

LOSSLESS DATA HIDING FOR ELECTRONIC INK

Hong Cao and Alex C. Kot

School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore

ABSTRACT

This paper presents a novel lossless data hiding technique for electronic inks. The proposed algorithm first computes the analytical ink-curve function for each stroke as a set of smoothly concatenated cubic Bezier curves. During embedding, a set of data carrier points on the ink-curve are computed, perturbed and inserted back into the original point array together with some marker points. In extraction, data carrier points are identified and compared with their original positions in order to decode the secret message. Through experiments, we demonstrate that significant amount of secret message can be embedded and the ink-curve computed from the marked ink data closely resembles the original ink-curve.

Index Terms— Authentication, Bezier curve, data hiding, electronic ink, lossless, Tablet PC

1. INTRODUCTION

The advent of Tablet PCs has given birth to a new and emerging data type named “ink”. With “ink”, Tablet PC users can preserve their electronic handwriting in ink format and none of the original handwriting features are lost. The electronic ink need not be converted to text or images in order to be saved, sent, resized or otherwise manipulated by pen-based applications or exported across different platforms.

As the world is moving towards electronic automation, the electronic inking technology is getting popular with stronger software support and it gradually replaces our traditional way of writing in various verticals of our life, e.g. in education and healthcare industry. However, similar to other digital media types such as image and videos, electronic inks can be easily tampered without any traces left. This affects people’s confidence in using electronic inks especially when security is needed. In this paper, we consider protecting the raw electronic ink data through lossless data hiding for several reasons: 1) the protection is self-contained as the protection data can be seamlessly integrated with the raw ink data with little perceptual differences; 2) hybrid image authentication systems of data hiding and cryptography have been demonstrated to be

highly secure [4-6]; 3) the original ink data can be recovered hence the temporal features are preserved.

To the best of our knowledge, this is the first lossless data hiding work proposed for electronic inks. Several related early works either embeds secret message into image curves on a map [3] or into parametrically represented curves such as non-uniform rational B-spline (NURBS) curves [1, 2]. These methods are designed either for fingerprint tracing or copyright protection. Very often, the cover media is required to decode a hidden message, which makes them inapplicable to the authentication scenarios where the cover media is not present.

This paper is organized as follows. Section 2 describes our parametric ink-curve representation as smoothly concatenated cubic Bezier curves. Section 3 describes the embedding and extraction algorithms. Section 4 shows some experimental results and section 5 concludes this paper.

2. INK CURVE REPRESENTATION

A raw ink stroke typically consists of a discrete set of points $\{\mathbf{p}_n = (x_n, y_n), \text{ for } 1 \leq n \leq N\}$ sampled from the trajectory when the stylus pen travels on a digitizer surface. Our proposed algorithm employs a piecewise cubic Bezier curve [7] model to represent the ink-curve. The unknown off-curve Bezier control points are computed by iteratively reinforcing smoothness constraints at the locations where two consecutive curve pieces segments meet. In the following section, we describe the steps in greater details.

2.1. Piecewise Cubic Bezier Curve

A cubic Bezier curve is defined on a third-order Bernstein basis and each curve segment is associated with two off-curve Bezier control points whose relative positions determine the curve shape. As illustrated in Fig. 1, if we join every two consecutive segment points with a cubic Bezier curve segment and the overall ink-curve $\mathbf{f}(t) = (x(t), y(t))$, where t is a continuous variable, can be written as

$$\mathbf{f}(t) = \begin{cases} \mathbf{s}_1(t) & \text{for } t_1 \leq t < t_2 \\ \mathbf{s}_2(t) & \text{for } t_3 \leq t < t_3 \\ \vdots & \\ \mathbf{s}_{N-1}(t) & \text{for } t_{N-1} \leq t \leq t_N \end{cases} \quad (1)$$

• $\{\mathbf{p}_n\}$ for $1 \leq n \leq N$: segment points

□ $\{\mathbf{b}_{1,n}, \mathbf{b}_{2,n}\}$ for $1 \leq n \leq N-1$: off-curve Bezier control points

