and saddle points can be identified.

Figure 3. The two types of terrain and their critical elevations: (a) Type-I with maxima and saddle points; (b) Type-II with minima and saddle points.

As there may be many small oscillations in a terrain, the direct application of the above method may identify many critical elevations, some of which may not represent significant topographic features of the terrain. In order to find the major critical elevations, Extrema filters are used in [8] to remove the small oscillations while preserving the major topographic changes. Applying Extrema filtering to a DEM, elevation values around the maxima/minima are decreased/increased by a certain amount so that the very short curves corresponding to the small oscillations can be eliminated. From such a filtered DEM, the major critical elevations can then be identified using the Morse theory described above.

After obtaining the critical elevations, either directly specified by particular applications, or found using the Morse theory, we can easily find their corresponding critical contours. For each critical elevation λ , we first identify its upper level set $[u \geq \lambda] \triangleq \{(x, y) \in D : u(x, y) \geq \lambda\}$. Assigning 1's to the locations in the upper level set and 0's to the other locations to form a binary image, we employ the canny edge detection method to extract the boundary of the upper level set $[u \geq \lambda]$ as the critical contour at elevation λ . Since the contour extracted in this way is in a raster format, we finally use a contour following algorithm in [9] to traverse it and represent it using a set of ordered curve points.

2.1.2. Hiding Data in Critical 2-D Contours

After obtaining the critical contours, we parameterize them according to the B-spline model. Using coordinates of B-spline control points as the feature domain, we embed the fingerprint into contours through spread spectrum embedding.

B-spline Modelling of Contours B-splines are piecewise polynomial functions that provide local approximations of curves using a small number of parameters known as the *control points* [10] [11]. 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

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

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

Given a set of samples on the curve, finding a 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. Representing

coordinates of $m + 1$ samples as an $(m + 1) \times 2$ matrix $\mathbf{P} \triangleq (\mathbf{p}_x, \mathbf{p}_y)$ and coordinates of $n + 1$ control points as an $(n + 1) \times 2$ matrix $\mathbf{C} \triangleq (\mathbf{c}_x, \mathbf{c}_y)$, we can write the least-squares problem with its solution as

$$\begin{cases} \min_{\mathbf{c}_x} \|\mathbf{B}\mathbf{c}_x - \mathbf{p}_x\|^2 \\ \min_{\mathbf{c}_y} \|\mathbf{B}\mathbf{c}_y - \mathbf{p}_y\|^2 \end{cases} \implies \begin{cases} \mathbf{c}_x = \mathbf{B}^\dagger \mathbf{p}_x \\ \mathbf{c}_y = \mathbf{B}^\dagger \mathbf{p}_y \end{cases} \quad (2)$$

where $\{\mathbf{B}\}_{ji}$ is the value of the k^{th} -order B-spline blending function $B_{i,k}(t)$ evaluated at $t = s_j$ for the i^{th} control point and $\dagger$ denotes the pseudo inverse of a matrix.

Curvature-based Sampling of Curve Points The estimation of B-spline control points of a contour requires a set of appropriately chosen samples to feed into the least-squares estimator. As contours extracted from DEMs often exhibit a considerable amount of variations and irregularities, a curvature-based method can be employed to non-uniformly sample the curve points, with more samples selected for higher-curvature segments. Formally, the curvature [12] of a point $\mathbf{p}(t) = (p_x(t), p_y(t))$ on a curve $\{\mathbf{p}(t)\}$ is defined as

$$k(t) \triangleq \frac{p'_x p''_y - p'_y p''_x}{(p'^2_x + p'^2_y)^{3/2}}, \quad (3)$$

where $p'_x = \frac{dp_x}{dt}$, $p''_x = \frac{d^2 p_x}{dt^2}$, $p'_y = \frac{dp_y}{dt}$, and $p''_y = \frac{d^2 p_y}{dt^2}$. In practical implementations, we approximate the curvature of each point on the contour 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. We then feed the selected sample points to the least-squares estimator to obtain the B-spline control points of the contour.

