# Watermarking Polygonal Lines Using Fourier Descriptors

**Vassilios Solachidis and Ioannis Pitas**
*Aristotle University of Thessaloniki*

**To protect the copyright of vector images, the authors embed a watermark in the image's contours, making it robust against a number of attacks.**

Digital products are easy to copy, reproduce, and maliciously process in a network environment. Over the past decade, watermarking has emerged as an important technology for protecting copyright of multimedia products. A *watermark* is a digital signal carrying information about the copyright owner that is permanently embedded into a digital product, making the product robust to any alteration (deliberate or unintentional) of its content. (The "Watermarking Basics" sidebar gives an overview of watermarking techniques and principles.) At the same time, this watermark data embedding shouldn't cause the multimedia product to suffer severe quality degradation.

In addition to bitmap images, many multimedia applications use vector graphics images—for example, digital maps, geographic information system (GIS) data, 2D graphics, and cartoon images. Classical watermarking methods embed a watermark by applying luminance alterations to the image. These watermarking methods aren't suitable for vector graphics images because in such images watermarks are perceivable and easily removed.

Vector graphics images usually consist of large homogeneous regions, which often have uniform luminance value. Thus, even a small perturbation on the luminance values is easily visible in these regions. Furthermore, the most important information of vector graphics is stored in the contours. Classical watermarking methods embed the watermark in the homogeneous regions. Even if we could achieve minor visual changes on such images by embedding a low-power watermark, a pirate could easily corrupt the watermark—for example, by applying a malicious perturbation that would affect only the homogeneous region intensity (by quantization or by filtering the homogeneous image regions). Thus, for these image types, altering the contours (and not the luminance) is essential for imperceptible and robust watermarking.

We present a method for embedding and detecting watermarks in vector graphics images containing polygonal lines, and present results from simulated attacks.

## Our system

Our proposed algorithm embeds watermarks in polygonal lines describing image contours. These contours are usually described in vector format. Our watermarking method slightly modifies the vertex coordinates of the polygonal line. We embed the watermark in the magnitude of the curve's Fourier descriptors to exploit its location, scale, and rotation invariant properties. Watermark detection is blind—that is, the watermark detection procedure doesn't need the original polygonal line.

Our technique has certain similarities to bitmap image watermarking in the discrete Fourier transform (DFT) domain. However, it's essentially different because it can be applied to vector rather than bitmap images. Furthermore, Fourier descriptor watermarking problems differ from DFT bitmap image watermarking problems. For example, watermarks can cause curve discontinuity or curve self-crossing. Such undesirable side effects don't exist in DFT bitmap image watermarking.

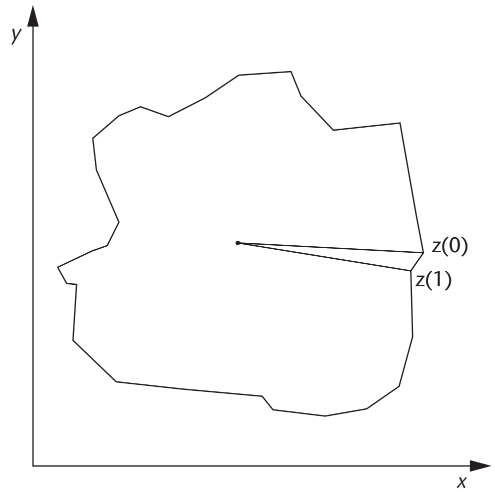
### *Fourier descriptors*

Let $L$ be a closed polygonal line that consists of $N$ vertices. Let also $[x(n), y(n)]$, $n = 0, 1, \ldots, N-1$ be the Cartesian coordinates of each vertex. We construct the complex signal $z(n)$, where $z(n) = x(n) + i \cdot y(n)$, $n = 0, 1, \ldots, N-1$, by randomly selecting a vertex as a starting point and tracing the vertices in a clockwise or counterclockwise direction, as Figure 1 shows. As we describe in the next section, the watermark embedding and detection procedures don't require knowledge of the starting point or the traversing path.

Let $Z(k)$ be the Fourier transform representation of the signal $z(n)$:

$$Z(k) = \sum_{n=0}^{N-1} z(n) \exp\left(\frac{-j2\pi kn}{N}\right), 0 \le k \le N-1$$


Published by the IEEE Computer Society




**1** **Complex signal representation of a polygonal line.**

The inverse transform is

$$z(n) = \frac{1}{N}\sum_{k=0}^{N-1} Z(k)\exp\left(\frac{j2\pi kn}{N}\right), 0 \le n \le N-1$$

We call the complex coefficients $Z(k)$ *Fourier descriptors* of the polygonal line.

In the Fourier descriptor representation, the $Z(k)$ coefficients around $Z(0)$ (DC term) describe low-frequency information, whereas the coefficients around $Z(N/2)$ describe high-frequency information. Low-frequency coefficients typically represent the general shape, whereas outline and high-frequency coefficients describe shape details and sharp corners. The $Z(0)$ coefficient represents the center of gravity of the shape ($Z(0) = \sum z(n)$), which is why translations are only effective on the DC term $Z(0)$.

