Watermarking polygonal lines using Fourier descriptors

V. Solachidis and I.Pitas

Department of Informatics, University of Thessaloniki

Thessaloniki 54124, Greece Tel,Fax: +30-2310996304

e-mail: {vasilis,pitas}@zeus.csd.auth.gr

Abstract

In this paper a blind watermarking method for vector graphics images is presented. Unlike the common watermarking methods, where the image luminance is altered, the coordinates of the polygonal curve vertices are modified in the presented method. The watermark is embedded in the Fourier descriptors of the vertices of the polygonal lines, resulting to slight alterations of the vertex coordinates of the polygonal line. Due to the Fourier descriptor properties, the watermark is detectable even after rotation, translation, scaling, additive Gaussian noise, smoothing, reflection attacks or combination of the above mentioned attacks.

Keywords: Watermarking, Vector graphics, Fourier descriptors

I. Introduction

Digital products can be easily copied, reproduced and sometimes maliciously processed in a network environment. Therefore, copyright protection of multimedia products using watermarking techniques has emerged as an important research topic over the last decade. A digital watermark is a digital signal carrying information about the copyright owner and is expected to be permanently embedded into the digital products, i.e. to be robust to any deliberate or other unintentional alteration of their content. Nevertheless the multimedia product should not suffer from a severe degradation of its quality, resulting from the watermark data embedding. In the following, we shall limit our presentation to the robust watermarking of vector images containing polygonal lines.

Watermarking has recently become a very active research area, a fact established by the publication of numerous methods dealing with the watermarking of audio, image or video content that have appeared in the literature to date. These methods can be distinguished in two classes, according to the embedding and detection domain. The first class contains the spatial-domain techniques. In these methods, the watermark is embedded directly in the spatial image domain by modifying properly the image pixel data. In the second class, embedding is performed in a transform domain, attempting to exploit the domain properties mainly for watermark imperceptibility and robustness. Several papers have presented watermarking methods with watermark embedding in the DCT, DFT, DWT, or fractal-based coding domain.Overview papers on existing multimedia watermarking methods may be found in [1], [2].

Many approaches adopt principles from spread spectrum communications in their watermarking system model [3], [4]. For copyright protection purposes, blind watermarking is the most desirable method. Recent methods exploit principles of the human auditory/visual system (HAS, HVS) to attain improved watermark imperceptibility. The most widely used detector is the correlator, which is optimal for signals modeled as Additive White Gaussian Noise (AWGN). An important property of watermarking systems is their robustness against a variety of unintentional or malicious attacks including lowpass/highpass distortions, compression, noise corruption, geometric distortions. An overview of possible attacks and the limitations of many current watermarking approaches under such attacks is given in [4]-[6]. Another important property of watermarking systems is imperceptibility. It is not demanded only for perceptual purposes (data quality reduction) but also for security reasons, because, if someone can locate the watermark, he can weaken or remove it in an easier way. The watermarks must be robust to distortions such as those caused by audio/image/video processing algorithms. These algorithms do not modify only the digital media but may also modify the watermark as well. Thus, the watermark may become undetectable after intentional or unintentional processing attacks. Generally, watermarks should be secure, meaning that only the owner or authorized user should be able to localize/remove the watermark. Also, statistical watermark removal should be impossible. Finally, watermark detection should produce low false alarm and false rejection probabilities.

Apart from bitmap images, another image type that is widely used in many applications is vector graphic images. Many applications, notably digital maps, geographic information system (GIS) data, 2D graphics and cartoon images are based on vector graphics data. The classical watermarking methods embed a watermark by applying luminance alterations to the image. These watermarking methods are not suitable for vector graphics images because, in that case, the watermark is perceivable and it can be easily removed. Vector graphics images usually consist of large homogeneous regions. Also, in most of the times, these homogeneous regions have uniform luminance value. Thus, even a small perturbation on the luminance values can be easily visible in these homogeneous regions. Furthermore, in vector graphics, all the information is located in the contours. Nonetheless, the classical watermarking methods embed the watermark in the homogeneous regions. Even if we could achieve minor visual changes on such images by embedding low power watermark, the watermark could easily be corrupted by a pirate. This could be achieved by applying such a malicious perturbation that would affect only the homogeneous region intensity (e.g. by quantization, or by filtering the homogeneous image regions). Thus, for these image types, contour (and not luminance) alteration is essential for an imperceptible and robust watermarking.

In this paper we propose an algorithm, that embeds watermarks to polygonal lines that describe image contours. These contours are usually described in vector format. The proposed watermarking method slightly modifies the vertex coordinates of the polygonal line. The watermark is embedded in the magnitude of the Fourier descriptors of the curve, to take into advantage its location, scale and rotation invariance properties. It should noted that the watermark detection is blind, i.e., the original polygonal line is not required in the watermark detection procedure.

The proposed technique has certain similarities to bitmap image watermarking in the DFT domain. However, it is essentially different since it can be applied to vector rather than to bitmap images. Furthermore, Fourier descriptor watermarking has different problems than DFT bitmap image watermarking. One such notable difference is that watermarks can cause curve discontinuity or curve self crossing. Such undesirable side-effects do not exist in DFT bitmap image watermarking.

In section II an overview of the Fourier descriptors and their properties are presented. The watermark generation and embedding is described in section III. In section IV the watermark detection algorithm is explained. The robustness of the presented method against geometric attacks in described in section V. Finally, experimental results and conclusions are presented in sections VI, VII respectively.

II. Fourier descriptors

Let L be a closed polygonal line that consists of N vertices. Let also $[x(n), y(n)]$, $n = 0, 1, ..., N-1$ be the Cartesian coordinates of each vertex. We can construct the following complex signal $z(n)$ by selecting randomly a vertex as a starting point and by tracing the vertices in an clockwise or anticlockwise way.

$$z(n) = x(n) + i \cdot y(n), \;\; n = 0, 1, ..., N-1.$$

as can be shown in Figure 1. In the following section, it will be explained that neither the knowledge of the starting point nor of the traversing way is required in the watermark embedding and detection procedures.

