Rotation, scale, and translation invariant image watermark using higher order spectra

Hyung -Shin Kim
Yunju Baek
Heung-Kyu Lee

Division of Computer Science
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
373-1, Kusung-dong, Yusong-gu, Teajun
305-701, Korea
Phone: (+82) 42-869-3566 Fax : (+82) 42-869-3510
E-mail: hskim@casaturn.kaist.ac.kr

Abstract

Digital watermarks offer means of protecting copyright of digital multimedia. However, many of the proposed watermarking methods are vulnerable to the geometrical distortions which occur during normal use of the media. In this paper, we propose a new image watermarking method that is resilient to rotation, scaling, and translation (RST). We use the higher order spectra (HOS), in particular bispectrum feature vector of an image as the watermark. Bispectrum is the Fourier spectrum of the triple correlation of a signal. Phases of the integrated bispectra are invariant to translation and scaling. Rotation invariance is achieved using the Radon transform of the image. An image is decomposed into the 1-D projections and we construct a feature vector from them. A watermark is embedded by modifying the vector. We measure the distance between the feature vector extracted from the test image and the watermark at detector. Results of experimental studies show that our method is robust to geometric attacks, JPEG compression and Gaussian noise.

Subject terms: copy protection, watermark, image coding, feature extraction, higher order spectra, radon transform, geometrical attack robustness

1 Introduction

There has been a very intensive research in the digital watermarking area in the last few years[1]. A useful image watermarking method must be robust to the distortions occurred by any normal use of images. Those distortions include a wide range of image processing such as image enhancement, JPEG compression and geometrical modifications. However, conventional image watermarking algorithms are sensitive to geometric distortions[2]. Simple rotation, scale and translation (RST) may significantly reduce the detection level since it changes the alignment of the watermark. Random geometric distortion which is known as StirMark attack[2] greatly reduces the watermark strength at the detector.

Some watermarking methods that are resilient to geometrical attacks were reported in recent papers. One approach is to embed a known template into images along with the watermark[3,4]. The template contains the information of the geometric transform undergone by the image. During detection, the image is inverse transformed using the distortion information estimated from the template, then the watermark can be extracted. This method requires embedding a template in addition to the watermark so that this may reduce image fidelity and watermark capacity. Watermarks itself can be used as a template[5].

RST-invariant watermarks can be designed with the magnitude of the Fourier-Mellin transform of an image[6,7]. Though their watermarks were designed within the RST-invariant domain, they suffer severe implementation difficulty. It is mainly due to the computational complexity and the unstable log-polar mapping during the Fourier-Mellin transform.

Watermarking algorithms using a feature of an image were proposed as the second generation watermark[8,9]. As feature vectors of images are invariant to most of the image distortions, they were used as the keys to find embedding location.

In this paper, we propose a new watermarking algorithm using a RST invariant feature of images. However, we use the feature vector as a watermark not as a key. The vector is defined with higher order spectra (HOS) of the Radon transform of the image. For the use of HOS, we adopt the bispectrum (the third-order spectra) feature which is known to have invariance to geometrical distortions and signal processing[10].

Our algorithm is different from the previous methods using the Fourier-Mellin transform in that we use the Radon transform which has less aliasing during embedding than the log-polar mapping. Furthermore, we embed the watermark signal in the Fourier phase spectrum, and this makes our system more difficult to tamper than the Fourier-Mellin based methods where they use the Fourier magnitude spectrum[11]. We devise a

new insertion method that can avoid the usual interpolation errors during embedding procedure.

The proposed method is evaluated using the Stirmark 3.1[2] benchmark software and Corel image library[12]. The experimental results show that our algorithm performs well against the geometric distortions and other signal attacks.

The rest of this paper is organized as follows: Section 2 describes the bispectrum feature of images; Section 3 presents the watermarking algorithm; Section 4 shows the experimental results of the proposed method; Section 5 discusses results and in Section 6, we conclude with the contribution of our approach and directions for future development.