Fourier descriptors have several interesting properties:[1]

- *translation* of the line $L$ (by a quantity $Dx$ on the $x$-axis and $Dy$ on the $y$-axis) affects only the DC term $Z(0)$: $z_t(n) = z(n) + Dx + i \cdot Dy, \forall n = 0, 1, \ldots N-1$ $Z_t(0) = Z(0) + N \cdot (Dx + i \cdot Dy)$, $Z_t(n) =, Z(n), \forall n = 1, 2, \ldots, N-1$
- *rotation* of $L$ by an angle $\theta$ around the shape's center of the gravity results in translation of the transform factor phase, leaving the magnitude unaltered: $z_r(n) = z(n) \cdot e^{i\theta}, \quad \forall n = 0, 1, \ldots, N-1$ $Z_r(n) = Z(n) \cdot e^{i\theta} \rightarrow |Z_r(n)| = |Z(n)|, \forall n = 0, 1, \ldots, N-1$
- *scaling* of $L$ by a factor $s$ leads to scaling of the transform factors by the same factor: $z_s(n) = s \cdot z(n), \quad \forall n = 0, 1, \ldots, N-1$ $Z_s(n) = s \cdot Z(n), \forall n = 0, 1, \ldots, N-1$
- *changing the starting vertex* for traversing the closed polygonal line results only in a phase modulation: $z_{sp}(n) = z\,((n+k) \bmod N), \forall n = 0, 1, \ldots, N-1$ $Z_{sp}(n) = Z(n) \cdot e^{i2\pi n k/N} \rightarrow |Z_{sp}(n)| = |Z(n)|, \forall n = 0, 1, \ldots, N-1$
- *mirroring* $L$ causes mirroring of the coordinates of the Fourier descriptors' magnitude. The mirrored line

## Watermarking Basics

Numerous methods for watermarking audio, image, and video content exist in the literature. We classify these techniques using the embedding and detection domains. The first class contains spatial-domain techniques—that is, the watermark is embedded directly in the spatial image domain by properly modifying the image pixel data. In the second class, embedding occurs in a transform domain, and attempts to exploit domain properties mainly for watermark imperceptibility and robustness. Several papers have presented watermarking methods with watermark embedding in the discrete cosine transform, discrete Fourier transform, discrete wavelet transform, or fractal-based coding domain. Other papers provide overviews of existing multimedia watermarking methods.[1,2]

Many watermarking system models adopt principles from spread-spectrum communications.[3,4] For copyright protection purposes, blind watermarking is the most desirable method. Recent methods exploit human auditory system and human visual system principles to attain improved watermark imperceptibility. The *correlator* is the most widely used detector. It's optimal for signals modeled as additive white Gaussian noise. An important property of watermarking systems is their robustness against a variety of unintentional or malicious attacks, including low- and high-pass distortions, compression, noise corruption, and geometric distortions. Several papers overview possible attacks and the limitations of many current watermarking approaches under such attacks.[4-6]

Watermarking systems should also be imperceptible, not only to preserve data quality, but also for security, because anyone who can locate the watermark can easily weaken or remove it. Watermarks must be robust to distortions such as those caused by audio-, image-, or video-processing algorithms. In addition to modifying the digital media, these algorithms can modify the watermark. Thus, intentional or unintentional processing attacks can leave a watermark undetectable.

Generally, a watermark should be secure, meaning that only the owner or authorized user can localize or remove it. Also, statistical watermark removal should be impossible. Finally, watermark detection should produce low false-alarm and false-rejection probabilities.

### References

1. *Proc. IEEE*, special issue on identification and protection of multimedia information, vol. 87, no. 7, 1999.
2. *IEEE J. Selected Areas Comm.*, special issue on copyright and privacy protection, vol. 16, no. 4, May 1998.
3. I.J. Cox et al., "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, Dec. 1997, pp. 1673-1687.
4. F. Hartung, J.K. Su, and B. Girod, "Spread Spectrum Watermarking: Malicious Attacks and Counter-Attacks," *Proc. SPIE: Security and Watermarking of Multimedia Contents*, vol. 3657, SPIE, 1999, pp. 147-158.
5. S. Craver et al., "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications," *IEEE J. Selected Areas Comm.*, vol. 16, no. 4, May 1998, pp. 573-586.
6. I.J. Cox and J.P. Linnartz, "Some General Methods for Tampering with Watermarks," *IEEE J. Selected Areas Comm.*, vol. 16, no. 4, May 1998, pp. 587-593.

around the $x$-axis of $L$ equals the complex conjugate of $z(n)$:

$$z_m(n) = \bar{z}(n),\ \forall n = 0,1,\ldots,N-1,$$

$$Z_m(0) = \bar{Z}(0), Z_m(n) = \bar{Z}(N-n),\ \forall n = 1,2,\ldots,N-1 -$$

$$\left|Z_m(0)\right| = \left|Z(0)\right|,\ \left|Z_m(n)\right| = \left|Z(N-n)\right|,\ \forall n = 1,2,\ldots,N-1$$

where $\bar{z}$ is the complex conjugate of $z$. In this case, mirroring occurs on the $x$-axis. Mirroring around an arbitrary line is equivalent to mirroring, rotating, and translating around the $x$-axis. Rotation and translation don't affect the Fourier descriptors' magnitude, thus mirroring around an arbitrary line is equivalent to mirroring around the $x$-axis.

- *inversion of the traversal direction* of the line causes mirroring of the coordinates of the Fourier descriptors' magnitude:
  $z_i(n) = z(N-n),\ \forall n = 0, 1, \ldots, N-1$,
  $|Z_i(0)| = |Z(0)|,\ |Z_i(n)| = |Z(N-n)|,\ \forall n = 1, 2, \ldots, N-1$

These properties let the watermark remain detectable even after ordinary attacks on watermark integrity, such as translation, scaling, and other distortion attacks.

### Watermark construction and embedding

Using a pseudorandom generator, we create a bivalued ±1 random sequence and use it to produce a watermark. A bivalued ±1 random sequence has zero mean value and unit variance, and thus we construct the watermark as follows:

$$\mathbf{w}(i) = \begin{cases} 0, & \text{if } i < aN \text{ or } bN < i < (1-b)N \\ & \text{or } (1-a)N < i \\ \pm 1, & \text{if } aN < i < bN \text{ or } (1-b)N < i < (1-a)N \end{cases}$$

where $0 < a < b \leq 0.5$ and $N$ is the number of Fourier coefficients. Obviously, the smaller the $a$ and bigger the $b$, the more nonzero elements of vector $\mathbf{w}$.