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

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

The inverse transform is the following:

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

The complex coefficients $Z(k)$ are called *Fourier descriptors* (FD) of the polygonal line.

In the above FD representation the $Z(k)$ coefficients around $Z(0)$ describe low frequency information, while the coefficients around $Z(N/2)$ describe high frequency information. Low frequency coefficients generally 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 ($Z(0) = \sum z(n)$). This is the reason why translations result only in translations of the DC term $Z(0)$.

Fourier descriptors have the following interesting properties [7]:

- **translation**:

$$z_t(n) = z(n) + Dx + i \cdot Dy, \;\; \forall\, n = 0, 1, ..., N-1 \tag{1}$$

$$Z_t(0) = Z(0) + N \cdot (Dx + i \cdot Dy), \quad Z_t(n) =, Z(n), \;\; \forall\, n = 1, 2, ..., N-1 \tag{2}$$

- **rotation**:

$$z_r(n) = z(n) \cdot e^{i\theta}, \;\; \forall\, n = 0, 1, ..., N-1$$

$$Z_r(n) = Z(n) \cdot e^{i\theta} \rightarrow |Z_r(n)| = |Z(n)|, \;\; \forall\, n = 0, 1, ..., N-1$$

- **scaling**:

$$z_s(n) = s \cdot z(n), \;\; \forall\, n = 0, 1, ..., N-1$$

$$Z_s(n) = s \cdot Z(n), \;\; \forall\, n = 0, 1, ..., N-1$$

- **change of the starting vertex**:

$$z_{sp}(n) = z((n+k)\; mod\; N), \;\; \forall\, n = 0, 1, ..., N-1$$

$$Z_{sp}(n) = Z(n) \cdot e^{i2\pi nk/N} \rightarrow |Z_{sp}(n)| = |Z(n)|, \;\; \forall\, n = 0, 1, ..., N-1$$

- **mirroring**:

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

$$Z_m(0) = \overline{Z}(0), \quad Z_m(n) = \overline{Z}(N-n), \;\; \forall\, n = 1, 2, ..., N-1 \rightarrow$$

$$|Z_m(0)| = |Z(0)|, \quad |Z_m(n)| = |Z(N-n)|, \;\; \forall\, n = 1, 2, ..., N-1$$

where $\overline{x}$ is the complex conjugate of x. In this case, the mirroring has been performed on x-axis. Mirroring around an arbitrary line is equivalent to mirroring around x-axis, rotation and translation. Rotation and translation do not affect the magnitude of the FD, thus mirroring around an arbitrary line is equivalent to mirroring around the x-axis.

- **Inversion of the traversal direction**:

$$z_i(n) = z(N-n), \forall\, n = 0, 1, ..., N-1,$$

$$|Z_i(0)| = |Z(0)|, \quad |Z_i(n)| = |Z(N-n)|, \;\; \forall\, n = 1, 2, ..., N-1$$

III. Watermark construction and embedding

A pseudorandom generator can be used to create a bi-valued ± 1 random sequence that can be used to create the watermark. This sequence has zero mean value and unit variance. The watermark is then constructed from this random sequence as follows:

$$W(i) = \begin{cases} 0, & \text{if} \quad i < a\,N \quad \text{or} \quad b\,N < i < (1-b)N \quad \text{or} \quad (1-a)N < i \\ \pm 1, & \text{if} \quad a\,N < i < b\,N \quad \text{or} \quad (1-b)N < i < (1-a)N \end{cases} \tag{3}$$

where $0 < a < b \leq 0.5$ and N is the number of the Fourier coefficients. It is obvious that the smaller the a is and the bigger the b is, the bigger is the number of the nonzero elements of vector W.

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

$$|Z'(k)| = |Z(k)| + p|Z(k)|W(k) = |Z(k)|[1 + pW(k)], \tag{4}$$

where $k = 0, 1, ..., N-1$ and p is a factor that determines the watermark power. Since the magnitude of the FDs of the watermarked line has to be non-negative, the multiplicative factor p has to be less than 1. The embedding has been done in a multiplicative way because this corresponds to a simple watermark masking (the watermark amplitude is larger in large FD coefficients). Coefficients a and b control the low and high frequency ranges, respectively, that are affected by the watermark. The watermark is embedded only in the magnitude of the Fourier descriptors due to its interesting invariance properties described in section II. The watermarked curve L' is produced by taking the inverse Fourier transform of the Fourier coefficients. This watermarking method can be also applied in the control points of B-spline contour representations. In both polygonal curves and in B-spline curves, watermark embedding may create curve discontinuities or curve self crossing.

IV. Watermark detection

Let $|Z'(k)|$ be the magnitude of the FD of the watermarked polygonal line. The correlation c between $|Z'(k)|$ and the watermark W can be used to detect the presence of the watermark:

$$c = \sum_{k=1}^{N} W(k)|Z'(k)| \tag{5}$$

If the polygonal line L is watermarked with W', $W \neq W'$, then the correlation is given by:

$$c = \sum_{k=1}^{N} W(k)|Z(k)| + p|Z(k)|W(k)W'(k) \tag{6}$$

If the polygonal line L is watermarked with W, then the correlation is:

$$c = \sum_{k=1}^{N} W(k)|Z(k)| + p|Z(k)|W^2(k) \tag{7}$$

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

$$\mu_c = \begin{cases} \frac{1}{2(b-a)N} \sum_{k \in A} |Z(k)| p & \text{if } W = W' \\ 0 & \text{if } W \neq W' \\ 0 & \text{if no watermark is present} \end{cases} \tag{8}$$

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

A detector output normalization in the range $[0, 1]$ can be performed by the division of the correlator c by its theoretical mean value. Thus, the normalized correlator c' equals c/μ_c. $Z(k)$ is unknown during detection, since the watermarking method is blind. However, from the assumption that W has zero mean value:

$$\mu_{|Z'(k)|} = \overline{Z(k)} + p\overline{W}(k)\overline{Z(k)} = \overline{Z(k)} = \mu_{|Z(k)|} \tag{9}$$

Thus, instead of c, a normalized correlator c' can be used in detection. The mean value of the normalized correlator c' equals 1 in the case $W = W'$. The detection rule could have the form:

H_0: L' is watermarked by W' if $c \geq T$

H_1: L' is not watermarked by W' if $c < T$.