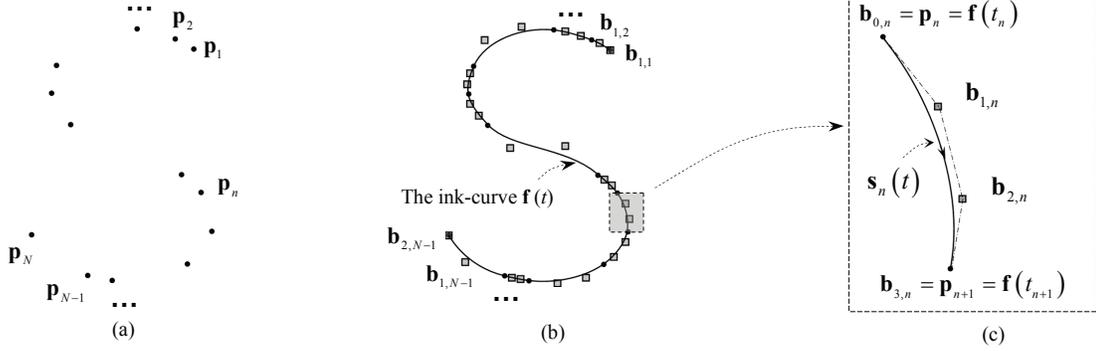


Fig. 1 An ink curve ‘S’ as smoothly concatenated cubic Bezier curves, (a) Raw ink as a discrete set of segment points; (b) The computed Bezier control points and the corresponding ink-curve; (c) An enlarged view of the n^{th} cubic Bezier curve

where $\mathbf{s}_n(t) = \sum_{i=0}^3 \mathbf{b}_{i,n} Z_{i,n}(t)$ is the n^{th} piece of cubic Bezier curve, $\{\mathbf{b}_{i,n} = [b_{i,n}^{(x)}, b_{i,n}^{(y)}]\}$ for $0 \leq i \leq 3$ denote the corresponding Bezier control points and the cubic Bernstein basis for the n^{th} piece is defined as

$$Z_{i,n}(t) = \binom{3}{i} \left(\frac{t-t_n}{t_{n+1}-t_n} \right)^i \left(\frac{t_{n+1}-t}{t_{n+1}-t_n} \right)^{3-i} \quad \text{for } t_n \leq t \leq t_{n+1} \quad (2)$$

and $t_1 < \dots < t_n < \dots < t_N$ are the variables that satisfy

$$\mathbf{f}(t_n) = \mathbf{s}_{n-1}(t_n) = \mathbf{s}_n(t_n) = \mathbf{p}_n \quad (3)$$

Since $\mathbf{f}(t)$ is continuous, the on-curve Bezier control points are $\mathbf{b}_{0,n} = \mathbf{p}_n$ and $\mathbf{b}_{3,n} = \mathbf{p}_{n+1}$. By evaluating the left-side and right-side derivatives of $\mathbf{f}(t)$ at t_n , we obtain

$$\begin{aligned} \mathbf{f}'(t_n^-) &= \mathbf{s}'_{n-1}(t_n) = 3(\mathbf{b}_{3,n-1} - \mathbf{b}_{2,n-1}) / (t_n - t_{n-1}) \\ \mathbf{f}'(t_n^+) &= \mathbf{s}'_n(t_n) = 3(\mathbf{b}_{1,n} - \mathbf{b}_{0,n}) / (t_{n+1} - t_n) \end{aligned} \quad (4)$$

or equivalently

$$\begin{aligned} \mathbf{b}_{2,n-1} &= \mathbf{b}_{3,n-1} - \mathbf{f}'(t_n^-) \times (t_n - t_{n-1}) / 3 \\ \mathbf{b}_{1,n} &= \mathbf{b}_{0,n} + \mathbf{f}'(t_n^+) \times (t_{n+1} - t_n) / 3 \end{aligned} \quad (5)$$

2.2. Off-Curve Bezier Control Points

To analytically represent each curve segment, we need to determine the off-curve Bezier control points $\{\mathbf{b}_{1,n}, \mathbf{b}_{2,n}\}$. Equation (5) shows that $\mathbf{b}_{1,n}$ and $\mathbf{b}_{2,n}$ are computable from $\mathbf{d}_n^+ = \mathbf{f}'(t_n^+) \times (t_{n+1} - t_n)$ and $\mathbf{d}_{n+1}^- = \mathbf{f}'(t_{n+1}^-) \times (t_{n+1} - t_n)$. Since $\mathbf{f}'(t_n^+)$, $\mathbf{f}'(t_{n+1}^-)$ and $t_{n+1} - t_n$ are all unknown, \mathbf{d}_n^+ and \mathbf{d}_{n+1}^- cannot be directly computed. Assuming the ink-curve is C^2 -smooth (i.e. up to second-order derivatives are continuous) at each joint where two curve segments meet, we can approximate \mathbf{d}_n^+ and \mathbf{d}_n^- in the following steps:

Step 1: Initialize the ratios $\{r_n = (t_{n+1} - t_n) / (t_n - t_{n-1}) = 1\}$;

Step 2: With known $\{r_n\}$, based on the right-side and the left-side sliding windows of \mathbf{p}_n respectively, the estimates of $\{\hat{\mathbf{d}}_n^+, \hat{\mathbf{d}}_n^-\}$ can be obtained from a regularized least square solution [8] by employing a second-order Taylor’s approximation;

Step 3: Assume perfect C^1 -smoothness condition is met at each joint, update the ratios using

$$r_n = \|\hat{\mathbf{d}}_n^+\| / \|\hat{\mathbf{d}}_n^-\| \quad (6)$$

where $\|\cdot\|$ denotes Frobenius norm;

Step 4: Repeat step 2 and 3 until the ratios $\{r_n\}$ stabilizes. Experimentally, we found that the change of $\{r_n\}$ sharply reduces for a few initial iterations and gradually stabilize to a low-level for all 20 test ink strokes we captured from a digitizer;

Step 5: Based on the refined $\{r_n\}$ and a central sliding window of \mathbf{p}_n , we compute $\{\mathbf{d}_n^+, \mathbf{d}_{n+1}^-\}$ as a regularized least-square solution;

Step 6: Compute $\{\mathbf{b}_{1,n}, \mathbf{b}_{2,n}\}$ using equation (5).

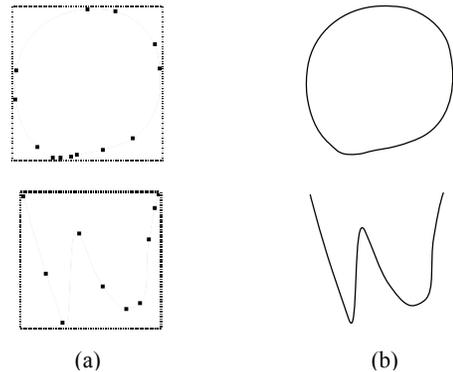


Fig. 2 Computed Ink Curves; (a) Raw Ink Strokes and (b) the Corresponding Ink-Curves

With $\{\mathbf{b}_{0,n}, \mathbf{b}_{1,n}, \mathbf{b}_{2,n}, \mathbf{b}_{3,n}\}$ computed for $1 \leq n < N$, all curve segments are known analytically. A large number of raw ink strokes are used to display ink-curves based on the steps

above. In all cases, our method generates visually pleasing smooth curves from a limited number of raw ink points and Fig 2 shows two such examples.

3. EMBEDDING AND EXTRACTION

3.1. Embedding

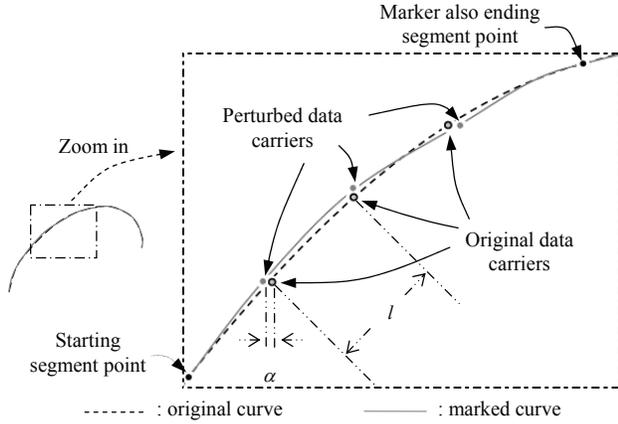


Fig. 3 Embedding in a curve segment

A secret message is embedded in several steps: selection of embeddable curve segments, evaluation of data carriers, perturbation and insertion of perturbed data carriers into the original point array. Heuristic rules are designed to first select a set of embeddable curve segments with large arc-distance. As illustrated in Fig. 3, for each selected segment, we then compute a set of data carriers $\{q_j\}$ on the curve to divide the curve segment evenly and the minimum straight-line distance between two consecutive data carriers is no less than a certain threshold l . To embed a secret message $\{w_i\}$, we first partition it and map it into a sequence of unit-length perturbation vectors $\{u_j\}$ through a pre-defined lookup table and the perturbed data carriers $\{\hat{q}_j\}$ is derived by $\hat{q}_j = q_j + \alpha u_j$, where α is a scaling factor controlling the perturbation strength. The perturbed data carriers are then inserted back into the point array of the original ink. Besides the data carriers, for each embeddable segment, we also insert a unique marker point to signify the end of an embedding stream and this marker is chosen to be identical to the ending segment point as illustrated in Fig. 3.