Spread Spectrum Embedding of Fingerprints In the embedding, we use mutually independent, noise-like sequences as digital fingerprints to represent different users/IDs for trace and track purposes. The practical implementation employs a pseudo-random number generator to produce a sequence of independent random numbers as a fingerprint and uses different seeds for different users [13] [14] [15]. To apply spread spectrum embedding on a contour, we add a scaled version of the fingerprint sequence $(\mathbf{w}_x, \mathbf{w}_y)$ to the coordinates of the B-spline control points obtained before, and get a set of marked control points $(\mathbf{c}'_x, \mathbf{c}'_y) = (\mathbf{c}_x + \alpha \mathbf{w}_x, \mathbf{c}_y + \alpha \mathbf{w}_y)$, where α is a scaling factor adjusting the fingerprint strength. A fingerprinted contour can then be constructed from these watermarked control points by using the B-spline synthesis equation (1).

2.1.3. Constructing Fingerprinted DEM from Marked 2-D Contours

The final step in the embedding is to construct a fingerprinted DEM to incorporate those watermarked critical contours so that the fingerprint embedded in contours is also detectable from this newly constructed DEM. In order to reliably detect the fingerprint embedded in contours, we need to make the watermarked 2-D contours recoverable from this marked DEM with high accuracy. Additionally, in order to preserve the geospatial information conveyed by the DEM, we need to make the distortion between this constructed DEM and the original DEM as small as possible.

In the original DEM $u : D \rightarrow \mathbb{R}$, the critical contour at an elevation λ is the boundary of its upper level set $[u \geq \lambda]$. After curve-based fingerprinting, the original contour is slightly deviated to be a marked contour, which prescribes an upper level set slightly different from $[u \geq \lambda]$. Now we construct a fingerprinted DEM $u' : D \rightarrow \mathbb{R}$ so that its upper level set $[u' \geq \lambda]$ at elevation λ takes the marked contour as the boundary. To accomplish this, we increase the elevation value to be $\lambda + \delta$ for locations that are in $[u' \geq \lambda]$ but not in $[u \geq \lambda]$, decrease it to be $\lambda - \delta$ for those that are in $[u \geq \lambda]$ but not in $[u' \geq \lambda]$, and keep it the same as in the original DEM for all the other locations, i.e.,

$$u'_\lambda(x, y) = \begin{cases} \lambda + \delta & (x, y) \in [u' \geq \lambda] \wedge (x, y) \notin [u \geq \lambda] \\ \lambda - \delta & (x, y) \notin [u' \geq \lambda] \wedge (x, y) \in [u \geq \lambda] \\ u(x, y) & \text{otherwise} \end{cases} \quad (4)$$

where δ is a positive number to specify the amount of elevation increment/decrement. One possible choice of δ is to use the smallest unit of the elevation coordinate, which is determined by the DEM vertical resolution.

As the watermarked contour maintains high fidelity with respect to its original version, the two upper level sets $[u \geq \lambda]$ and $[u' \geq \lambda]$ are different in only a small number of points. Using equation (4), we further make the minimal amount of change in the fingerprinted DEM to incorporate the watermarked contours. Therefore, our fingerprinted DEM can preserve the geospatial information conveyed by the original DEM with high precision.

2.2. Fingerprint Detection from Digital Elevation Maps or Their 2-D Rendering

To detect the fingerprint(s) from a suspicious DEM, we first extract from it the critical contours at the elevations corresponding to those used in the embedding. If the data in question are already rendered in 2-D contours, these rendered contours will be passed directly to the detector. Then, we estimate a fingerprint sequence from these extracted/rendered 2-D contours. Finally, correlation-based detection is performed to identify the sources of information leak.

2.2.1. Curve Alignment and B-spline Control Point Retrieval

When affine transformations (e.g., rotation, scaling, translation, and shearing) are applied to a fingerprinted DEM, its rendered contours undergo the same kind of distortions. To combat these geometric distortions, preceding the basic fingerprint detection module, there must be a pre-processing registration step to align the test contour with its original version. To achieve a good estimation of the fingerprint sequence embedded in B-spline control points, another important task is to accurately retrieve from the test contour the set of watermarked control points $(\mathbf{c}'_x, \mathbf{c}'_y)$ as in Section 2.1.2. This B-spline control point retrieval is a tough job because of the inherent non-uniqueness 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.

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

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (5)$$

where $\{a_{ij}\}$ are parameters representing the collective effect of scaling, rotation, translation, reflection, and shearing. These transformation parameters can be represented by two column vectors $\mathbf{a}_x = [a_{11} \ a_{12} \ a_{13}]^T$ and $\mathbf{a}_y = [a_{21} \ a_{22} \ a_{23}]^T$. Similarly, we represent the inverse transformation by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_x^T \\ \mathbf{a}_y^T \\ 0 \ 0 \ 1 \end{bmatrix}^{-1} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{g}_x^T \\ \mathbf{g}_y^T \\ 0 \ 0 \ 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix}. \quad (6)$$

Given two curves $\{(x, y)\}$ and $\{(\tilde{x}, \tilde{y})\}$ that are affine-related to each other, the task of curve alignment is to estimate the transformation parameters $(\mathbf{a}_x, \mathbf{a}_y)$, or equivalently $(\mathbf{g}_x, \mathbf{g}_y)$.