The multiplicative embedding of the watermark on the magnitude of the polygonal line vertices' Fourier descriptors produces the watermarked polygonal line. Specifically, the magnitude of the Fourier descriptors of the watermarked polygonal line $Z'(k)$ is

$$|Z'(k)| = |Z(k)| + p|Z(k)|W(k) = |Z(k)|[1 + pW(k)]$$

where $k = 0, 1, \ldots, N-1$, and $p$ is a factor that determines the watermark power. Because the magnitude of the watermarked line's Fourier descriptors must be nonnegative, the multiplicative factor $p$ must be less than 1. We used multiplicative embedding because it corresponds to a simple watermark masking (the watermark amplitude is larger in large Fourier descriptor coefficients). Coefficients $a$ and $b$ control the low- and high-frequency ranges that the watermark affects. Because of the watermark's invariant properties, we embed the watermark only in the Fourier descriptor magnitude. We use the inverse Fourier transform of the Fourier coefficients to produce the watermarked curve $L'$. This watermarking method is also applicable in the control points of B-spline contour representations. In both polygonal curves and in B-spline curves, watermark embedding can create curve discontinuities or curve self-crossing.

### Watermark detection

Let $|Z'(k)|$ be the Fourier descriptor magnitude of the watermarked polygonal line. We use the correlation $c$ between $|Z'(k)|$ and the watermark $W$ to detect the watermark's presence:

$$c = \sum_{k=1}^{N} W(k)\left|Z'(k)\right| \tag{1}$$

If $W'$ watermarks the polygonal line $L$, and $W \neq W'$, the correlation is given by:

$$c = \sum_{k=1}^{N} W(k)\left|Z(k)\right| + p\left|Z(k)\right|W(k)W'(k)$$

If $W$ watermarks the polygonal line $L$, the correlation is

$$c = \sum_{k=1}^{N} W(k)\left|Z(k)\right| + p\left|Z(k)\right|W^2(k)$$

Assuming $W$ and $|Z(k)|$ are independent and identically distributed random variables, and $W$ has zero mean value, the mean value of $c$ is

$$\mu_c = \begin{cases} \dfrac{1}{2(b-a)N} \displaystyle\sum_{k \in A} \left|Z(k)\right| p & \text{if } W = W' \\ 0 & \text{if } W \neq W' \\ 0 & \text{if no watermark} \\ & \text{is present} \end{cases}$$

where $A = \{n \in N, aN \leq n \leq bN, (1-a)N \leq n \leq (1-b)N\}$.

We perform a detector output normalization in the range [0, 1] by dividing the correlator $c$ by its theoretical mean value. Thus, the normalized correlator $c'$ equals $c/\mu_c$. Because the watermarking method is blind, $Z(k)$ is unknown during detection. However, from the assumption that $W$ has zero mean value,

$$\mu_{|Z'(k)|} = \overline{Z(k)} + p\overline{W}(k)\overline{Z(k)} = \overline{Z(k)} = \mu_{|Z(k)|}$$

Thus, instead of $c$, we use a normalized correlator $c'$ in detection. The mean value of the normalized correlator $c'$ equals 1 if $W = W'$. The detection rule could have the form,

$H_0$: $W'$ watermarks $L'$ if $c \geq T$
$H_1$: $W'$ doesn't watermark $L'$ if $c < T$

Because $T$ is the detection threshold, we must estimate two error probabilities: the *false alarm* probability $P_{fa}$ (the probability of detecting a watermark in an unmarked line) and the *false rejection* probability $P_{fr}$ (the probability of not detecting the watermark in a watermarked line).

To estimate these error probabilities, we embed a watermark in a vector image and perform detection using the correct key (the key used in the embedding) and then an erroneous key. We do this for many keys. Finally, we have two sets of detector outputs: one for detection by the erroneous key (set A) and one for detection by the correct key (set B).

Assuming the independence of the summation parts of the detector shown in Equation 1, and using the central limit theorem, we derive that both detector output set distributions are Gaussian. Using the detection experiments, we find the empirical detector output pdfs. Given these pdfs, we construct the resulting receiver operating characteristic (ROC) curves, first calculating the intervals

$$P_{fa} = \int_{T}^{\infty} f_1(x)dx \text{ and } P_{fr} = \int_{\infty}^{T} f_2(x)dx$$

where $f_1(x)$ and $f_2(x)$ are the theoretical distributions of sets A and B. Each threshold value $T$ corresponds to a pair ($P_{fa}$, $P_{fr}$). The ROC curve consists of all the ($P_{fa}$, $P_{fr}$) pairs calculated for many values of $T$. $T$ values are usually between the mean values $f_1(x)$ and $f_2(x)$.

## Robustness against attacks

The proposed method is robust to many geometrical transformations as well as to combinations of them.

### *Translation*

As stated previously, translation affects only the DC term. Thus, in the embedding procedure, the watermark parameter $a$ must be positive ($a > 0$) to exclude the DC term from the embedding/detection procedure.

### *Rotation*

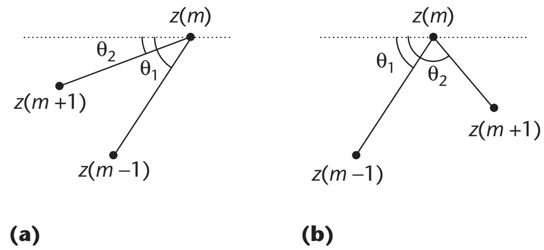
The proposed method is robust to rotation because rotation doesn't affect the magnitude of the line's Fourier descriptors. For the discrete case, if we want the polygonal line vertices' coordinates to be integers, we must also round off the coordinates of the rotated polygonal line. Experimental results show that this round-off doesn't decrease the method's performance because the magnitude of the rotated line's Fourier descriptors after round-off is slightly altered.