2. Theoretical Background

The bispectrum, $B(f_1, f_2)$, of a 1-D deterministic real-valued sequence is defined as

$$B(f_1, f_2) = X(f_1)X(f_2)X^*(f_1 + f_2) \quad (1)$$

where $X(f)$ is the discrete-time Fourier transform of the sequence $x(n)$ at the normalized frequency f. By virtue of its symmetry properties, the bispectra of a real

signal is uniquely defined in the triangular region of computation, $0 \le f_2 \le f_1 \le f_1 + f_2 \le 1$.

A 2-D image is decomposed into *N* 1-D sequences $g(s,\theta)$ using the Radon transform. The Radon transform $g(s,\theta)$ of a 2-D image *i*(*x,y*) is defined as its line integral along a line inclined at an angle θ from the y-axis and at a distance *s* from the origin. The projection slice theorem[13] states that the Fourier transform of the projection of an image on to a line is the 2-D Fourier transform of the image evaluated along a radial line. From the theorem, we can use 2-D Fourier transform instead of the Radon transform during implementation.

A parameter $p(\theta)$ is defined as the phase of the integrated bispectra of a 1-D Radon projection $g(s,\theta)$ along the line of $f_1 = f_2$ and it can be expressed with the polar mapped 2-D DFT , $I_p(f,\theta)$, as follows using the projection slice theorem:

$$\begin{aligned} p(\theta) &= \angle\left[\int_{f_1=0^+}^{0.5} B(f_1, f_1)df_1\right] \\ &= \angle\left[\int_{f_1=0^+}^{0.5} I_p^2(f,\theta)I_p^*(2f,\theta)df\right] \end{aligned} \tag{2}$$

Though the parameter can be defined along a radial line of slope a, $0 < a \le 1$ in the bifrequency space, we compute $p(\theta)$ at $a = 1$, where $f_1 = f_2$. In this way, we can

avoid interpolation during the computation of $p(\theta)$.

A vector $\mathbf{p}$ of length N is defined as $\mathbf{p} = (p(\theta_1), p(\theta_2), ...p(\theta_N))$. From the properties of the Radon transform and bispectrum parameter $p(\theta)$, $\mathbf{p}$ is invariant to dc-level shift, amplification, translation, scaling, and Gaussian noise[10]. As a rotation of the 2-D image results in a cyclic shift in the set of projections, $\mathbf{p}$ will be cyclically shifted as well. Figure 1 shows the feature vector $\mathbf{p}$ of the test images.

3. Algorithm

We use a modified feature vector of an image as the watermark. The watermark is embedded by selecting a vector from the set of extracted feature vectors. The chosen feature vector is used as the watermark and the inverted image is used as the watermarked image. The watermarks are generated through an iterative feature modification and verification procedure. This procedure avoids the interpolation errors that can occur during insertion and detection of the watermark. At detector, the feature vector is estimated from the test image. We use root-mean-square-error (RMSE) as our similarity measure instead of the traditional normalized correlation. It is because the feature vectors are not white and the correlation measure can not produce peak when they are same vectors. Hence, we measure the distance between the two vectors using

RMSE function. If the RMSE value is smaller than a threshold, the watermark is detected. The original image is not required at the detector. We define the detector first and an iterative embedder is designed using the detector.

3.1 Watermark Detection

Given a possibly corrupted image $i(x,y)$, the $N \times N$ 2-D DFT is computed with zero padding.

$$I(f_1, f_2) = DFT\{i(x,y)\} \tag{3}$$

$M \times N$ polar map $I_p(f,\theta)$ is created from $I(f_1, f_2)$ along N evenly spaced θ's in $\theta = 0..180^\circ$ and it is shown as

$$I_p(f,\theta) = I(f\cos\theta, f\sin\theta) \tag{4}$$

where

$$\begin{aligned} f &= \sqrt{f_1^2 + f_2^2} \\ \theta &= \arctan(f_2 / f_1) \end{aligned} \tag{5}$$

