ROBUST DIGITAL FINGERPRINTING FOR CURVES

Hongmei Gou and Min Wu

Electrical & Computer Engineering Department, University of Maryland, College Park

ABSTRACT

Hiding data in curves can be achieved by parameterizing a curve using the B-spline model and adding spread spectrum sequences in B-spline control points. In this paper, we propose an iterative alignment-minimization algorithm to perform curve registration and deal with the non-uniqueness of B-spline control points. We demonstrate through experiments the robustness of our method against various attacks such as collusion, geometric transformation, and printing-and-scanning. We also show the feasibility of our method for fingerprinting topographic maps and detecting fingerprints from printed copies.

1. INTRODUCTION

Curve is one of the major components appearing in maps, 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 the TabletPC). Digital maps and drawings are also generated directly by various computer programs such as map-making software and CAD 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, intelligence, and commercial operations. For example, trace-and-track capabilities can be provided through invisibly embedding a unique ID, referred to as a *digital fingerprint*, to each copy of a document before distributing to users [1].

As a forensic mechanism to deter information leakage and to trace traitors, digital fingerprint 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 different copies to generate a new copy in which the original fingerprints are removed or attenuated [1]; geometric transformations such as rotation, scaling, and translation (RST); and D/A-A/D conversions such as printing-and-scanning.

The curve-based documents, such as maps and handwritten notes, can be represented as binary bitmap images (raster representation) or as a set of vectors. In the existing data embedding works for binary images [2][3], the fragility of the embedding and the reliance on precise sampling of pixels for correct decoding pose challenges in surviving malicious removal in fingerprinting applications. As for watermarking vector graphics, vertices are perturbed through Fourier descriptors of polygonal lines [4] or spectral analysis of mesh models [5] to embed copyright marks. However, the embedding in [4] introduces visible distortions, and the approach in [5] has high complexity resulting from mesh spectral analysis.

In our previous work [6], we have proposed a new data hiding technique for curves by identifying and manipulating curve parameters. In particular, a set of control points from the B-spline model forms a compact collection of salient features representing the shape of curve. In such a feature domain, we add mutually independent, noise-like sequences as digital fingerprints to the coordinates of the control points. This additive spread spectrum embedding and the associated correlation based detection generally provide a good tradeoff between imperceptibility and robustness.

To determine which fingerprint sequence(s) are present in a test curve, we first need to perform registration using the original unmarked curve that is commonly available to a detector in fingerprinting applications [1][7]. The 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. Through the B-spline based data embedding and detection plus the proposed IAM algorithm, we are capable of building robust curve fingerprinting systems that can sustain a number of challenging attacks such as collusion, geometric transformations, and printing-and-scanning.

The paper is organized as follows. Section 2 briefly reviews the B-spline based data hiding in curves. Section 3 details the proposed iterative alignment-minimization algorithm for robust fingerprint detection. Experimental results on fingerprinting topographic maps are presented in Section 4 and conclusions are drawn in Section 5.

2. B-SPLINE BASED CURVE FINGERPRINTING

B-splines are piecewise polynomial functions that provide local approximations of curves using a small number of pa-

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

rameters known as the *control points* [8]. Let $\{\mathbf{p}(t)\}$ denote a curve, where $\mathbf{p}(t) = (p_x(t), p_y(t))$ and t is a continuous time variable. Its B-spline approximation can be written as

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

where $\mathbf{c}_i = (c_{x_i}, c_{y_i})$ is the i^{th} control point, and $B_{i,k}(t)$ is a corresponding 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 a $(m+1)\times 2$ matrix $\mathbf{P} \triangleq (\mathbf{p}_x, \mathbf{p}_y)$ and coordinates of $n+1$ control points as a $(n+1)\times 2$ matrix $\mathbf{C} \triangleq (\mathbf{c}_x, \mathbf{c}_y)$, we can write the problem with its solution as

$$\begin{cases} \mathbf{p}_x \approx \mathbf{B}\mathbf{c}_x \\ \mathbf{p}_y \approx \mathbf{B}\mathbf{c}_y \end{cases} \Longrightarrow \begin{cases} \mathbf{c}_x = \mathbf{B}^{\dagger}\mathbf{p}_x \\ \mathbf{c}_y = \mathbf{B}^{\dagger}\mathbf{p}_y \end{cases}, \tag{2}$$

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

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 the set of control points obtained before and get a set of marked control points $(\mathbf{c}_x + \alpha\mathbf{w}_x, \mathbf{c}_y + \alpha\mathbf{w}_y)$. A fingerprinted curve can then be constructed by the B-spline model (equation (1)) using these marked control points.