3.2. Extraction

In extraction, compatible heuristic rules together with detection of the marker points are used to separate the embedded data carriers $\{\hat{q}_j\}$ and the original points. Based on the original point array, we re-compute the original ink-curve as well as the original set of data carriers $\{q_j\}$. By subtracting $\{\hat{q}_j\}$ from $\{q_j\}$, we derive a sequence of scaled perturbation vectors and the secret message is

decoded from these perturbation vectors using the same lookup table as in the embedding process.

3.3. Payload, Distortion and Parameter Selection

The payload, i.e. the amount of secret message that can be embedded, depends on both the total number of data carriers as well as the number of bits D that each data carrier can carry. The total number of data carriers is proportional to the total arc length of the embeddable curve segments and inversely proportional to l . With total arc-length fixed for a given cover ink, the smaller l we choose, the denser the data carriers, hence the larger the payload. Another important parameter is the scaling factor α , which is usually chosen large enough to withstand some practical distortions, e.g. quantization error due to limited precision but not too large to affect fidelity of the original curve.

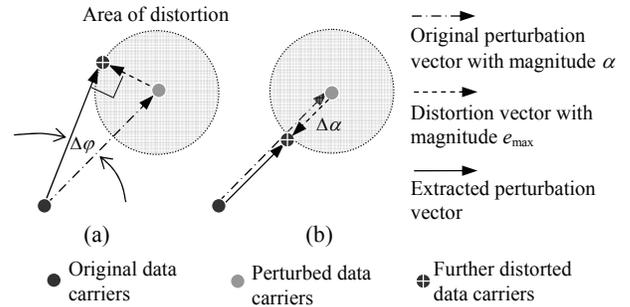


Fig. 4 Worst distortion scenarios for perturbation vectors: (a) Maximal angular distortion and (b) maximal magnitude distortion

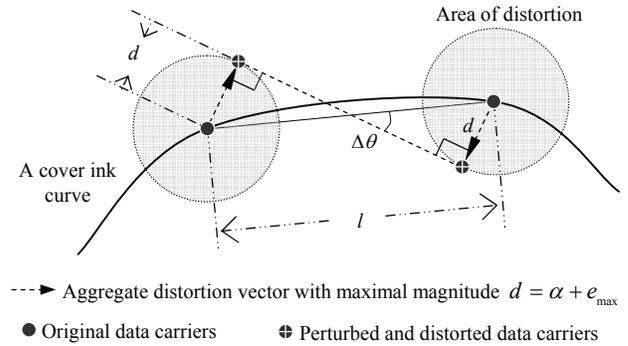


Fig. 5 A worst angle distortion scenario for a tiny straight-line segment bounded by two adjacent data carriers

Assuming the practical distortion e on the perturbed data carriers is bounded by e_{max} , the lower bound of α can be computed to ensure successful extraction under two worst distortion scenarios of perturbation vectors shown in Fig. 4. Practically, we choose α to be close to its lower-bound value in order to have least ink-curve distortion. With α selected, we study another worst distortion scenario is in Fig. 5, where the maximal angular distortion $\Delta\theta$ of a tiny straight-line segment bounded by two consecutive data carriers can be computed. It is usually desirable to constraint