The frequency f is uniformly sampled along the radial direction and bilinear interpolation is used for simplicity. This is equivalent to the 1-D Fourier transform of the Radon transform. The $p(\theta)$ is computed along the columns of I_p by equation (2) to construct the feature vector $\mathbf{p}$ of length N. The similarity s, is defined with RMSE between the extracted vector $\mathbf{p}$ and the given watermark $\mathbf{w}$ as following,

$$s(\mathbf{p},\mathbf{w}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[p(\theta_i) - w_i\right]^2} \tag{6}$$

where N is the length of the feature vector. If s is smaller than the detection threshold T, the watermark is detected. Fig. 2 (a) shows the detection procedure.

3.2 Watermark Embedding

Let $i(x,y)$ be a $N \times N$ grayscale image. We compute the 2-D DFT $I(f_1, f_2)$ of the image. $M \times N$ polar map $I_p(f,\theta)$ is created from $I(f_1, f_2)$ along N evenly spaced θ's in $\theta = 0..180^\circ$ as (4) and (5). The feature vector $\mathbf{p} = (p(\theta_1), p(\theta_2), ...p(\theta_N))$ is computed as (2). The watermark signal is embedded by modifying k elements of the vector. From a pseudo-random number generator, the number of modifications, k and the insertion angles θ_w are determined to select projections for embedding at $\theta_w \in [0^\circ \cdots 180^\circ)$. If we shift all the phases of a column of $I_p(f,\theta_w)$ by δ, we have a modified component $p'(\theta_w)$ as follows:

$$\begin{aligned} p'(\theta_w) &= \angle\left[\int_{f_1=0^+}^{0.5} X(f_1)e^{j\delta}X(f_1)e^{j\delta}X^*(f_1+f_1)e^{-j\delta}df_1\right] \\ &= \angle\left[\int_{f_1=0^+}^{0.5} X(f_1)X(f_1)X^*(f_1+f_1)df_1\right] + \delta \\ &= p(\theta_w) + \delta \end{aligned} \tag{7}$$

After shifting the phases of the selected columns of $I_p(f,\theta)$, we inverse transform it to have the watermarked image i'.

However, we cannot extract the exact embedded signal at detector. As reported in the previous researches[6,7], algorithms that modify the Fourier coefficients in polar or log-polar domain suffer three problems. First, interpolation at embedder causes errors at detector. During the polar or log-polar mapping, an interpolation method should be involved because we are dealing with discrete image data. Though we choose more accurate interpolation function, there will be some errors as long as we are working with discrete images. Second, zero-padding at detector degrades the embedded signal further. By zero-padding, spectrum resolution is improved but interpolation error is increased. Third, the interrelation of the Fourier coefficients in the neighboring angles causes 'smearing' of the modified feature values. If we modify a single element $p(\theta)$, it affects other values nearby. In the Reference 7, the authors have provided approximation methods to reduce the effects of these errors. Instead of using a similar method, we approach this problem differently. After modifying some elements of the feature vector, the watermarked image which contains the implementation errors is produced by the inverse 2-D DFT. We apply the feature extractor from this watermarked image and use the extracted feature $\mathbf{p}^*$ as the embedded watermark instead of the

initially modified feature vector $\mathbf{p}'$. In this way, we can avoid the inversion errors.

However, to guarantee the uniqueness of the watermark and its perceptual invisibility after insertion, we need a validation procedure to use $\mathbf{p}^*$ as a watermark. We empirically measure the maximum noise level $r1$ resulted from geometric distortions with the detector response s as follows

$$r1 = \max\{s(\mathbf{p}_i, \mathbf{q}_i) \mid i = 0,...,M\} \tag{8}$$

where $\mathbf{p}_\mathbf{i}$ is the feature vector of an image and $\mathbf{q}_i$ is the feature vector of the image after RST distortions. We should adjust the embedding strength so that the distance between the two features from the unmarked and the marked image show a higher value than $r1$. However, embedding strength can not be higher than $r2$ which defines the minimum distance between features of the images

$$r2 = \min\{s(\mathbf{p}_i, \mathbf{p}_j) \mid i \neq j, \mathbf{and}\ i, j = 0,...M\} \tag{9}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are the feature vectors of different images.