Considering that T is the detection threshold, two error probabilities must be estimated, namely the false alarm probability P_{fa} (which is the probability to detect a watermark in an unmarked line) and the false rejection probability P_{fr}, i.e. the probability of not detecting the watermark in a watermarked line.

In order to estimate these error probabilities, we embed a watermark in a vector image and then we perform detection using firstly the correct key (the key that was used in the embedding) and then an erroneous key. This is performed for many different keys. Finally, we have two sets of detector outputs, one for the detection by the erroneous key (set A) and one for detection by the correct key (set B).

By assuming the independence of the summation parts of the detector shown in (5) and using the central limit theorem, it can be derived that both detector output sets distributions are Gaussian ones. The empirical detector output pdfs can be found from the above mentioned detection experiments. Then, given the estimated detector output pdfs, the resulting ROC (Receiver Operating Characteristic) curves are constructed. In order to construct the ROC the intervals below have to be calculated

$$P_{fa} = \int_{T}^{\infty} f_1(x)dx,$$

$$P_{fr} = \int_{\infty}^{T} f_2(x)dx.$$

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

V. Robustness against attacks

The proposed method is robust to geometrical transformations such translation, rotation, scaling, change of the starting traversal point, inversion of the traversal direction, reflection. The method is robust not only to each of these distortions but also to their combinations.

A. Translation

According to Section 2, translation affects only the DC term. Thus, in the embedding procedure the parameter a of the watermark has to be positive ($a > 0$), in order to exclude the DC term from the embedding/detection procedure.

B. Rotation

The proposed method is robust to rotation since the magnitude of the FD of the line is not affected by this operation. Although rotation operation introduces round off errors, the magnitude of the FD of the rotated line is almost unaltered.

C. Scaling

In a scaled polygonal line the magnitude of FD becomes $s|Z(k)|$. However, the normalized correlator c'_s remains invariant, because both the numerator and the denominator are multiplied by s.

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

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

D. Change of the starting point of the traversal of the polygonal line

If the starting point of the polygonal line traversal is set to the s-th point of $z(n)$, then the new polygonal line representation $z'(n)$ is given by:

$$z'(n) = z((n+s)\ mod(N))$$

i.e. the graphical representation of the curve remains exactly the same, but the signal $z(n)$ is circularly shifted. However, the magnitude of the FD of the new line is the same, because the magnitude is not affected by circular shifting.

E. Inversion of the traversal direction

The polygonal line also remains graphically unaltered after this attack. The signal $z(n)$ is mirrored. In order to cope with this distortion, we can detect the watermark both in the original and the mirrored signal (in the frequency domain), or we can use a watermark signal

W that is symmetrical with respect to its DC term: $W(i) = W(N-1-i)$. An easier way is to embed/detect the watermark in polygonal lines that have a specific traversal direction (clockwise or anticlockwise). Thus, the first step is to find the traversal direction of the polygonal line, reverse it, if it is not the specified and the embed the watermark. The same procedure is done in watermark detection. The following algorithm finds the traversal direction of the polygonal line.

- *Step 1* Find the vertex $z(m)$ that has the largest y coordinate. If there are more that one, pick one such that $y(m-1) < y(m)$.
- *Step 2* From the values of $z(m-1)$, $z(m)$ and $z(m+1)$ calculate the angles θ_1 and θ_2 ($0 \leq \theta_1, \theta_2 \leq \pi$), shown in Figure 3.
- *Step 3* If $\theta_1 \geq \theta_2$, then the traversal direction is anticlockwise, otherwise it is clockwise.

F. Mirroring

Mirroring of the polygonal line causes mirroring to the magnitude of the FD. Thus, the watermarking method is robust, if the detector tries detect both the watermark and its mirrored version. Alternatively, symmetrical watermarks should be produced.

VI. Simulation results

We have tested this method in several polygonal lines. The polygonal line that we used in the results shown, is a contour line from a map (Figure 4 a). Its number of vertices is $N = 1635$. We have chosen a polygonal line consisting of a small number of vertices in order to show the performance of this method in small data sets. Results always improve, when N increases.

In order to prove the robustness of this method to the geometrical distortions that were mentioned above, we have tested it for several keys. The selected attacks are the widely used ones for bitmap image watermarking, adapted to the vector image case [8]. We used both individual and combined attacks. The simulation results follow. In each of the Figures 6-

11, two pdfs are shown. The left hand side one is the empirical distribution of the correlator detector output for a polygonal line that has been watermarked with 1000 different keys where detection has been performed with an erroneous watermark (different than the one used in embedding). The right hand side one is the empirical distribution of the correlator detector output for 1000 watermarked polygonal lines where detection has been performed with the truly embedded (correct) watermark. If these pdfs are well separated, then the watermarking method will have good detection performance due to the fact that a carefully chosen detection threshold will produce few detection errors. For all the studied attacks, ROC curves have been constructed and are shown in Figure 5.

In Figure 6 the empirical distribution of the normalized correlator of 1000 watermark detections with an erroneous key (left-hand) and 1000 watermark detections with the correct key (right-hand) are presented. Then, the robustness against rotation attack is examined. The corresponding ROC curve of Figure 5, shows that the proposed method is robust against rotation.

The next studied attack is curve translation. The polygonal line is translated by -100 pixels on the x-axis and 200 on the y-axis. The ROC curve in Figure 5 shows that the method's performance is there good against this attack.

The proposed method was also tested against scaling. In Figure 5, ROC curves for scaling by factors 2 and 0.5 are shown. Furthermore, anisotropic scaling was used as an attack. A scaled polygonal line for scale factors 2 for x-axis and 0.5 for y axis is shown in Figure 7. In this case, the watermark is also robust to that scaling attack (Figure 5).