In our fingerprinting applications, we have a raster contour $(\tilde{\mathbf{r}}_x, \tilde{\mathbf{r}}_y)$ extracted/rendered from the test DEM. We also know its original version $(\mathbf{r}_x, \mathbf{r}_y)$ with the set of sample points $(\mathbf{p}_x, \mathbf{p}_y)$ used to estimate the set of B-spline control points $(\mathbf{c}_x, \mathbf{c}_y)$ for carrying the fingerprint. As it still remains to be determined which fingerprints are present in the test curve, we register the test contour $(\tilde{\mathbf{r}}_x, \tilde{\mathbf{r}}_y)$ with its original version to find the transformation parameters $(\mathbf{a}_x, \mathbf{a}_y)$, or $(\mathbf{g}_x, \mathbf{g}_y)$. In order to address the non-uniqueness of B-spline control points, we leverage the one-to-one relationship between sample points and B-spline control points, i.e., a set of sample points leads to a set of B-spline control points. We then resort the problem of B-spline control point retrieval to a *point correspondence problem*, which aims at finding from the test contour a set of sample points $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y)$ that corresponds to the set $(\mathbf{p}_x, \mathbf{p}_y)$ used in the embedding. In our recent work, we develop an iterative alignment-minimization (IAM) algorithm [6, 7] that can perform curve alignment and solve the point correspondence problem simultaneously. In the following, we explain the key steps of the algorithm.

Step-1: Initial Estimation of Sample Points on Test Curve We initialize the sample points $(\tilde{\mathbf{p}}_x^{(1)}, \tilde{\mathbf{p}}_y^{(1)})$ 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 $\mathbf{J} = [j_0, j_1, j_2, \dots, j_m]$ of the original curve's $m+1$ sample points, where $j_0 < j_1 < j_2 < \dots < j_m$ are integers ranging from 0 to $M-1$, we estimate the indices of the test curve's $m+1$ sample points by $\tilde{\mathbf{J}} = \text{round}\left(\frac{\tilde{M}-1}{M-1} \cdot \mathbf{J}\right)$. 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.

Step-2: Curve Alignment with the Estimated Sample Points Given the estimated point correspondence with sample points $(\tilde{\mathbf{p}}_x^{(i)}, \tilde{\mathbf{p}}_y^{(i)})$ for the test curve in the i^{th} iteration, we apply the curve alignment method in [16] to estimate the transformation parameters and the control points of the test curve. We use “View-I” to refer to the geometric setup of the original curve and “View-II” to refer to the setup of the test curve. Let the transformation parameters from View-I to View-II be $(\mathbf{a}_x^{(i)}, \mathbf{a}_y^{(i)})$, and those from View-II to View-I be $(\mathbf{g}_x^{(i)}, \mathbf{g}_y^{(i)})$. Using the latter, we can transform the sample points $(\tilde{\mathbf{p}}_x^{(i)}, \tilde{\mathbf{p}}_y^{(i)})$ on the test curve to View-I. 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 [16]), and search for both the transformation parameters $(\hat{\mathbf{g}}_x^{(i)}, \hat{\mathbf{g}}_y^{(i)})$ and the B-spline control points $(\hat{\mathbf{c}}_x^{(i)}, \hat{\mathbf{c}}_y^{(i)})$ to minimize the fitting error

$$f(\hat{\mathbf{c}}_x^{(i)}, \hat{\mathbf{c}}_y^{(i)}, \hat{\mathbf{g}}_x^{(i)}, \hat{\mathbf{g}}_y^{(i)}) = \left\| \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix} \hat{\mathbf{c}}_x^{(i)} - \begin{bmatrix} \mathbf{p}_x \\ \tilde{\mathbf{P}}^{(i)} \hat{\mathbf{g}}_x^{(i)} \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix} \hat{\mathbf{c}}_y^{(i)} - \begin{bmatrix} \mathbf{p}_y \\ \tilde{\mathbf{P}}^{(i)} \hat{\mathbf{g}}_y^{(i)} \end{bmatrix} \right\|^2, \quad (7)$$

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 1's. The partial derivatives of (7) 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 can obtain an estimate of the transformation parameters $(\hat{\mathbf{g}}_x^{(i)}, \hat{\mathbf{g}}_y^{(i)})$ and the B-spline control points $(\hat{\mathbf{c}}_x^{(i)}, \hat{\mathbf{c}}_y^{(i)})$.