### *Scaling*

In a scaled polygonal line, the Fourier descriptor magnitude becomes $s|Z(k)|$. However, the normalized correlator remains invariant because both the numerator and the denominator are multiplied by $s$:

$$c_s = \sum W(k) \cdot |Z(k)| \cdot s + p \cdot s \cdot |Z(k)| \cdot W^2(k)$$

$$c'_s = \frac{\sum W(k) \cdot \left|Z(k)\right| \cdot s + p \cdot s \cdot \left|Z(k)\right| \cdot W^2(k)}{\frac{1}{2(b-a)N} \sum \left|Z(k)\right| \cdot s \cdot p} = \frac{c}{\mu_c} = c'$$




**2 Detection of the traversal direction: (a) counterclockwise and (b) clockwise.**

### *Change of traversal starting point*

If the starting point of the polygonal line traversal is the $s$th point of $z(n)$, the new polygonal line representation $z'(n)$ is given by $z'(n) = z((n + s) \, mod(N))$—that is, the curve's graphical representation remains exactly the same, but the signal $z(n)$ shifts circularly. However, the new line's Fourier descriptor magnitude stays the same because circular shifting doesn't affect it.

### *Traversal direction inversion*

In an inversion of traversal direction attack, the same vertex coordinates represent the polygonal line but the direction is inverted. Thus, the new line vertex coordinates are $[z(0), z(N-1), z(N-2), \ldots z(1)]$. The polygonal line also remains graphically unaltered after this attack. This type of attack creates a mirror $z(n)$ of the original signal. To cope with this distortion, we detect the watermark in both the original and the mirrored signals (in the frequency domain) or use a watermark signal $W$ that's symmetrical with respect to its DC term: $W(i) = W(N - 1 - i)$. An easier way to deal with this attack is to embed or detect the watermark in polygonal lines with specific traversal directions (clockwise or counterclockwise). The first step is to find the polygonal line's traversal direction, reverse it if it's not in the specific direction, and embed the watermark.

The same procedure works in watermark detection. The following algorithm finds the polygonal line's traversal direction:

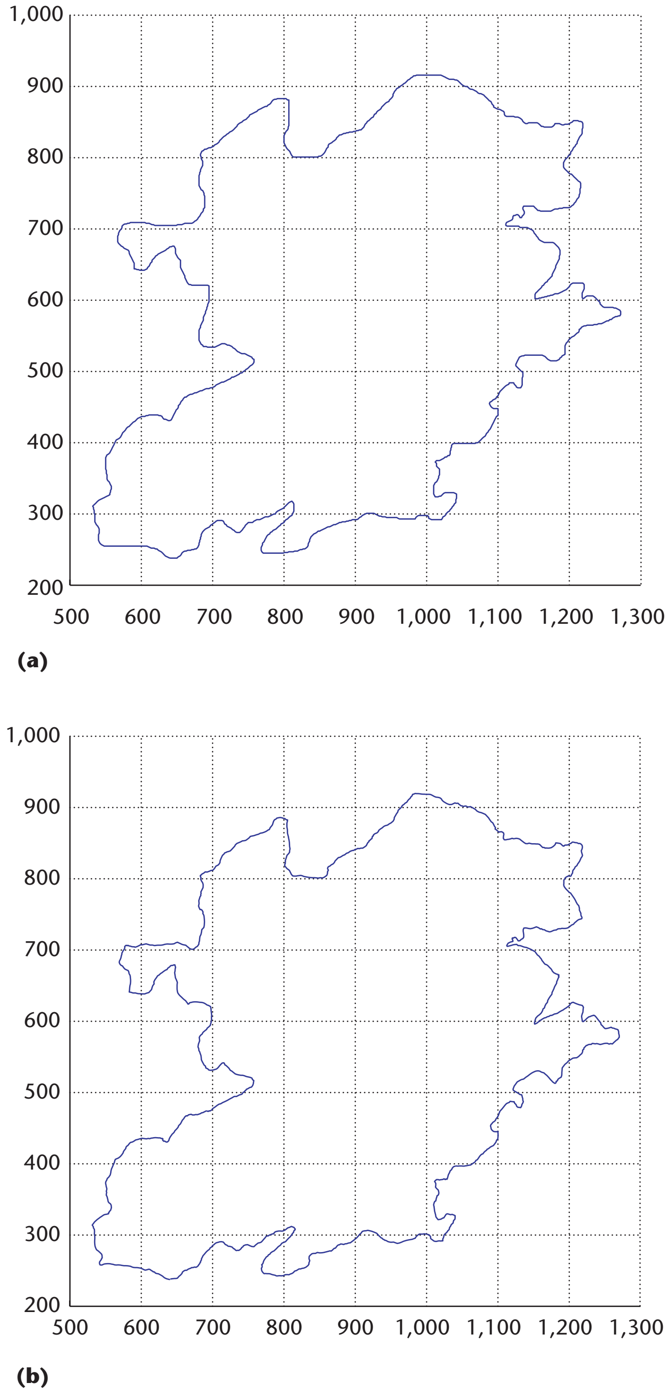
1. Find the vertex $z(m)$ with the largest $y$ coordinate. If more than one exists, pick one such that $y(m-1) < y(m)$.
2. From the values of $z(m-1)$, $z(m)$, and $z(m+1)$ calculate the angles $\theta_1$ and $\theta_2$ ($0 \le \theta_1, \theta_2 \le \pi$), as Figure 2 shows.
3. If $\theta_1 \ge \theta_2$, the traversal direction is counterclockwise; otherwise, it's clockwise.