We preset $r1$ and $r2$ values empirically. With varying the embedding

strength δ, k and θ_w, it is checked if $r1 < s(\mathbf{p},\mathbf{p}^*) < r2$. If this condition is satisfied and the embedded signal is unobtrusive, $\mathbf{p}^*$ is accepted as a watermark. We repeat this validation procedure until we get the right result. In this way, we can embed the watermark without exact inversion of the modified signal. Fig. 2 (b) shows the watermark embedding procedure.

4. Experimental Results

For valid watermark generation, $r1$ and $r2$ are determined empirically using unwatermarked images. The similarity s is measured between unmarked test images and the smallest s is chosen for $r2$. Fig. 3 shows the histogram of s and we have chosen $r2 = 20$. For the determination of $r1$, robustness of the defined feature vector is tested. We used the Stirmark[2] to generate attacked images. Similarity s is measured between the original image and attacked images. The largest s is chosen for $r1$. Fig. 4 (a), (b), (c), (d), (e), (f) shows the variation of feature vector after rotation, scale, cropping, random geometric attack, JPEG compression, and Gaussian noise, respectively. From the graphs in Fig. 4, we set $r1 = 4.5$.

Feature vectors are modified with $\delta = 5^\circ \sim 7^\circ$ at randomly selected angles. The number of insertion angles is randomly determined between 1 and 3. A threshold

$T = 4.5$ is used for the detection threshold. Watermarks are generated using the iterative procedure described in section 3.2. During the iteration, parameters are adjusted accordingly. Fig. 5 (a) shows the watermarked Lena image and Fig. 5 (b) shows the amplified difference between original and watermarked images. The watermarked image shows PSNR of $36dB$ and the embedded signal is invisible. During the watermark insertion, we maintained the PSNR of the watermarked images higher than $36\,dB$.

Two experiments are performed for the demonstration of the robustness. Using images of Lena, mandrill and fishingboat, the watermark detection ratio is measured as implemented by the Stirmark benchmark software. The second experiment is performed to estimate the false positive (P_{fp}) and negative probability (P_{fn}) with 100 images from the Corel image library[12].

Table 1 shows the watermark detection ratio using the Stirmark benchmark tool. The ratio of 1 means 100% detection success and 0 means the complete detection failure. Against scaling the watermark is successfully detected in 50% scaling down. For small angle rotations in the range -2° to 2°, the watermark is successfully detected without synchronization procedure. It shows our method outperforms the other commercially available algorithms against geometric attacks. We referred the results of other methods

from the Reference 3.

Robustness of the watermark against each attack is measured with 100 unmarked images and 100 marked images. We measure the empirical probability density function (pdf) of the computed s with histogram. Though we don't know the exact distribution of *s,* we approximate the empirical pdf of s to the Gaussian distribution to show rough estimation of the robustness. The false positive probability and false negative probability can be computed using the estimates of mean and variance. The resulted estimated errors are shown in Table 2. Random geometric attack performance is the worst with $P_{fp} = 7.89 \times 10^{-2}$ and $P_{fn} = 2.90 \times 10^{-3}$. It shows that our method performs well over the intended attacks. The similarity histograms and receiver operating characteristic (ROC) curves (P_{fp} versus P_{fn} for several thresholds) are produced for analysis.

In this section, five attacks are examined: rotation, scaling, random geometric distortion, compression and Gaussian noise.

4.1 Rotation

Fig. 6 shows the histogram of *s* and ROC curve. Though the rotation by large angle can be detected by cyclically shifting the extracted feature vector, the performance of

rotation by a large angle is poor due to the difficulty of interpolation in the Fourier phase spectrum. For this reason, we show the results of rotation by small angles. This problem is discussed in section 5.1. With $T = 4.5$, P_{fp} is 3.36×10^{-2} and P_{fn} is 2.30×10^{-3}. False negative probability shows better performance than false positive probability in this attack. This is because the pdf of the similarity between unmarked images and watermarks has relatively large variance that resulted into the larger false positive probability. As P_{fp} and P_{fn} show, our method is robust against rotation by small angle.