To detect the fingerprint sequence(s) embedded in a test curve, assuming we have a set of test 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 $(\tilde{\mathbf{c}}_x, \tilde{\mathbf{c}}_y)$ by applying $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y)$ to equation (2). Then we compute the difference between the coordinates of the test and the original control points to arrive at an estimated fingerprint sequence $(\tilde{\mathbf{w}}_x, \tilde{\mathbf{w}}_y) = (\frac{\tilde{\mathbf{c}}_x - \mathbf{c}_x}{\alpha}, \frac{\tilde{\mathbf{c}}_y - \mathbf{c}_y}{\alpha})$. We further evaluate the similarity between this estimated fingerprint sequence and each fingerprint sequence in our database through a correlation-based Z statistic. If the similarity is higher than a threshold (usually set around 3 to 6 for Z statistics), 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. The details on the basic embedding and detection can be found in [6].

3. ROBUST FINGERPRINT DETECTION

The set of test sample points $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y)$ assumed in Section 2 may not always be available, especially when a test curve undergoes geometric transformations, and/or is scanned from a printed hard copy. Preceding the basic fingerprint detection module there must be a pre-processing registration step to align the test curve with the original curve. In order to improve accuracy and efficiency of the registration, an automatic registration is desirable.

Another issue related to the assumed test sample points is the inherent non-uniqueness of B-spline control points, which refers to the fact that a curve can be well approximated by different sets of B-spline control points. With a different choice of the sample points, we may induce a quite different set of control points that can still describe the same curve accurately. Therefore, if we can not find the set of test 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 time values $\{s_j\}$) and control points, we try to find the set of sample points on 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 non-uniqueness issue of B-spline control points can be addressed through finding the point correspondence.

3.1. Problem Formulation

We now formulate the curve registration and point correspondence problem in the context of fingerprint detection.

We use "View-I" to refer to the geometric setup of the original unmarked curve and "View-II" the setup of the test curve. Thus we can register these two curves by transforming the test curve from "View-II" to "View-I", or the original curve from "View-I" to "View-II". We focus on registration under affine transformations, which can represent combinations of scaling, rotation, translation, and shearing.

We call two points (x, y) and $(\tilde{x}, \tilde{y})$ *affine related* if

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{3}$$

where $\{a_{ij}\}$ are parameters representing the collective effect of scaling, rotation, translation, and shearing. These transform 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, the inverse transform can be represented by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_x^T \\ \mathbf{a}_y^T \\ 0\ 0\ 1 \end{bmatrix}^{-1} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{g}_x^T \\ \mathbf{g}_y^T \\ 0\ 0\ 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix}. \tag{4}$$

The original curve available to the fingerprint detector can be a raster curve or a vector curve. The detector also knows the set of original sample points $(\mathbf{p}_x, \mathbf{p}_y)$ 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 points $(\tilde{\mathbf{v}}_x, \tilde{\mathbf{v}}_y)$ or a raster curve with pixel coordinates $(\tilde{\mathbf{r}}_x, \tilde{\mathbf{r}}_y)$. Without the set of test sample points $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y)$ assumed in Section 2, both transform parameters for curve registration and the point correspondence must be estimated in order to locate the fingerprinted

control points successfully. As a vector curve can be rendered as a raster curve by interpolation, we consider that the original and the test curves are represented in raster format and formulate the problem as:

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.*

3.2. Iterative Alignment-Minimization (IAM) Algorithm

Taking an existing curve alignment method [9] as a building block, we propose an Iterative Alignment-Minimization (IAM) algorithm that can perform curve registration and solve the point correspondence problem simultaneously. The IAM algorithm consists of three main steps.

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 N and $\tilde{N}$ be the number of points on the original and the test raster curve, 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 $j_0 < j_1 < j_2 < \ldots < j_m$ are integers ranging from 0 to $N-1$, we estimate the indices of the test curve's $m+1$ sample points by $\tilde{\mathbf{J}} = round\left(\frac{\tilde{N}-1}{N-1} \cdot \mathbf{J}\right)$. Using this estimated index vector $\tilde{\mathbf{J}}$, we identify the corresponding sample points from the test curve and take them as the initial estimate.