### *Mirroring*

Mirroring of the polygonal line causes mirroring of the Fourier descriptor magnitude. Thus, the watermarking method is robust if the detector tries to find both the watermark and its mirrored version. An alternative is to produce symmetrical watermarks.

## Simulation results

We've tested our method in several polygonal lines. The polygonal line used in the results discussed here is




**3** **Map used in our experiments with the (a) original polygonal line and (b) watermarked line.**

a contour line from a map, shown in Figure 3a, with $N = 1{,}635$ vertices. We chose a polygonal line consisting of a small number of vertices to show the method's performance in small data sets. Results improve when $N$ increases.

To prove the method's robustness to the geometrical distortions we've discussed, we tested it for several keys. We selected the most widely used attacks for bitmap image watermarking, adapting them to vector images,[2] and used both individual and combined attacks. Figure 4 shows the ROC curves we constructed for the studied attacks. Subsequent figures show our simulation results. Each figure includes two pdfs showing the empirical distributions of the correlator detector output for 1,000 watermarked polygonal lines when we used

- erroneous watermarks—that is, different from the watermark used in embedding (left side), and
- correct watermarks—that is, the same watermark used in embedding (right side).

In well-separated pdfs, the watermarking method will have good detection performance, because a carefully chosen detection threshold will produce few detection errors.

Figure 5 presents the empirical distribution of the normalized correlator of 1,000 watermark detections with an erroneous key (left side) and 1,000 watermark detections with the correct key (right side). The corresponding ROC curve in Figure 4 (rotation 30°) shows that the proposed method is robust against rotation.

For the curve-translation attack, we translate the polygonal line by –100 pixels on the $x$-axis and 200 on the $y$-axis. The ROC curve in Figure 4 shows that the method's performance is robust against this attack.

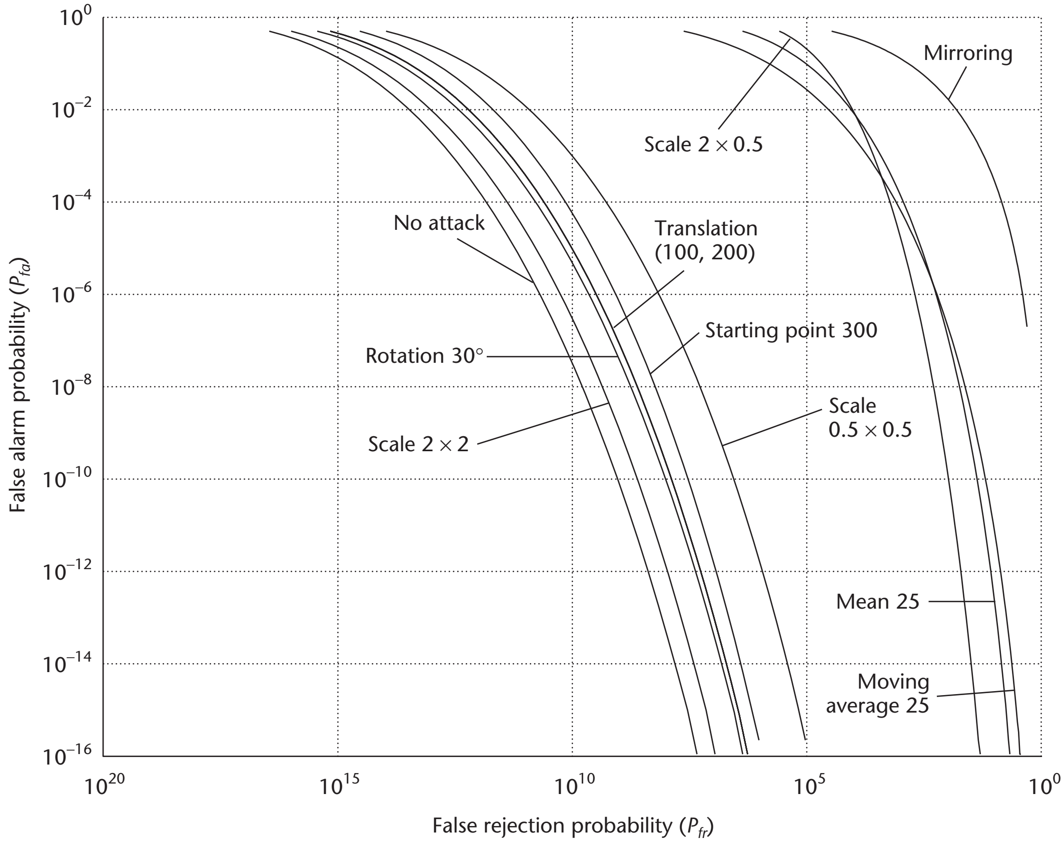
We also tested the proposed method against scaling. Figure 4 shows ROC curves for scaling by factors 2 and 0.5. In addition, we used anisotropic scaling as an attack. Figure 6 (on p. 50) shows a scaled polygonal line for scale factors 2 for the $x$-axis and 0.5 for the $y$-axis. The watermark in this case is also robust against the scaling attack, as Figure 4 shows.

Figure 4 also shows the ROC curves produced by the normalized correlator of 1,000 watermarked (with erroneous and correct keys) polygonal lines whose starting points shifted. In this experiment, the starting vertex is the 100th vertex of the original sequence. As the ROC curve shows, our method is also robust to this distortion.