Also, the ROC curve produced by the the normalized correlator of 1000 watermarked with an erroneous key (left) and 1000 watermarked with the correct key (right) polygonal lines having shifted starting point is presented in Figure 5. In this experiment, the starting vertex is the $100th$ vertex of the original sequence. From the ROC curve is shown that the presented method is also robust to this distortion.

In the next experiment, horizontal flip of the watermarked contour has been preformed. A symmetrical watermark has been used of the form $W'(i) = W(N-i)$. As a result of the use of a symmetrical watermark, the effective watermark length is halved and the detection performance are not as good as the previous ones. The corresponding ROC curve is shown in Figure 5. Alternatively, we could use non-symmetrical watermarks and perform the detection twice: once by using the watermark and once by using its flipped version. In this case, the detection performance would be the same to that of the no attack case, at the expense of double computational load.

Another attack that has been applied is contour smoothing. In order to apply this attack, a contour filtering procedure is used. Two kinds of filters have been implemented, the moving average and the median filters on the two independent contour vertex coordinates $x(n)$, $y(n)$. Several tests have been performed for various window sizes. The results are shown in Figures 8 and 9. The effect of the filtering is quite visible as can be seen in Figures 8a and 9a. The window size that has been used for both filters is 25. The performance of the proposed method under these attacks is very satisfactory, as can been seen in Figures 8b and 9b. The performance depends on the selection of the parameters a and b. If a and b be carefully chosen to watermark the low frequencies of the contour, the method has improved robustness against contour smoothing.

The method has also been tested against additive Gaussian noise of standard deviation $\sigma = 0.5$. The noise was added both in x and y coordinates of the contour vertexes. The resulted attacked contour can be seen in Figure 10a. Due to the frequency characteristics of the embedded watermark, the method is also robust against this attack as the distributions of the correlator the watermarked (with an erroneous key) and attacked contours (left) and the watermarked (with the correct key) and attacked contours (right) shown in Figure 10b.

In order to validate the performance against a combination of attacks we performed the following test. First, the watermarked polygonal line was rotated by 30 degrees. Then, we have performed translation of the coordinates (+200 on the x and -500 on the y axis). Afterwards, the 15th point became the starting point of the of the traversal of the polygonal line, which means that the form of the new vector is $((x_{15}, y_{15}), (x_{16}, y_{16}), ..., (x_{1635}, y_{1635}), (x_1, y_1), ...(x_{14}, y_{14})))$. Then, a moving average filter has been applied resulting the smoothing of the polygonal line. The window size of the filter that was used is 11. Finally, additive Gaussian noise has been added in the polygonal line. The standard deviation of the noise equals 0.5. The resulting polygonal line is shown in Figure 11b. From the correlation output distributions shown in Figure 11a, it shown that the proposed watermarking method is very efficient even against this complicated attack.

Another interesting attack that could be applied is the conversion of the vector image to a bitmap image (e.g. by printing) and then the conversion back to vector image (by scanning and converting to vector image by using a appropriate software). This kind of attack would be faced if the bitmap-to-vector software and the watermarking method were both constructed in such a way that the watermarking method took advantage of the bitmap-to-vector algorithm. However, in our case, our aim is not the design of such an algorithm therefore we use a bitmap-to-vector software that has already been implemented.

The method was also tested for much more complex shapes. The bitmap cartoons shown in Figures 12a,13a, were converted in vector images using the Corel trace software. Then, the vector images was saved as a vector image file. The watermarked vector images are shown in Figures 12b,13b . There is almost no visible difference between the two vector images 12a,13a and 12b13b respectively.

VII. Conclusions

In this paper, a blind watermarking method for polygonal lines and contours is proposed. The watermark is embedded in the Fourier descriptors of the vertices of the polygonal lines. The watermark is robust against rotation, translation, scaling, smoothing and mirroring attacks.

This method can be also used in higher order B-spline contour watermarking.

References

[1] "Special issue on identification & protection of multimedia information," *Proceedings of the IEEE*, vol. 87, no. 7, 1999.

[2] "Special issue on copyright & privacy protection," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.

[3] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, December 1997.

[4] F. Hartung, J.K. Su, and B. Girod, "Spread spectrum watermarking: Malicious attacks and counter-attacks," in *Proc. of SPIE: Security and Watermarking of Multimedia Contents*, California, USA, January 1999, vol. 3657.

[5] S. Craver, N. Memon, B-L. Yeo, and M. Yeung, "Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks and implications," *IEEE Journal of Selected Areas in Communications*, vol. 16, no. 4, May 1998.

[6] I.J. Cox and J.P. Linnartz, "Some general methods for tampering with watermarks," *IEEE Journal of Selected Areas of Communications*, vol. 16, no. 4, May 1998.

[7] A.K. Jain, *Fundamentals of digital image processing*, Prentice Hall, 1989.

[8] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Attacks on copyright marking systems," *Second workshop on information hiding papers*.

List of Figures

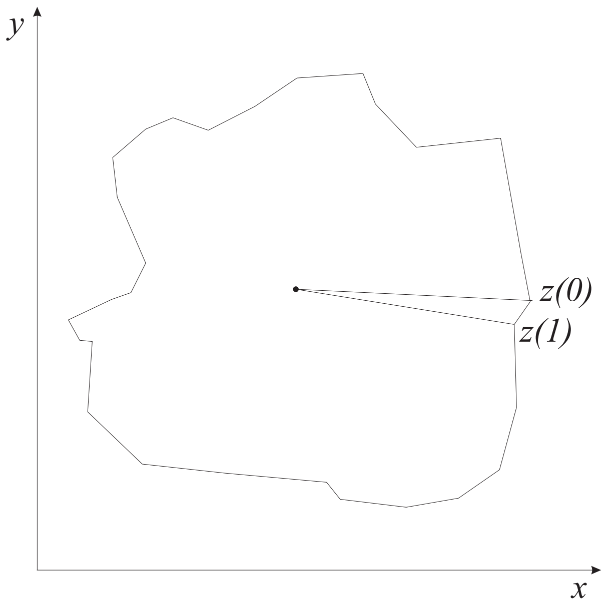


Fig. 1. Complex signal representation of a polygonal line.