4.2 Scaling

The detection histogram was measured using 50% scaled down images and 200% scaled up images. As the histogram in Fig. 7 (a) shows, the watermarked images show strong resistance to scaling attack. Fig. 7 (b) shows that P_{fp} is 3.5×10^{-6} and P_{fn} is 2.21×10^{-6}. These values are relatively lower than other attacks and this means our method performs well with scaling attacks. Our method has strong robustness against scaling attack even after scaling down to 50%.

4.3 Random geometric distortion

This attack simulates the print-and-scanning process of images. It applies a minor geometric distortion by an unnoticeable random amount in stretching, shearing, and/or rotating an image[2]. In Fig. 8 (a), the histogram shows large variance in the similarity between watermark and unmarked image. As the result, P_{fp} is 7.89×10^{-2} and P_{fn} is 2.90×10^{-3}, which are relatively large compared with others. Not many previous methods survive this attack and our algorithm works well even with those numbers.

4.5 Compression

JPEG compression with Q=30 and 70 was applied after watermark embedding. With Q=30, the watermarked image fidelity is unacceptable. However, our method survives the harsh compression attack. Fig. 9 shows the histogram and ROC curve. P_{fp} is 2.8×10^{-3} and P_{fn} is 2.2×10^{-20}. The false negative probability is extremely low and this is because our feature vector is not affected by any high frequency noises. Our method has strong resilience to JPEG compression.

4.6 Gaussian noise

Gaussian noise was added to the watermarked image by convolving a 3x3 kernel as

follows

$$\mathbf{G} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{6}$$

The histogram of the similarity of unmarked and marked images is shown in Fig. 10. (a). The ROC curve is shown in Fig. 10 (b). From the curve, $P_{fp} = 6.64 \times 10^{-15}$ and $P_{fn} = 2.85 \times 10^{-3}$ are determined. These probabilities show that our method is robust against the Gaussian noise.

5. Discussions

5.1 Rotation invariance

As we use the rotation property of DFT for rotation invariance, we need to employ methods that can compensate the problems identified in the literature[7,14]. For algorithms that use the Fourier magnitude spectrum, zero-padding and windowing show the required rotation property. Zero-padding, centered padding or side padding show no difference. This is because the magnitude spectrum is invariant to the shift resulted by the zero-padding. Symmetric 2-D windowing removes the cross artifact in the frequency domain. Windows such as Hanning, Hamming and Kaiser reduce the effect of image boundary keeping the signal loss low.

For the methods using phase spectrum, such as our algorithm, the zero-padding and windowing are not as effective as with magnitude spectrum. It is because phase spectrum changes more rapidly than the magnitude spectrum in frequency domain. Direct interpolation in the frequency domain can solve this problem. The jinc function, which is the circular counterpart of sinc function in the 2-D spectrum domain, is preferable for the interpolation function.

5.2 Complexity Analysis

The embedding algorithm requires computation of two FFT's (one to embed a watermark signal, and one to find the actual watermark) and one IFFT (the second FFT does not need to be inversed). Two polar mappings and one inverse polar mapping are required. Computations for the determination of thresholds are not considered as they are one time operations. The extraction algorithm requires computation of one FFT and one polar mapping. The embedding algorithm takes 15-20 sec. while detection takes 5-7 sec. on a Pentium 1 GHz with the Mathworks' Matlab[15] implementation. The polar and inverse polar mapping consumes most of the computation time. The valid watermark embedding was achieved after one or two iteration most of the time.

5.3 Capacity

Current implementation works for zero-bit watermark. With simple modification, we can embed more bits. After the iterative procedure to generate the valid watermarks, we can construct a code book such as "*dirty-paper code*[16]". Information is assigned to each valid watermark code during embedding. At detector, the code book is implemented within detector. During detection, detector compares extracted feature with the vectors registered in the code book. When a measured similarity value reaches a previously determined threshold, it shows the assigned information from the code book.