The next experiment involved a horizontal flip of the watermarked contour. We used a symmetrical watermark of the form $W'(i) = W(N - i)$. Doing so halved the effective watermark length and reduced detection performance and can be seen in Figure 4 in the mirroring ROC curve. Alternatively, we could use nonsymmetrical watermarks and perform the detection twice: once using the watermark and once using the flipped version. In this case, the detection performance would be the same as without an attack, but at the expense of double computational load.

We also applied a contour-smoothing attack using a contour-filtering procedure. We implemented two kinds of filters—moving average and median—on the two independent contour vertex coordinates $x(n)$, $y(n)$, and performed several tests for various window sizes. Figures 7 and 8 (on p. 50) show our results. The effect of the filtering is quite visible, as Figures 7a and 8a show. The window size we used for both filters was 25. Our watermarking method performed well under these attacks, as Figures 7b and 8b show. Careful selection of parameters $a$ and $b$ to watermark the contour's low frequencies improves the method's robustness against contour smoothing.

We've also tested the method against additive Gaussian noise of standard deviation $\sigma = 0.5$. We added the

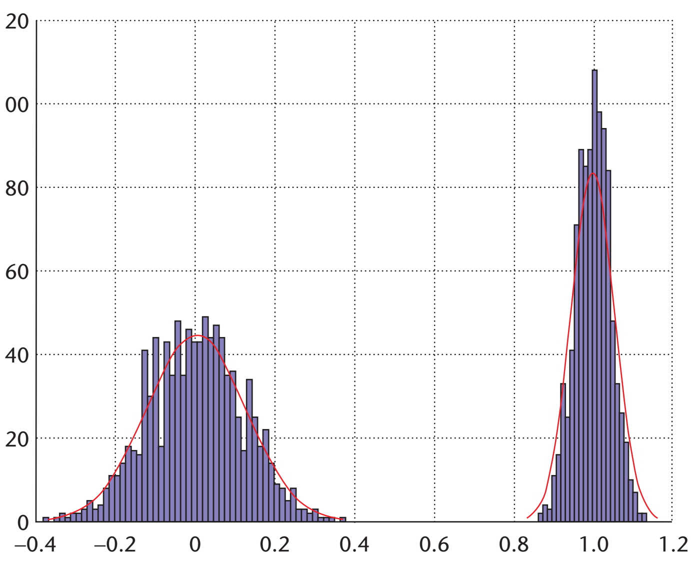



**4** **Receiver operating characteristic (ROC) curves for attacks performed on polygonal lines.**

noise in both $x$ and $y$ coordinates of the contour vertices. Figure 9a (on p. 51) shows the resulting attacked contour. The embedded watermark's frequency characteristics make the method robust against this attack, as the correlator distributions of lines watermarked with an erroneous key and attacked contours (left) and lines watermarked with the correct key and attacked contours (right) in Figure 9b show.

To validate performance against a combination of attacks, we performed the following test. First, we rotated the watermarked polygonal line 30 degrees. We then translated the coordinates (+200 on the $x$-axis and –500 on the $y$-axis). This made the 15th point the starting point of the polygonal line traversal, which means the new vector is of the form $((x_{15}, y_{15}), (x_{16}, y_{16}), \ldots, (x_{1,635}, y_{1,635}), (x_1, y_1), \ldots, (x_{14}, y_{14})))$. Next, we applied a moving-average filter to smooth the polygonal line. The filter we used had a window size of 11. Finally, we included additive Gaussian noise with a standard deviation of 0.5 in the polygonal line. The correlation output distributions in Figure 10a (on p. 51) show that the proposed watermarking method is efficient even against this complicated attack; Figure 10b shows the resulting polygonal line.

Finally, we tested the method for complex shapes. We converted a bitmap cartoon into vector images using Corel trace software. Almost no visible difference between the two existed.



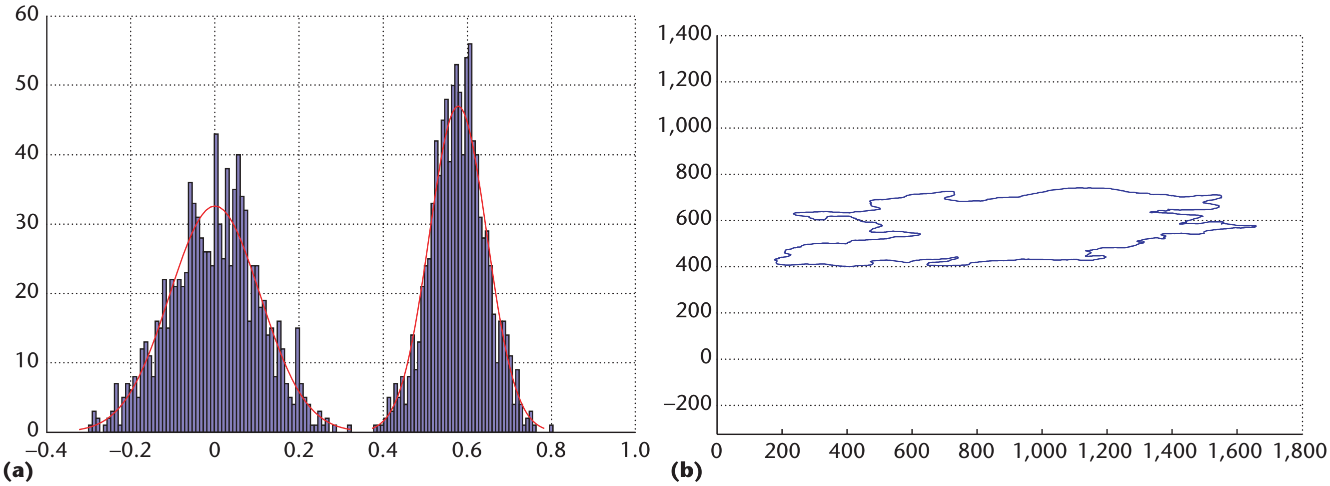

**5** **Empirical probability distribution of the normalized correlator output of 1,000 polygonal lines watermarked with an erroneous key (left) and the correct key (right).**