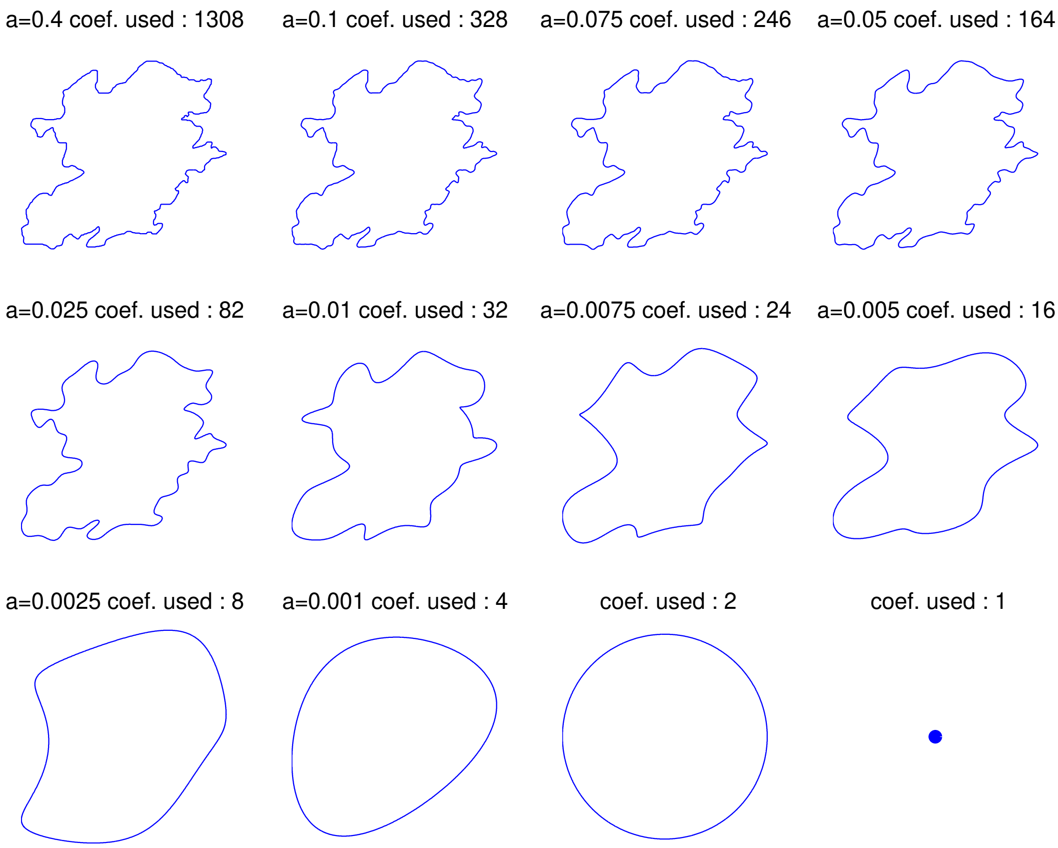


Fig. 2. Reconstructed contour using only $a \cdot N$ FD coefficients

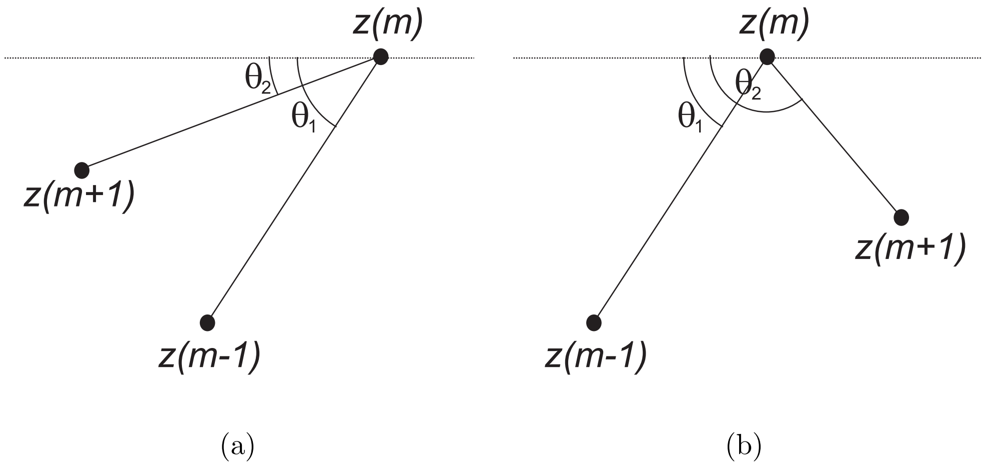


Fig. 3. Detection of the traversal direction: (a) anticlockwise (b) clockwise.