Step-3: Refined Estimation of Sample Points on the Test Curve Given the estimated transformation parameters $(\hat{\mathbf{g}}_x^{(i)}, \hat{\mathbf{g}}_y^{(i)})$, we align the test raster curve $(\tilde{\mathbf{r}}_x, \tilde{\mathbf{r}}_y)$ with the original curve by transforming it to View-I as $(\tilde{\mathbf{r}}_{x,I}^{(i)}, \tilde{\mathbf{r}}_{y,I}^{(i)})$. Because the fingerprinted sample points are located at the neighborhood of their corresponding unmarked version $(\mathbf{p}_x, \mathbf{p}_y)$, we apply a *nearest-neighbor* rule to refine the estimation of the test curve's sample points. Specifically, for each point of $(\mathbf{p}_x, \mathbf{p}_y)$, 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 back to View-II with parameters $(\hat{\mathbf{a}}_x^{(i)}, \hat{\mathbf{a}}_y^{(i)})$ as a new estimate of the test sample points $(\tilde{\mathbf{p}}_x^{(i+1)}, \tilde{\mathbf{p}}_y^{(i+1)})$. 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.

2.2.2. Correlation-based Fingerprint Detection

Using the IAM algorithm described above, we accomplish curve registration and address the non-uniqueness of B-spline control points. After getting a good estimation of the right set of control points, $(\tilde{\mathbf{c}}_x, \tilde{\mathbf{c}}_y)$, we compute the difference between their coordinates and those of the original control points $(\mathbf{c}_x, \mathbf{c}_y)$ to arrive at an estimated fingerprint sequence $(\tilde{\mathbf{w}}_x, \tilde{\mathbf{w}}_y) = (\frac{\tilde{\mathbf{c}}_x - \mathbf{c}_x}{\alpha}, \frac{\tilde{\mathbf{c}}_y - \mathbf{c}_y}{\alpha})$.

The estimated fingerprint sequence consists of one or several users' contributions as well as some noise coming from distortions and/or attacks. The problem of finding out which user(s) has(have) contributed to this estimated fingerprint sequence can be formulated as hypothesis testing [17], 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. If the similarity is higher than a threshold, with high probability we can identify the user represented by this fingerprint sequence as a source of information leak. In our work, we compute the sample correlation coefficient r between two sequences and then convert it to a Z statistic [13, 18] by $Z \triangleq \frac{\sqrt{L-3}}{2} \log \frac{1+r}{1-r}$, where L is the length of fingerprint sequences. The Z statistic has been shown to follow approximately a unit-variance Gaussian distribution with a large positive mean when the true correlation coefficient ρ between the two sequences is large, and a zero mean when $\rho = 0$ [19]. Thus, different thresholds on the Z statistic will give

us different probabilities of false alarm P_{fa} . A threshold of 3 gives a P_{fa} on the order of 10^{-3} , while a threshold of 6 corresponds to the order of 10^{-9} .

3. ROBUSTNESS AND FIDELITY CONSIDERATIONS

In this section, we discuss the tradeoff between fingerprint robustness and imperceptibility when fingerprinting a DEM through hiding data in its rendered contours. We also propose a new attack of contour replacement and discuss the robustness of our method against this challenging attack.

3.1. Tradeoff between Robustness and Imperceptibility of Fingerprints

Because our fingerprint sequence is virtually embedded in a curve feature, we first discuss the fidelity and robustness issues in the curve domain. As pointed out in [7], we have two important parameters for tuning here: the scaling factor α specifying the strength of fingerprint sequences, and the number of B-spline control points. The scaling factor α affects both invisibility and robustness of fingerprints. The larger the scaling factor is, the more robust the fingerprint is, but it results in a larger distortion. This fingerprint strength can be tuned according to the requirements of particular applications. As for the number of B-spline control points, depending on the shape of a 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. In our work, we use a curvature-based rule to determine the number of control points, with more control points allocated to a curve with a larger amount of variations and irregularities. The number of control points also affects 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 incur visible distortions as we discussed before.