## Conclusion and future work

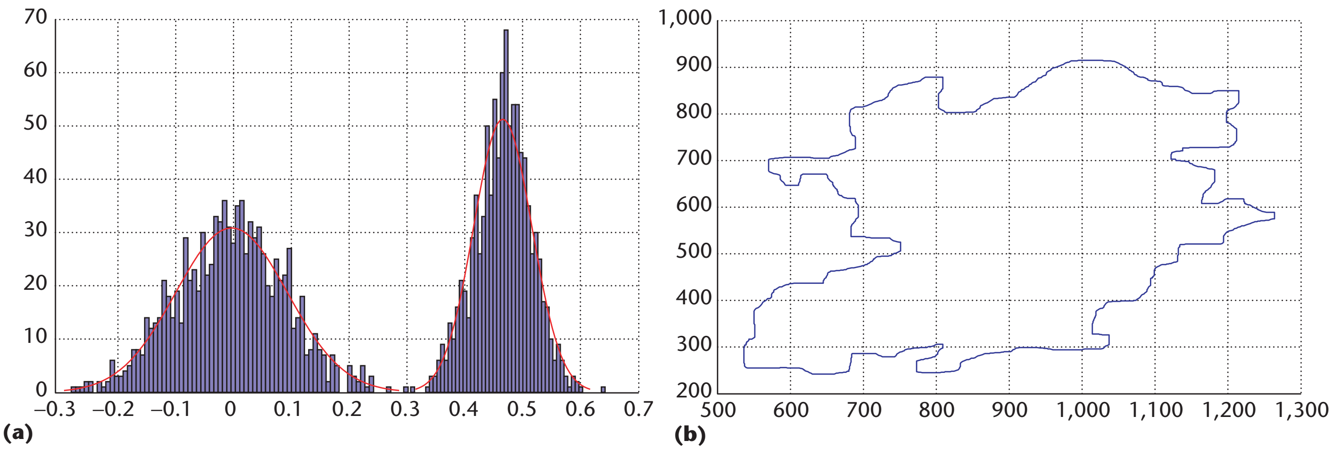
Despite our watermarking scheme's effectiveness, it's still not robust to polygonal line cropping and insertion or deletion of vertices. Research is under way to enhance the method in this direction. ■