5.4 Embedding without exact inversion

If an embedding function does not have exact inversion function, the resulting watermarked image will be distorted. This distortion reduces the image fidelity and watermark signal strength. As argued in Reference 7, having exact inversion is not the necessary condition for the embedding function. Two approaches can be considered. One method is defining a set of invertible vectors and work only with those vectors during embedding procedure. Though the embedding space is reduced, exact inversion is possible. Another approach is to use a conversion function that maps the embedded watermark and the extracted vector. Our approach belongs to this category. At detector,

after estimation of watermark, this signal is mapped into the inserted watermark using the conversion function.

6. Conclusion

We propose a new RST invariant watermarking method based on an invariant feature of the image. A bispectrum feature vector is used as the watermark and this watermark has a strong resilience on RST attacks. This approach shows a potential in using a feature vector as a watermark. An iterative embedding procedure is designed to overcome the problem of inverting watermarked image. This method can be generalized for other embedding functions that do not have exact inverse function.

In the experiments, we have shown the comparative Stirmark benchmark performance and the empirical probability density functions with histograms and the ROC curves. Experimental results show that our scheme is robust against wide range of attacks including rotation, scaling, JPEG compression, random geometric distortion and Gaussian noise. The use of the bispectrum feature as an index for efficient watermarked image database search may offer new application possibility. Various embedding techniques and capacity issues for the generic feature-based watermark system would be our next research topic.

Acknowledgements

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc). The authors would like to thank K. M. Park of Korea Advanced Institute of Science and Technology (KAIST) for helpful discussions on higher order spectra.

References

1. F. Hartung and M. Kutter, "Multimedia watermarking techniques," in *Proc. IEEE*, 87(7) 1079-1107 (1999)

2. F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Proc. 2nd Int. Workshop on Information Hiding,* 218-238 (1998)

3. S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. Image Processing,* 9(6) 1123-1129 (2000)

4. G. Csurka, F. Deguillaume, J. J. K. O'Ruanaidh, and T. Pun, "A Bayesian approach to affine transformation resistant image and video watermarking," in *Proc. 3rd Int. Workshop on Information Hiding,* 315-330 (1999)

5. M. Kutter, "Watermarking resisting to translation, rotation, and scaling", *Proc. SPIE Multimedia Systems Applications*, 3528 423-431 (1998)

6. J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale, and translation invariant spread spectrum digital image watermarking," *Signal Processing,* 66 303-317 (1998)

7. C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Processing,* 10(5) 767-782 (2001)

8. M. Kutter, S. K. Bhattacharjee, and T. Ebrahimi, "Towards second generation

watermarking schemes," in *Proc. IEEE Int. Conf. Image Processing,* 320-323 (1999)

9. S. Guoxiang and W. Weiwei, "Image-feature based second generation watermarking in wavelet domain," in *Lecture Notes in Computer Science,* 2251 16-21 (2001)

10. V. Chandran, B. Carswell, B. Boashash, and S. Elgar, "Pattern recognition using invariants defined from higher order spectra: 2-D image inputs," *IEEE Trans. Image Processing,* 6(5) 703-712 (1997)

11. J. O Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *Proc. IEEE Int. Conf. Image Processing,* 239-242 (1996)

12. *Corel Corporation, Corel Stock Photo Library 3*

13. A. K. Jain, "Image reconstruction from projections," Chap. 10 in *Fundamentals of Digital Image Processing*, Prentice Hall 431-475 Englewood Cliffs, NJ (1989)

14. J. Altmann, "On the digital implementation of the rotation-invariant Fourier-Mellin transform," *J. Inform. Process. Cybern.*, EIK 28 (1) 13-36 (1987)