We now examine the robustness and fidelity issues in the 3-D DEM domain. When fingerprinting a DEM, we hide the fingerprint sequence in its critical contours. When more critical contours are used, we get a longer fingerprint sequence that can provide stronger resilience against distortions. However, large distortions will be introduced to the fingerprinted DEM when more contours are manipulated to carry a longer fingerprint sequence. In the last step of DEM embedding, we modify the original DEM $u : D \rightarrow \mathbb{R}$ to be a fingerprinted DEM $u' : D \rightarrow \mathbb{R}$ according to equation (4) in Section 2.1.3. Such a modification enables us to correctly extract the marked contour from the fingerprinted DEM. However, it also results in some distortions due to the elevation increment/decrement in the non-overlapping regions ($C \subset \mathbb{R}^2$) of the two upper level sets $[u \geq \lambda]$ and $[u' \geq \lambda]$. A non-trivial elevation difference may be observed when a point $(x, y) \in [u' \geq \lambda] \wedge (x, y) \notin [u \geq \lambda]$ has an elevation value much smaller than λ , because it will be increased to be $\lambda + \delta$ in the fingerprinted DEM u' . Similarly, for a point $(x, y) \in [u \geq \lambda] \wedge (x, y) \notin [u' \geq \lambda]$ with an elevation value much larger than λ , its elevation will be decreased to be $\lambda - \delta$ in the fingerprinted DEM u' , which also results in a non-trivial distortion. To mitigate such non-trivial distortions, thresholding can be used to determine which points in the non-overlapping regions C should change their elevation values. Setting a threshold τ , a point $(x, y) \in C$ gets its elevation changed from the original $u(x, y)$ to $u'(x, y)$ only when $|u(x, y) - u'(x, y)| \leq \tau$ is satisfied. Such thresholding can effectively reduce the distortion introduced to the fingerprinted DEM. However, it affects the accuracy of extracting the marked contour from the fingerprinted DEM, which in turn has a negative effect on the robustness of fingerprints that are virtually embedded in contours. Here we still have to make a tradeoff between imperceptibility and robustness of DEM fingerprinting, and the threshold τ can be tuned according to application requirements.

3.2. Robustness against Contour Replacement Attack

Through hiding data in critical 2-D contours of a DEM, our method is effective in combating the attack of rendering a fingerprinted DEM to a topographic map, which contains some or all of the watermarked 2-D contours because of their critical role in conveying geospatial information. As neighboring points in a DEM often have gradually changing elevation values, a fingerprinted contour can be approximated by or inferred from its neighboring contours. Taking advantage of this property, an aggressive attacker may move one step further to remove the fingerprinted contour itself but preserve some of its neighboring contours during the 2-D rendering. We refer to this attack as a contour replacement attack, and discuss how to make our method resilient to this challenging attack in the following.

Use the contour at elevation λ to carry the fingerprint. In the last step of embedding, we modify the original DEM $u : D \rightarrow \mathbb{R}$ to be a fingerprinted DEM $u' : D \rightarrow \mathbb{R}$ according to equation (4) so that the watermarked contour at elevation λ can be correctly extracted from the latter as the boundary of its upper level set $[u' \geq \lambda]$. Now, we consider the following contour replacement attack. In the 2-D rendering of the fingerprinted DEM, the attacker removes the watermarked contour at elevation λ , but preserves one of its neighboring contours at elevation $\beta \in [\lambda_{min}, \lambda_{max}]$ ($\lambda_{min} \leq \lambda \leq \lambda_{max}$). To combat such a contour replacement attack, we modify the construction of the fingerprinted DEM so that in the newly constructed DEM $u'' : D \rightarrow \mathbb{R}$, the watermarked contour at elevation λ can still be preserved as the boundary of its upper level set $[u'' \geq \beta]$ for any value of $\beta \in [\lambda_{min}, \lambda_{max}]$. In order to achieve this preservation, for a point (x, y) in the upper level set prescribed by the watermarked contour at elevation λ , i.e., $(x, y) \in [u' \geq \lambda]$, we increase its elevation value to $\lambda_{max} + \delta$ if its original elevation $u(x, y) < \lambda_{max}$. On the contrary, for a point $(x, y) \notin [u' \geq \lambda]$, we decrease its elevation to $\lambda_{min} - \delta$ if its original elevation $u(x, y) \geq \lambda_{min}$. All the other points keep their elevation values the same as their original correspondence. That is, in order to combat the contour replacement attack within the range of $[\lambda_{min}, \lambda_{max}]$, we construct the fingerprinted DEM u'' as follows:

$$u''_{\lambda}(x, y) = \begin{cases} \lambda_{max} + \delta & (x, y) \in [u' \geq \lambda] \wedge (x, y) \notin [u \geq \lambda_{max}] \\ \lambda_{min} - \delta & (x, y) \notin [u' \geq \lambda] \wedge (x, y) \in [u \geq \lambda_{min}] \\ u(x, y) & \text{otherwise} \end{cases}, \quad (8)$$

where δ is a positive number that can be as small as the resolution in the elevation coordinate.

Extracted from this newly constructed DEM u'' , the contour at any elevation $\beta \in [\lambda_{min}, \lambda_{max}]$ gives us an accurate recovery of the watermarked contour. This enables us to successfully combat the contour replacement attack within the elevation range of $[\lambda_{min}, \lambda_{max}]$. With this robustness achievement, however, the elevation modification using equation(8) introduces a larger distortion to the fingerprinted DEM u'' . Thus, here we still have to make a tradeoff between imperceptibility and robustness of fingerprints, and the values of λ_{min} and λ_{max} can also be tuned according to specific application requirements. To improve the fingerprint imperceptibility, the thresholding technique discussed before can be applied here also. A point (x, y) will get its elevation changed from $u(x, y)$ to $u''(x, y)$ according to equation (8) only when $|u(x, y) - u''(x, y)| \leq \tau$.

4. EXPERIMENTAL RESULTS

In this section, we apply the proposed 3-D DEM fingerprinting method to the Monterey Bay DEM data set from NGDC [1] and shown in Figure 1. We present experimental results to demonstrate the effectiveness of our method, showing fidelity of fingerprinted DEMs as well as robustness against various challenging attacks.

For the Monterey Bay DEM we are using, its resolution in the elevation coordinate is 1, and 341×411 points are uniformly sampled in an area of latitude $36^{\circ}30'N \sim 37^{\circ}4'N$ and longitude $122^{\circ}26'W \sim 121^{\circ}45'W$. In our experiments, we focus on the ocean part of the Monterey Bay DEM. The deepest point has an elevation of -2918. Using the Morse theory and the concept of upper level sets as discussed in [8] and presented in Section 2.1.1, we extract three critical contours corresponding to saddle points at elevation -802, -494, and -220, respectively. They, we hide a fingerprint sequence with a scaling factor $\alpha = 1$ in these contours, and further construct a fingerprinted Monterey Bay DEM.

First, we examine the tradeoff between imperceptibility and robustness of fingerprints by using different thresholds in the process of constructing a fingerprinted DEM, as discussed in Section 3.1. The thresholding rule of $|u(x, y) - u'(x, y)| \leq \tau$ naturally limits the maximal point-wise elevation distortion in the fingerprinted DEM to be at most the threshold τ . In our tests, we also evaluate a Root Mean Square Error(RMSE) of the fingerprinted DEM u' with the original DEM u via

$$RMSE(u, u') = \sqrt{\frac{\sum_{x=1}^M \sum_{y=1}^N |u'(x, y) - u(x, y)|^2}{M \times N}}, \quad (9)$$

where $M \times N$ is the size of the DEM. As the square root of the mean square error, the RMSE indicates the average elevation difference between two DEMs. As shown in Figure 4(a), with a smaller threshold, the fingerprinted

DEM has a smaller RMSE besides a smaller maximal elevation distortion specified by the threshold itself. This is because a smaller threshold can preserve more elevation values in the original DEM unchanged during the construction of the fingerprinted DEM. For the Monterey Bay DEM data set used in our experiment, we test several thresholds in $[20, 100]$, and find that all of them can achieve a RSME smaller than 1.2. With a threshold of 20, the RMSE can be as small as 0.20. This demonstrates that our method can obtain fingerprinted DEMs with high fidelity. While maintaining the fingerprint invisibility using the thresholding technique, we further examine if the embedded fingerprint can be extracted with high confidence. As shown in Figure 4(b), under the thresholds selected above, all the detection statistics with the correct fingerprint are greater than the detection threshold of 6 that gives us a probability of false alarm on the order of 10^{-9} . This demonstrates that from a fingerprinted DEM with high fidelity, we can reliably extract the fingerprint embedded into its 2-D rendering.