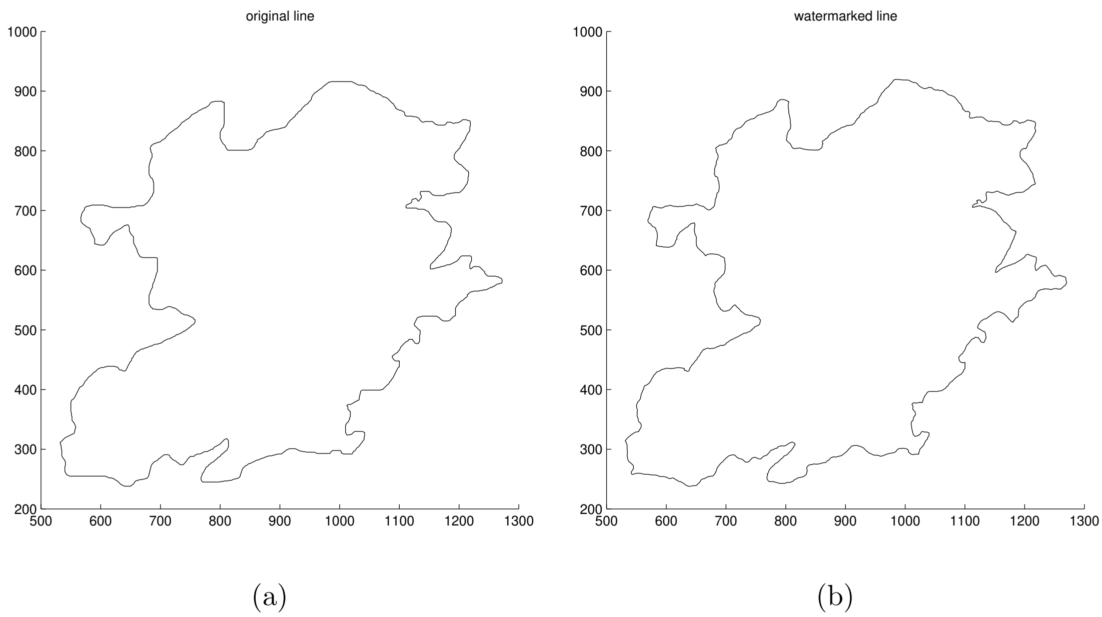


Fig. 4. (a) Original polygonal line (b) Watermarked line.

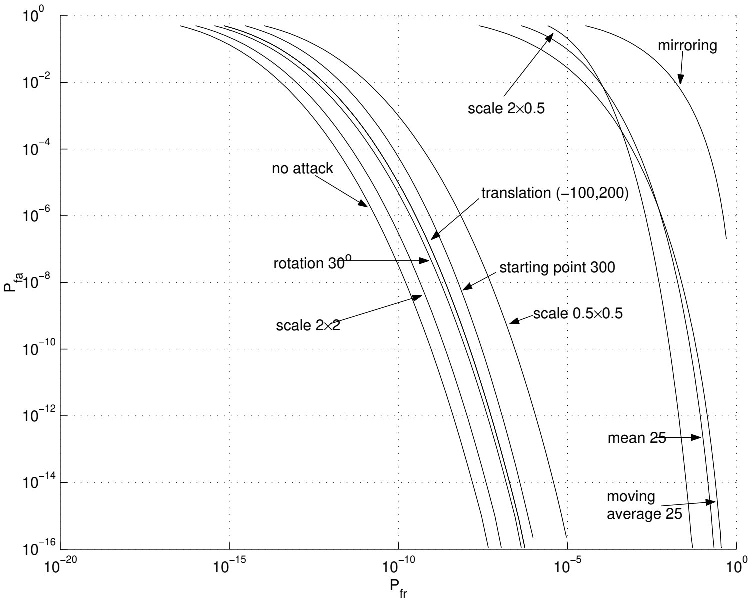


Fig. 5. ROC curves for all the attacks for polygonal line 4.

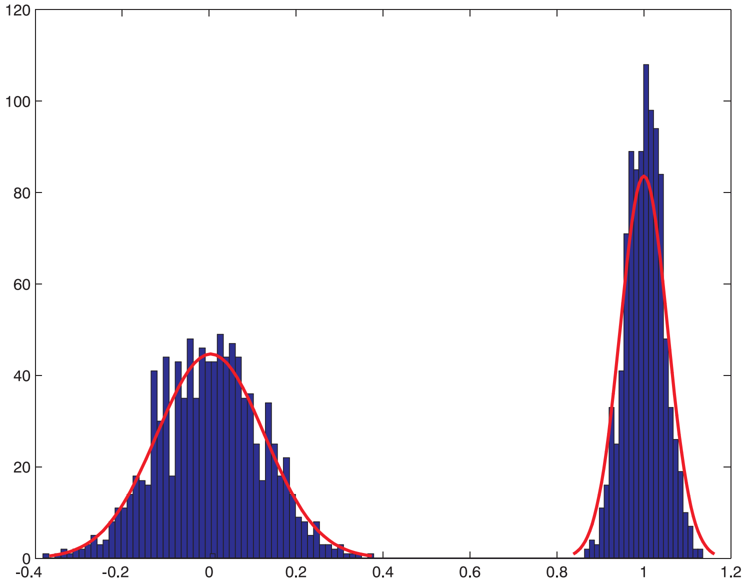
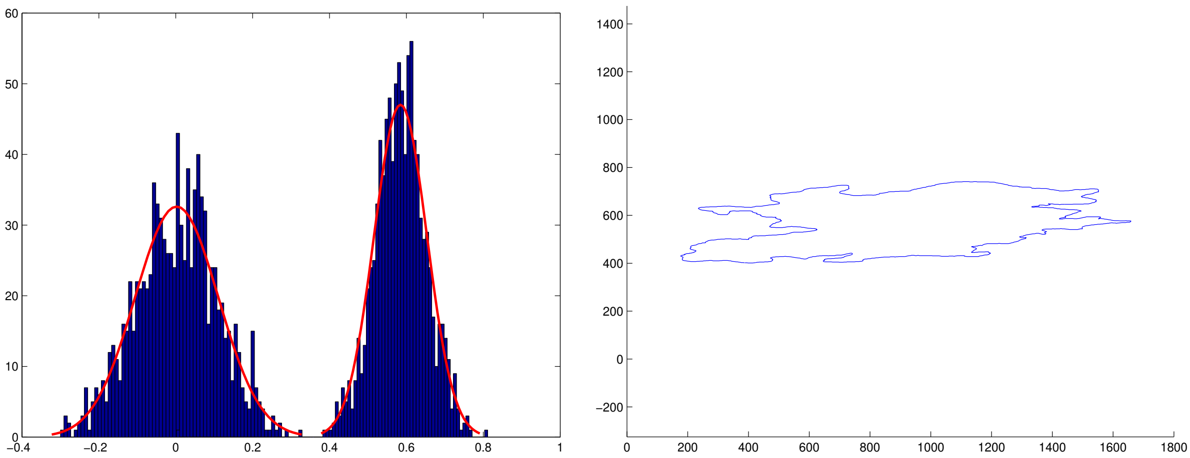
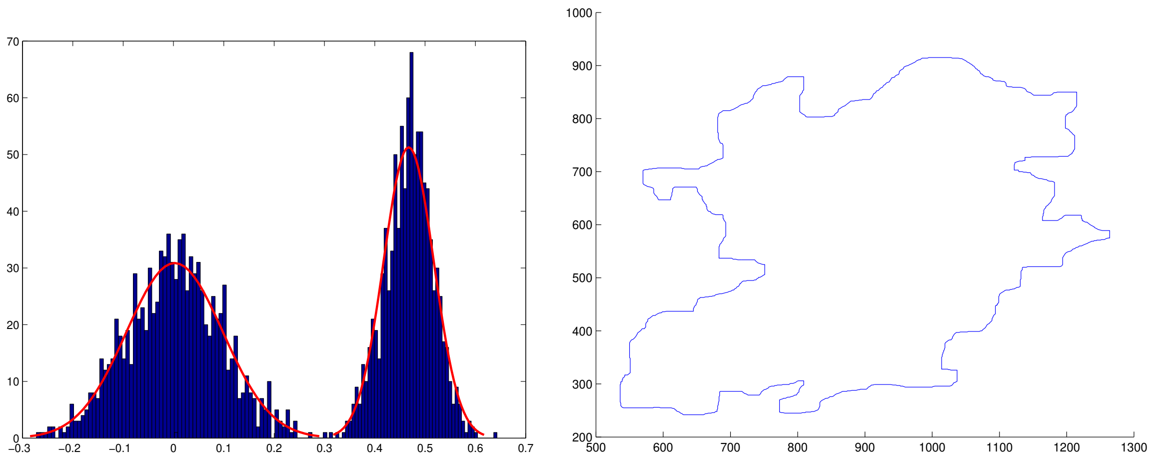