15. MATLAB, The MathWorks, Inc.

16. M. L. Miller, "Watermarking with dirty-paper codes," in *Proc. IEEE Int. Conf. Image Processing,* 538-541 (1999)

Biographies

Hyung-Shin Kim received his B.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), in 1990 and M.S. degree in satellite communication engineering from University of Surrey, U.K. in 1990. From 1992 to 2001, He was with Satellite Technology Research Center (SaTReC), KAIST as a senior researcher. At SaTReC, he worked on real-time operating system and on-board computer system. From 1998 to 2000, he was the project manager for the development of a small satellite, KAISTSAT-1. Since joining the real-time computing laboratory at KAIST in 1994, his research shifted to problems in multimedia signal processing and digital contents right management. He is currently pursuing Ph.D. degree in computer science at KAIST. His research interests include digital watermarking, multimedia signal processing, and digital contents right management.

Yunju Baek received B.S., M.S. and Ph.D. degree on Computer Science from the Korea Advanced Institute of Science and Technology (KAIST) in 1990, 1992, 1997, respectively. From 1999 to 2002 he served as a CTO of NHN corp., a major internet portal company in Korea. He is currently a research professor of KAIST. His research interests include digital watermarking, multimedia information retrieval, internet multimedia application and e-business technology.

Heung-Kyu Lee received the BS degree in electronics engineering from the Seoul National University, Seoul, Korea, in 1978, and M.S., Ph. D. Degrees in computer science from the Korea Advanced Institute of Science and Technology(KAIST), in 1981, and 1984, respectively. From 1984 to 1985 he served as a post-doc at the University of Michigan, Ann Arbor. Since 1986 he has been a professor in the Department of Computer Science, the Korea Advanced Institute of Science and Technology, Taejon, Korea 305-701(hklee@casaturn.kaist.ac.kr). He is now a director of the DRM-forum for the digital right management. His major interests are real-time processing, image processing, and digital watermarking.

Table Captions

Table 1. Stirmark test results compared to commercial algorithms

Table 2. False positive probability and false negative probability for various distortions of the 100 watermarked image

Table 1.

	Proposed approach	Digimarc	Suresign
Scaling	**1.0**	**0.72**	**0.95**
Rotation small angle and cropping	**0.95**	**0.94**	**0.5**
Random geometric distortion	**0.93**	**0.33**	**0**
JPEG compression	**1.0**	**0.81**	**0.95**

Table 2.

Distortion	False positive probability	False negative probability
Rotation	3.36×10^{-2}	2.3×10^{-3}
Scaling	3.5×10^{-6}	2.21×10^{-6}
Random geometric attack	7.89×10^{-2}	2.90×10^{-3}
Compression	2.8×10^{-3}	2.2×10^{-20}
Gaussian noise	6.64×10^{-15}	2.85×10^{-3}

Figure Captions

Fig. 1 Feature **p** of sample images, Lena, Fishingboat, and Pentagon

Fig. 2 Proposed watermarking scheme

(a) Detection procedure

(b) Insertion procedure

Fig. 3 Histogram of similarity s between 100 unmarked images

Fig. 4 Histogram of similarity s between the original and the attacked unmarked image

(a) Rotation ($\pm 0.25°$, $\pm 0.50°$, $\pm 0.75°$, $\pm 1.0°$, $\pm 2.0°$)

(b) Scaling (x0.5, x0.75, x0.95, x1.1, x1.5, x2.0)

(c) Cropping (1%, 2%, 5%)

(d) Random geometric attack

(e) JPEG Compression (Q=10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90)

(f) Gaussian noise (3x3 filter)

Fig. 5 Watermark embedding example

(a) Watermarked Lena image at $r = 35^{\circ}$and 125°

(b) Amplified difference between watermarked and original images

Fig. 6 Histogram and ROC curve of unmarked and marked image after rotation (±0.25°, ±0.50°)

(a) Histogram of *s* (+ : Marked image, ▫ : Unmarked image)

(b) ROC Curve

Fig. 7 Histogram and ROC curve of unmarked and marked image after scaling (x0.5, x2.0)