2) Curve Alignment with the Estimated Sample Points:

Given the estimated test sample points $(\tilde{\mathbf{p}}_x^{(i)}, \tilde{\mathbf{p}}_y^{(i)})$ in the i^{th} iteration, we apply the curve alignment method in [9] to estimate transform parameters and 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 $(\mathbf{a}_x^{(i)}, \mathbf{a}_y^{(i)})$. The estimated sample points on the test curve can be transformed back to View-I by $(\mathbf{g}_x^{(i)}, \mathbf{g}_y^{(i)})$. 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 [9]) and search for both the transform 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

$$\left\| \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix} \hat{\mathbf{c}}_x^{(i)} - \begin{bmatrix} \mathbf{p}_x \\ \tilde{\mathbf{P}}^{(i)} \hat{\mathbf{g}}_x^{(i)} \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mathbf{B} \\ \mathbf{B} \end{bmatrix} \hat{\mathbf{c}}_y^{(i)} - \begin{bmatrix} \mathbf{p}_y \\ \tilde{\mathbf{P}}^{(i)} \hat{\mathbf{g}}_y^{(i)} \end{bmatrix} \right\|^2, \tag{5}$$

where $\tilde{\mathbf{P}}^{(i)} \triangleq [\ \tilde{\mathbf{p}}_x^{(i)} \quad \tilde{\mathbf{p}}_y^{(i)} \quad \mathbf{1}\ ]$ and $\mathbf{1}$ is a column vector with all $1's$. 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 of the solution to this optimization problem. Thus we obtain an estimate of the transform parameters and the B-spline control points as

$$\begin{cases} \hat{\mathbf{g}}_x^{(i)} = \mathbf{C}^{(i)}\mathbf{D}^{(i)}\mathbf{p}_x, \quad \hat{\mathbf{g}}_y^{(i)} = \mathbf{C}^{(i)}\mathbf{D}^{(i)}\mathbf{p}_y \\ \hat{\mathbf{c}}_x^{(i)} = \mathbf{D}^{(i)}\mathbf{p}_x, \quad \hat{\mathbf{c}}_y^{(i)} = \mathbf{D}^{(i)}\mathbf{p}_y \end{cases}, \\ where \begin{cases} \mathbf{C}^{(i)} \triangleq \left(\tilde{\mathbf{P}}^{(i)T}\tilde{\mathbf{P}}^{(i)}\right)^{\dagger} \tilde{\mathbf{P}}^{(i)T}\mathbf{B} \\ \mathbf{D}^{(i)} \triangleq \left(2\mathbf{B}^T\mathbf{B} - \mathbf{B}^T\tilde{\mathbf{P}}^{(i)}\mathbf{C}^{(i)}\right)^{\dagger} \mathbf{B}^T \end{cases}. \tag{6}$$

The estimated control points $(\hat{\mathbf{c}}_x^{(i)}, \hat{\mathbf{c}}_y^{(i)})$ can be used to estimate the embedded fingerprint sequence and further compute the detection statistic $Z^{(i)}$, as described in Section 2.

3) Refinement of Sample Point Estimation on Test Curve:

Given the estimated transform 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:

$$\begin{cases} \tilde{\mathbf{r}}_{x,I}^{(i)} = [\tilde{\mathbf{r}}_x \quad \tilde{\mathbf{r}}_y \quad \mathbf{1}]\, \hat{\mathbf{g}}_x^{(i)} \\ \tilde{\mathbf{r}}_{y,I}^{(i)} = [\tilde{\mathbf{r}}_x \quad \tilde{\mathbf{r}}_y \quad \mathbf{1}]\, \hat{\mathbf{g}}_y^{(i)} \end{cases}. \tag{7}$$

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 the *nearest neighbor* rule to re-estimate 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 and denote it 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:

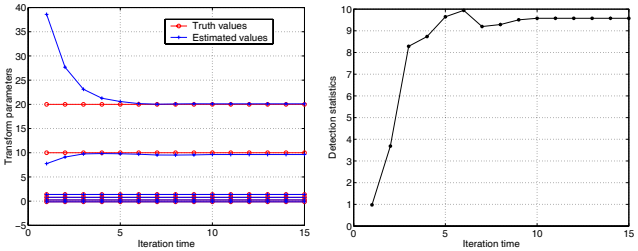
$$\begin{cases} \tilde{\mathbf{p}}_x^{(i+1)} = \begin{bmatrix} \tilde{\mathbf{p}}_{x,I}^{(i+1)} & \tilde{\mathbf{p}}_{y,I}^{(i+1)} & \mathbf{1} \end{bmatrix} \hat{\mathbf{a}}_x^{(i)} \\ \tilde{\mathbf{p}}_y^{(i+1)} = \begin{bmatrix} \tilde{\mathbf{p}}_{x,I}^{(i+1)} & \tilde{\mathbf{p}}_{y,I}^{(i+1)} & \mathbf{1} \end{bmatrix} \hat{\mathbf{a}}_y^{(i)} \end{cases}. \tag{8}$$

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.