Figure 4. Tradeoff between imperceptibility and robustness of fingerprints with different thresholds τ used in constructing the fingerprinted DEM: (a) RMSE vs. thresholds, (b) detection statistics vs. thresholds.

Then, we evaluate the embedding imperceptibility and detection reliability when using the method proposed in Section 3.2 to combat the contour replacement attack. In our tests, for a given elevation λ , we use a parameter called elevation tolerance $\Delta\lambda$ to specify the upper and lower bound in elevation (i.e., λ_{max} and λ_{min}) of its neighboring contours that may be used to replace the fingerprinted contour at elevation λ :

$$\begin{cases} \lambda_{max} = \lambda + \Delta\lambda \\ \lambda_{min} = \lambda - \Delta\lambda \end{cases} \quad (10)$$

After getting a marked contour at elevation λ , we use equation(8) to construct a fingerprinted DEM u'' to overcome the contour replacement attack in a range of $[\lambda_{min}, \lambda_{max}]$. To maintain fidelity of the fingerprinted DEM, we further enforce the thresholding procedure with a threshold of 60. Varying the elevation tolerance $\Delta\lambda$ from 30 to 0 with an decrement of 10, we first evaluate the detection performance. Under each value of $\Delta\lambda$, for a watermarked contour at elevation λ , we extract several of its neighboring contours at elevations $\lambda - 30$ to $\lambda + 30$ with an increment of 5. Then, we estimate a fingerprint sequence from each of these contours, and calculate its Z statistic with the true fingerprint. From Figure 5(a)-(d), we can see that the proposed method is effective in combating the challenging contour replacement attack within the designed tolerance range, by giving us Z statistics higher than the detection threshold of 6 with the true fingerprint. We can also see that a larger elevation tolerance $\Delta\lambda$ enables us to combat a stronger contour replacement attack. Considering that in the contour replacement attack, a watermarked contour may be more likely to be replaced by the attacker with a neighboring contour closer to it, we employ a weight factor to gradually adjust the extent of propagation changes made to nearby elevations in equation (8). A contour that is farther away from the watermarked critical contour gives fewer points that can be modified when constructing the fingerprinted DEM to combat the contour replacement attack. Under a maximal elevation tolerance $\Delta\lambda = 30$ and using this weighting method, we perform the same detection procedure as before, and obtain detection statistics with the true fingerprint as shown in Figure 5(e). We can see that higher detection statistics can be obtained from closer neighboring contours. We also evaluate fidelity of the fingerprinted DEM when the elevation tolerance $\Delta\lambda$ varies. Using the thresholding technique, we fix the maximal point-wise distortion. As shown in Figure 6, the RMSE of the fingerprinted DEM with the original DEM increases with the elevation tolerance $\Delta\lambda$, but all the RMSEs here are still quite small

with values less than 1.1. When using the weighting method with a maximal elevation tolerance of 30, we get a fingerprinted DEM with $\text{RMSE} = 0.99$, which is smaller than the RMSE of 1.08 for $\Delta\lambda = 30$ without using the weight factor. This reduced embedding distortion is attributed to the fact that a smaller number of locations surrounding the critical contours have their elevations modified during the embedding process.

Figure 5. Detection statistics from neighboring contours with different values of elevation tolerance $\Delta\lambda$.

Figure 6. Fidelity of the fingerprinted DEM with different values of elevation tolerance $\Delta\lambda$.

Finally, we examine fingerprint robustness against common signal processing as well as geometric distortions. In the embedding, we use both thresholding (with a threshold of 60) and elevation tolerance ($\Delta\lambda = 10$). These settings give us a fingerprinted DEM with $\text{RMSE} = 0.77$. We then act as attackers and apply different distortions to the fingerprinted DEM. From this distorted fingerprinted DEM, we extract the three contours corresponding to those used in the embedding, and then perform correlation-based detection. The Z detection statistic with the true fingerprint sequence after additive Gaussian noise with a standard deviation $\sigma = 12$ is 8.86, and the one after 5×5 Gaussian low-pass filtering with a standard deviation $\sigma = 0.6$ is 6.79. We also apply some affine transformations to the fingerprinted DEM. The Z statistics with the true fingerprint sequence after $1.2\times$ scaling, $0.8\times$ scaling, and 10° rotation are 9.40, 8.10, and 9.40, respectively. All of the above Z values are higher than the detection threshold of 6 (corresponding to a probability of false alarm on the order of 10^{-9}) and suggest

positive identification of the correct fingerprint. When more critical contours or larger DEM data sets are used to carry fingerprints, we can have a larger number of markable features. This in turn enables us to obtain higher detection statistics that lead to higher robustness against attacks, and/or to assign orthogonal fingerprint sequences to more users. When attacks and distortions are applied to the 2-D rendering of the DEM consisting of the fingerprinted contours, our previous work has demonstrated that the embedded fingerprint can survive many challenging attacks such as cropping, geometric distortions, curve smoothing, and printing-and-scanning [7].