### References

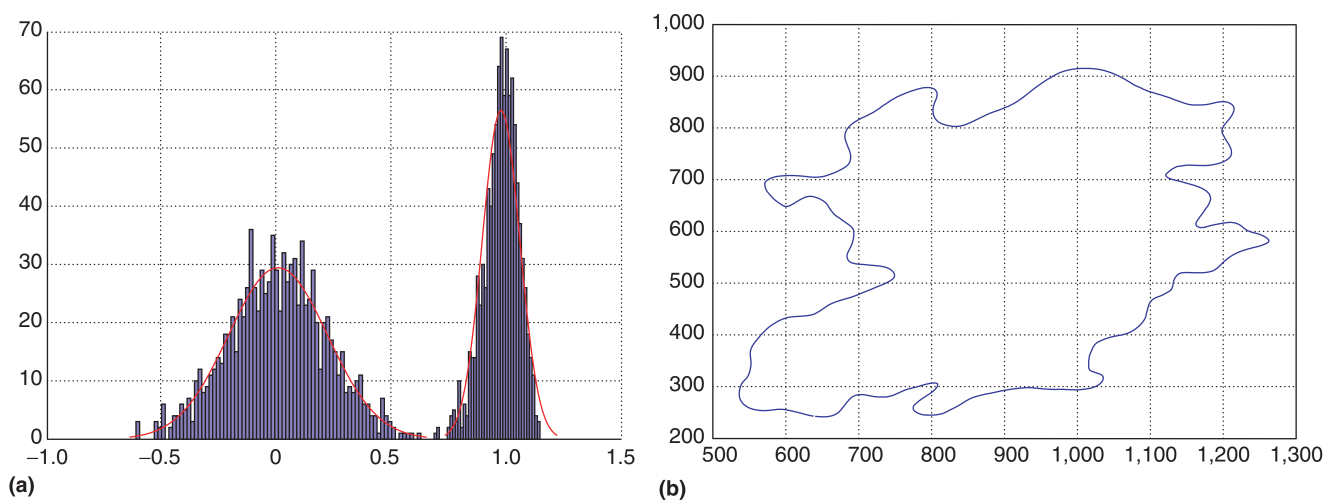

1. A.K. Jain, *Fundamentals of Digital Image Processing*, Pren-



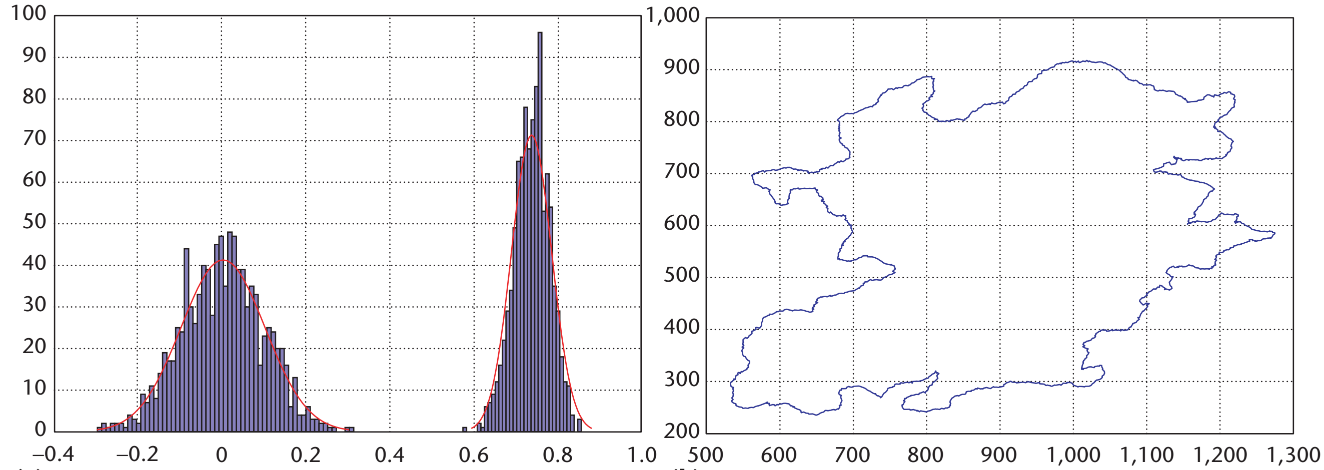

**6** **Results for scaling attack: (a) Empirical probability distribution of the normalized correlator of 1,000 scaled polygonal lines watermarked with an erroneous key (left) and the correct key (right), and (b) scaled polygonal line ($\times$ 2 (*x*-axis), $\times$ 0.5 (*y*-axis)).**




**7** **Results for median filtering: (a) Empirical probability distribution of the normalized correlator of 1,000 polygonal lines watermarked with an erroneous key (left) and with the correct key (right) after median filtering, and (b) polygonal line after median filtering (window size 25).**

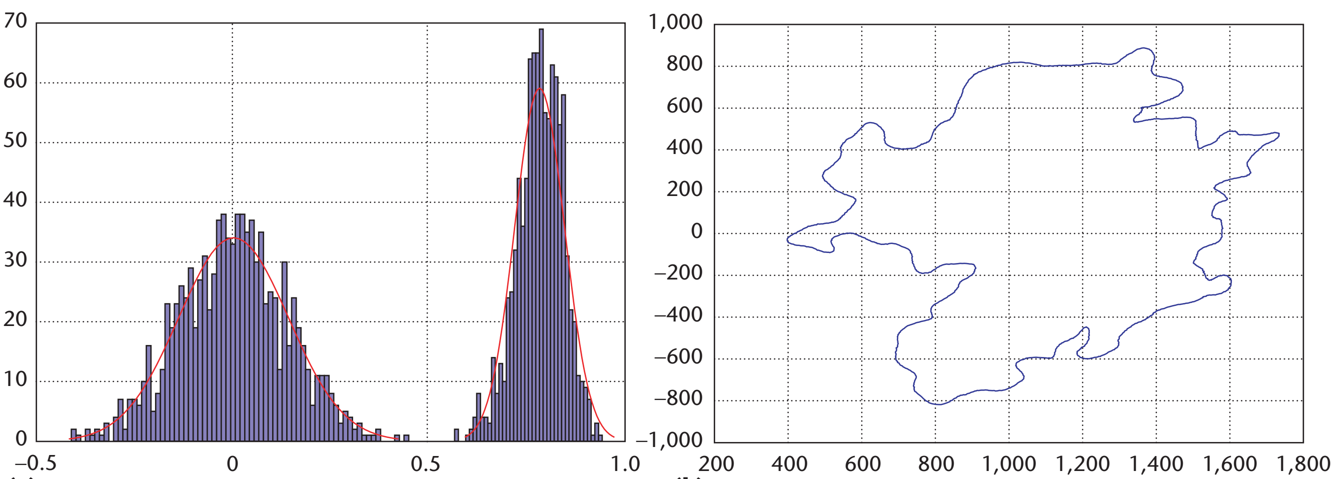



**8** **Results for moving-average filtering: (a) Empirical probability distribution of the normalized correlator of 1,000 polygonal lines watermarked with an erroneous key (left) and with the correct key (right) after moving-average filtering, and (b) polygonal line after moving-average filtering (window size 25).**




**(a)** **(b)**

**9 Results for Gaussian noise attack: (a) Empirical probability distribution of the normalized correlator of 1,000 polygonal lines watermarked with an erroneous key (left) and with the correct key (right) after additive Gaussian noise attack (s = 0.5), and (b) polygonal line after additive Gaussian noise attack (s = 0.5).**




**(a)** **(b)**

**10 Results for combination attack: (a) Empirical probability distribution of the normalized correlator of 1,000 polygonal lines watermarked with an erroneous key (left) and with the correct key (right) after attack combination, and (b) polygonal line after attack combination.**

tice Hall, 1989.

2. F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Attacks on Copyright Marking Systems," *Proc. 2nd Workshop Information Hiding Papers,* LNCS 1525, Springer-Verlag, 1998, pp. 218-238.

***Vassilios Solachidis*** *is a research assistant and graduate student in the Artificial Intelligence and Information Analysis group in the Department of Informatics at the Aristotle University of Thessaloniki, Greece. His research interests include image and signal processing and analysis and copyright protection of multimedia. Solachidis has a diploma in mathematics from the Aristotle University of Thessaloniki.*

***Ioannis Pitas*** *is a professor in the Department of Informatics at the University of Thessaloniki. His research interests include digital image processing, multidimensional signal processing, watermarking, and computer vision. Pitas has a diploma and a PhD in electrical engineering from the University of Thessaloniki. He is an associate editor of the* IEEE Transactions on Neural Networks.

*Readers may contact Vassilios Solachidis at the Dept. of Informatics, Univ. of Thessaloniki, Thessaloniki 54124, Greece;* vasilis@zeus.csd.auth.gr.