Fig. 6. Empirical probability distribution of the normalized correlator output of 1000 polygonal lines watermarked with an erroneous key (left) and with the correct key (right).



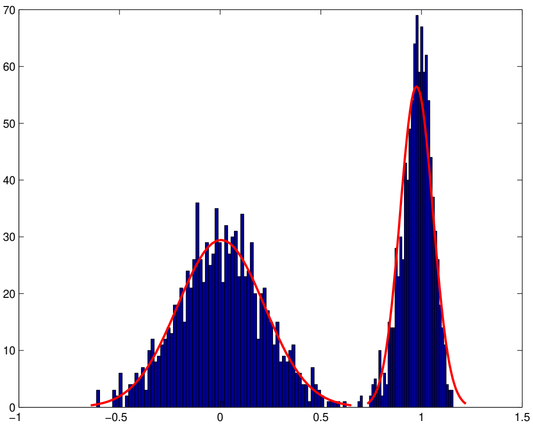
(a) (b)

Fig. 7. (a) Empirical probability distribution of the normalized correlator of 1000 watermarked with an erroneous key (left) and with the correct key (right) scaled polygonal lines (b) Scaled polygonal line ($\times 2$ $(x - axis)$, $\times 0.5$ $(y - axis)$,).

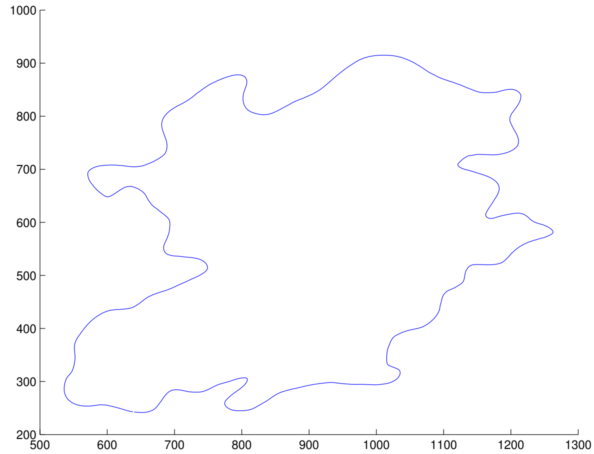


(a) (b)

Fig. 8. (a) Empirical probability distribution of the normalized correlator of 1000 watermarked with an erroneous key (left) and with the correct key (right) polygonal lines after median filtering (b) Polygonal line after median filtering (window size 25).

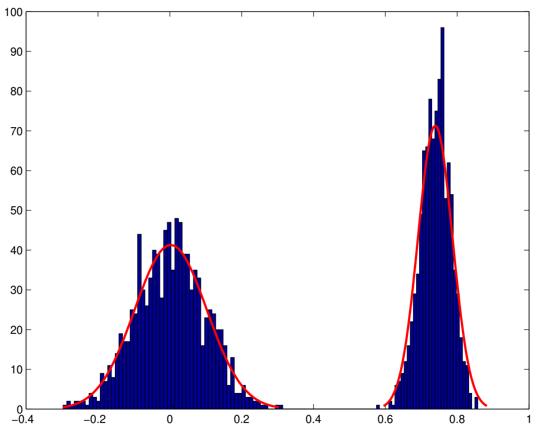


(a)

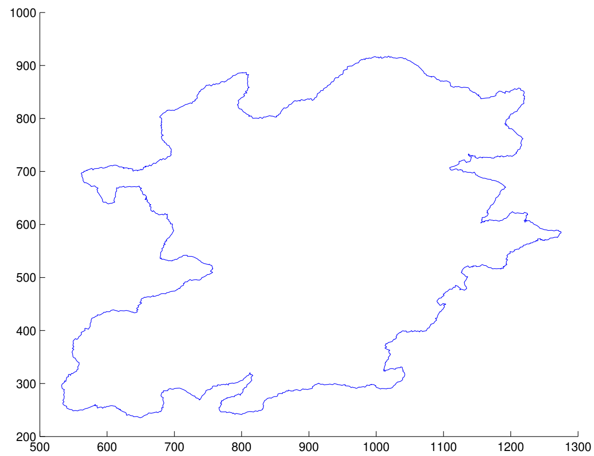


(b)

Fig. 9. (a) Empirical probability distribution of the normalized correlator of 1000 watermarked with an erroneous key (left) and with the correct key (right) polygonal lines after moving average filtering (window size 25) (b) Polygonal line after moving average filtering (window size 25).