5. CONCLUSIONS

In this paper, we have proposed a new digital fingerprinting technique to protect DEM data from illegal redistribution. The new method enables reliable detection of fingerprints from both 3-D DEM data set and its 2-D rendering, whichever format that is available to a detector. In the embedding, we extract critical contours of a DEM to carry the fingerprint, and then construct a fingerprinted DEM to incorporate the marked 2-D contours. Both embedding and detection of the fingerprint are virtually performed in the contour domain, utilizing B-spline based parametric curve modeling, spread spectrum embedding, and correlation-based detection preceded by high-precision curve registration. We have examined the tradeoff between the fidelity of the marked DEM data and the robustness of fingerprints, and have introduced several fine-tuning techniques to adjust the tradeoff. We have demonstrated through experiments the fidelity of the proposed method as well as its robustness against some challenging attacks, such as contour replacement and geometric distortions.

REFERENCES

1. National Geophysical Data Center (NGDC), <http://www.ngdc.noaa.gov>.
2. M. Wu, W. Trappe, Z.J. Wang, and K.J.R. Liu, "Collusion resistant fingerprinting for multimedia," *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 15–27, March 2004.
3. I. Kweon, and T. Kanade, "Extracting topographic terrain features from elevation maps," *Journal of Computer Vision, Graphics and Image Processing: Image Understanding*, Vol. 59, No. 2, pp. 171–182, March, 1994.
4. I. Cox, J. Killian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.
5. H. Gou and M. Wu, "Data hiding in curves for collusion-resistant digital fingerprinting," *ICIP*, pp. 51–54, 2004.
6. H. Gou and M. Wu, "Robust digital fingerprinting for curves," *ICASSP*, Volume II, pp. 529–532, 2005.
7. H. Gou and M. Wu, "Data hiding in curves with application to fingerprinting maps," *IEEE Trans. on Signal Processing*, vol. 53, no. 10, part 2, pp. 3988–4005, Oct. 2005.
8. A. Sole, V. Caselles, G. Sapiro, and F. Arandiga, "Morse description and geometric encoding of Digital Elevation Maps," *IEEE Trans. on Image Processing*, vol. 13, no. 9, pp. 1245–1262, Sept. 2004.
9. C. Cabrelli and U. Molter, "Automatic representation of binary images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 12, pp. 1190–1196, 1990.
10. A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
11. G. E. Farin, *Curves and Surfaces for Computer-aided Geometric Design: a Practical Guide*, 4th ed. Academic Press, 1997.
12. H. S. M. Coxeter, *Introduction to Geometry*, 2nd ed. New York: Wiley, 1969.
13. H. Zhao, M. Wu, Z.J. Wang, and K.J.R. Liu, "Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting," *IEEE Trans. on Image Processing*, vol. 14, no. 5, pp. 646–661, May 2005.
14. W. Trappe, M. Wu, Z.J. Wang, and K.J.R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. on Signal Processing*, vol. 51, no. 4, pp. 1069–1087, April 2003.
15. D. Kirovski, H.S. Malvar, and Y. Yacobi, "Multimedia Content Screening using a Dual Watermarking and Fingerprinting System," *Proc. of ACM Multimedia*, 2002.
16. M. Xia and B. Liu, "Image registration by 'super-curves'," *IEEE Trans. on Image Processing*, vol. 13, no. 5, pp. 720–732, May 2004.
17. Z.J. Wang, M. Wu, H. Zhao, W. Trappe, and K.J.R. Liu, "Collusion resistance of multimedia fingerprinting using orthogonal modulation," *IEEE Trans. on Image Processing*, vol. 14, no. 6, pp. 804–821, June 2005.
18. H. Stone, "Analysis of attacks on image watermarks with randomized coefficients," *Technical Report 96-045, NEC Research Institute*, 1996.
19. H. D. Brunk, *An Introduction to Mathematical Statistics*. Boston: Ginn and Company, 1960.