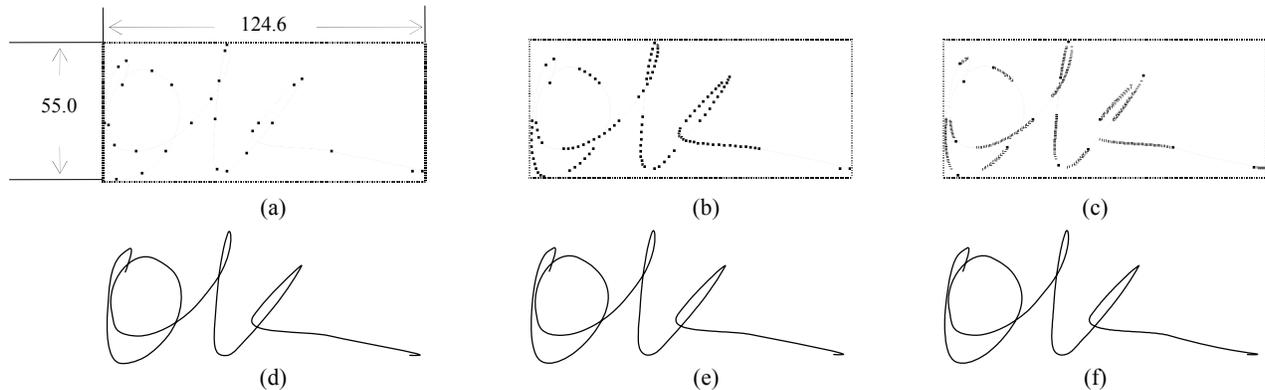


Fig. 6 Data hiding in “ok” ink where black dots denote the segment points; (a) Cover “ok” ink as a discrete set of points; (b) Marked “ok” ink after embedding 282 bits with 94 data carriers and 14 markers inserted ($\alpha=0.002$, $l=2$); (c) Marked “ok” ink after embedding 1380 bits with 460 data carriers and 14 markers inserted ($\alpha=0.002$, $l=0.5$); (d), (e) and (f) are ink-curves of (a), (b) and (c) respectively

$\Delta\theta$ to be smaller than a threshold $\Delta\theta_{\max}$ in order to ensure that no noticeable zigzags can be observed on the marked ink-curve, especially at the smooth curve region. $\Delta\theta_{\max}$ can be selected through perceptual study. From this distortion scenario, we can determine the lower bound of parameter l based on e_{\max} and the predefined α and $\Delta\theta_{\max}$.

4. EXPERIMENTAL RESULTS

Experimentally, we have tested the proposed lossless data hiding technique with more than 100 handwriting ink strokes captured from a digitizer. In all cases, the embedded pseudo-random messages are correctly extracted and the cover inks are losslessly recovered. Fig. 6 shows an embedding example for the cover ink “ok”, where $\alpha=0.002$ is chosen and $\Delta\theta_{\max}=1^\circ$. The angle of each perturbation vector is used to encode 3 secret bits ($D=3$), which correspond to 8 equal-size angular zones defined in the lookup table. From the results, we can see that significant amount of bits can be embedded and the payload is largely dependant on l . A smaller l typically results in much denser data carriers and significantly increased payload. In both cases ($l=2$ and $l=0.5$), the marked ink-curves closely resembles the original ink-curve. Since people usually see the ink-curves instead of the raw ink data, little human-perceptible distortions have been introduced to the ink-curves even if a large number of points have been inserted.

With a similar framework to [6], the security of this lossless data hiding technique can be enhanced by the existing public-key infrastructure (PKI) to secure integrity of electronic inks from malicious tampering.

5. CONCLUSION

In this paper, we have presented a novel lossless data hiding approach for electronic inks through insertion of data-carrier points into the original point array of a cover ink. We first compute the ink-curve as a set of smoothly concatenated cubic Bezier curves by iteratively reinforcing

a smoothness constraint at the joints. Experimental results show our method tends to generate visually pleasing smooth ink-curves. Once the ink-curve is known, a set of data-carrier points on the ink-curve are evaluated, perturbed and inserted back into the original points array to carry the secret message. As the embedded points can be extracted and the original ink data can be restored, lossless data hiding is achieved. With properly selected parameters, results of the data hiding experiment show that large amount of secret data can be embedded and at the same time, the marked ink-curves still closely resemble their original curves.

6. ACKNOWLEDGEMENT

This work is supported by grant from Microsoft Research Asia (MSRA).

7. REFERENCES

- [1] R. Ohbuchi, H. Masuda and M. Aono, “A Shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces,” in *Proc. CGI*, pp. 180-187, 1999
- [2] J. J. Lee, N. I. Cho, and S. U. Lee, “Watermarking Algorithms for 3D NURBS Graphic Data,” *EUROSIP J. Applied Signal Processing*, no. 14, pp. 2142-2152, 2004
- [3] H. Gou and M. Wu, “Data Hiding in Curves with Application to Fingerprinting Maps”, *IEEE Trans. on Signal Processing*, vol. 53-10, pp. 3988-4005, 2005
- [4] P. W. Wong and N. Memon, “Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification”, *IEEE Trans. Image Processing*, vol. 10-10, 2001
- [5] H. Y. Kim and A. Afif, “Secure Authentication Watermarking for Binary Images”, in *Proc. Sibgraphi-Brazilian Symposium on Computer Graphics and Image Processing*, pp. 199-206, 2003
- [6] H. Yang and A. C. Kot. “Pattern-Based Data Hiding for Binary Image Authentication by Connectivity-Preserving”, *IEEE Trans on Multimedia*, vol. 9-3, pp. 475-486, 2007
- [7] H. Prautzsch, W. Boehm and M. Paluszny, *Bezier and B-spline Techniques*, Springer, 2002
- [8] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, Philadelphia: SIAM, 1998