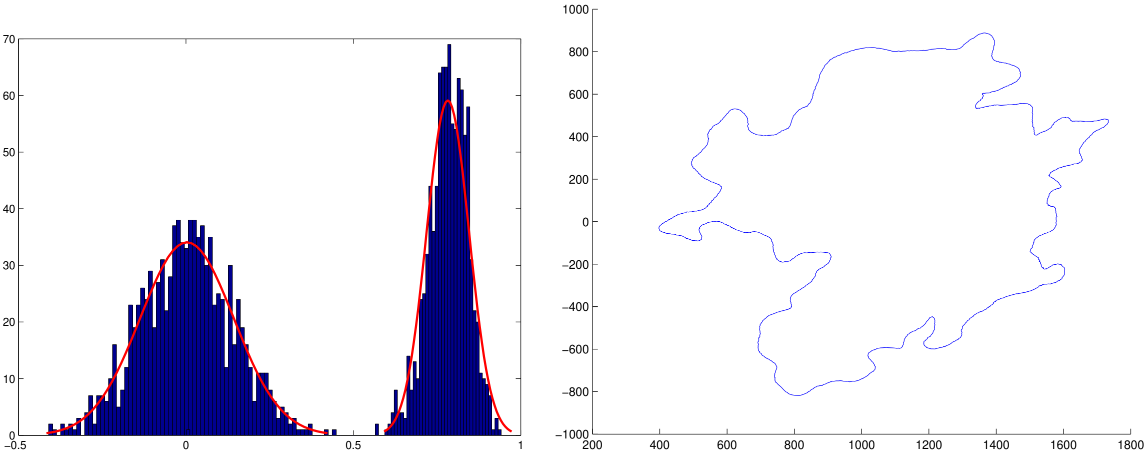


(a)



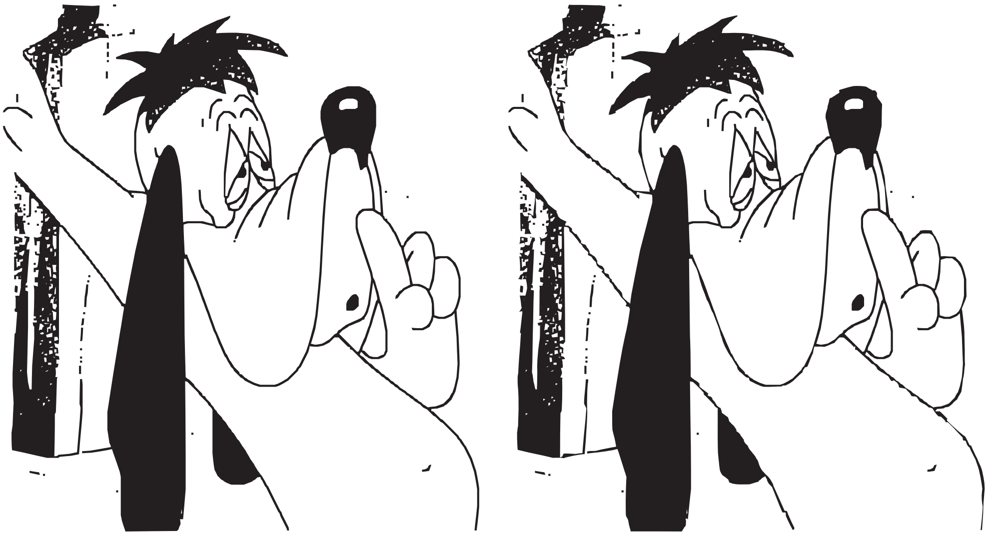
(b)

Fig. 10. (a) Empirical probability distribution of the normalized correlator of 1000 watermarked with an erroneous key (left) and with the correct key (right) polygonal lines after additive Gaussain noise attack ($\sigma = 0.5$) (b) Polygonal line after additive Gaussian noise attack ($\sigma = 0.5$).



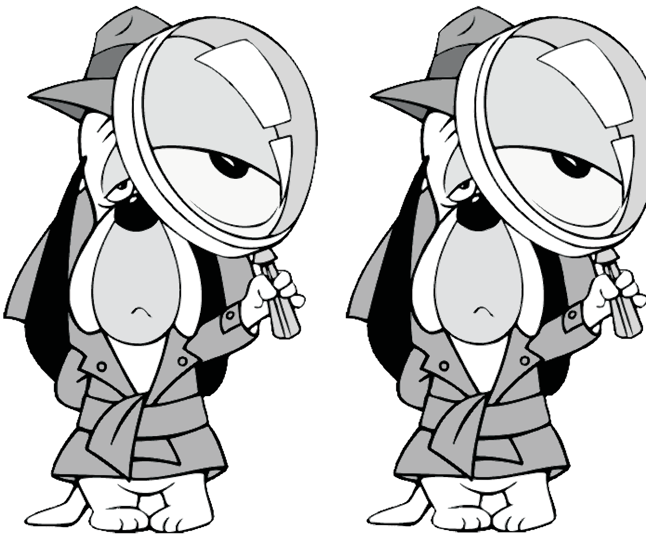
(a) (b)

Fig. 11. (a) Empirical probability distribution of the normalized correlator of 1000 watermarked with an erroneous key (left) and with the correct key (right) polygonal lines after attack combination (rotation 30 degrees, translation (+200,-500), change of starting point (15th), moving average filtering (window size 11), additive Gaussian noise ($\sigma^2 = 0.5$)) (b) Polygonal line after attack combination (rotation 30 degrees, translation (+200,-500), change of starting point (15th), moving average filtering (window size 11), additive Gaussian noise ($\sigma^2 = 0.5$)).



(a) (b)

Fig. 12. (a)Original cartoon (b)Watermarked cartoon.



(a) (b)

Fig. 13. (a)Original cartoon (b)Watermarked cartoon.