3.3. Detection Example and Robustness Analysis

We present a detection example employing the proposed IAM algorithm on a curve taken from a topographic map. Shown in Figure 1(a) are the six estimated transform parameters after each iteration, showing an accurate curve registration. Upon convergence, we use the estimated control points to perform detection with the fingerprint involved. The high fingerprint detection statistic value shown in Figure 1(b) suggests the positive identification of the correct fingerprint.

With good estimation of affine transform parameters, the IAM algorithm is resilient to combinations of scaling,



(a) (b)

Fig. 1. Detection example using the IAM algorithm: a) estimated transform parameters; c) detection statistics.

rotation, translation, and shearing. The explicit estimation of point correspondence provides resilience against vector-raster conversion. Along with the robustness from the spread spectrum embedding in B-spline control points, our method can resist a number of challenging attacks and distortions.

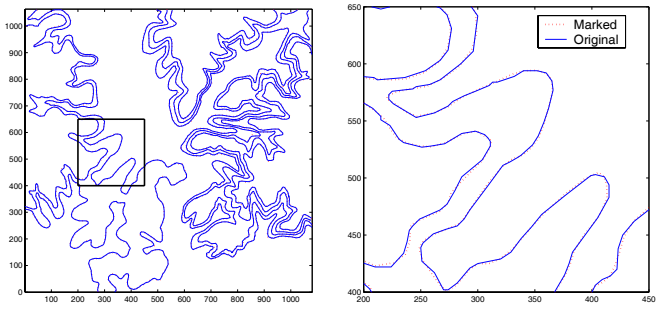
4. EXPERIMENTAL RESULTS

We now present experimental results of our approach in the context of tracing and tracking topographic maps.

Fingerprinted Topographic Maps Taking a 1100×1100 topographic vector map from http://www.ablesw.com as the original map, we mark nine curves that are sufficiently long and a total of 1331 control points are used to carry the fingerprint. We overlay in Figure 2(a) these nine original and 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 Figure 2(b). We can see that the fingerprinted map preserves the geospatial information in the original map up to a high precision.

Resilience to Vector-Raster Conversion and Affine Transformation We now examine the resilience to vector-raster conversion coupled with possible affine transformation. A fingerprinted vector map is first rendered as a 1100×1100 image and then transformed by 10-degree rotation, 80% and 140% scaling in X and Y direction, respectively, and 10- and 20-pixel translation in X and Y direction, respectively. We apply the proposed IAM algorithm to estimate the transform parameters and get a Z detection statistic of ***20.84*** with the fingerprint involved. This suggests that the embedded fingerprint is identified with high confidence.

Resilience to Collusion and Printing-and-Scanning To show the robustness of our approach against the combinational attack of collusion and printing-and-scanning, we first generate a colluded map by averaging coordinates of the control points from four users' fingerprinted maps, then render it and print it out using a HP laser printer, and finally scan back as a binary image by a Canon scanner with 360dpi 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. By using the proposed IAM algorithm, we get Z statistics of ***10.54, 11.74, 10.67, 6.93*** for the four colluders, indicating that the embedded fingerprints for all the four colluders survive this combinational attack thus the sources of leakage for this map can be identified.



(a) (b)

Fig. 2. Fingerprinting maps: (a) original and fingerprinted curves overlaid with each other; (b) enlarged difference.

5. CONCLUSIONS

Based on our new data hiding method for curves, we have proposed an iterative alignment-minimization algorithm to allow for robust fingerprint detection under unknown geometric transformations and in absence of explicit point correspondence. We have demonstrated the robustness of our method against various challenging attacks.

6. REFERENCES

[1] M. Wu, W. Trappe, Z. Wang, and K.J.R. Liu, "Collusion resistant fingerprinting for multimedia," *IEEE Signal Processing Magazine*, pp. 15–27, March 2004.

[2] E. Koch and J. Zhao, "Embedding robust labels into images for copyright protection," *Proc. Intellectual Property Rights for Specicalized Info., Knowledge and New Tech.*, 1995.

[3] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *IEEE Trans. Multimedia*, Aug. 2004.

[4] V. Solachidis and I. Pitas, "Watermarking polygonal lines using fourier descriptors," *IEEE Computer Graphics and Applications*, pp. 44–51, May/June 2004.

[5] R. Ohbuchi, H. Ueda, and S. Endoh, "Watermarking 2D vector maps in the mesh-spectral domain," *Proc. of the Shape Modeling International*, 2003.

[6] H. Gou and M. Wu, "Data hiding in curves for collusion-resistant digital fingerprinting," *to appear in ICIP 2004*.

[7] I. Cox, J. Bloom, and M. Miller, *Digital Watermarking*, Morgan Kaufmann, 2001.

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

[9] M. Xia and B. Liu, "Image registration by 'super-curves'," *IEEE Trans. on Image Processing*, pp. 720–732, May 2004.