(a) Histogram of *s* (+ : Marked image, ▫ : Unmarked image)

(b) ROC Curve

Fig. 8 Histogram and ROC curve of unmarked and marked image after random geometric attack

(a) Histogram of *s* (+ : Marked image, ▫ : Unmarked image)

(b) ROC Curve

Fig. 9 Histogram and ROC curve of unmarked and marked image after JPEG compression (Q=30, 70)

(a) Histogram of *s* (+ : Marked image, ▫ : Unmarked image)

(b) ROC Curve

Fig. 10 Histogram and ROC curve of unmarked and marked image after Gaussian noise

(c) Histogram of s (+ : Marked image, ▫ : Unmarked image)

(d) ROC Curve

Figure 1.

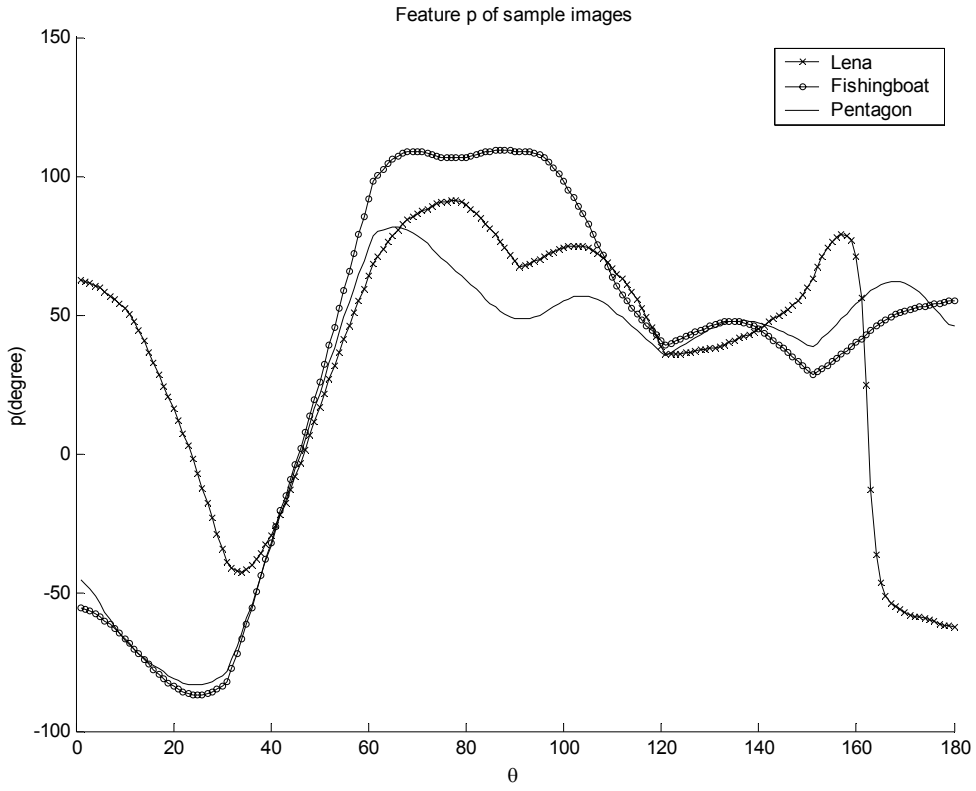
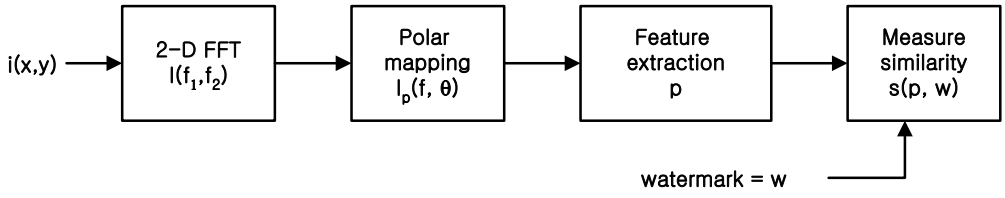
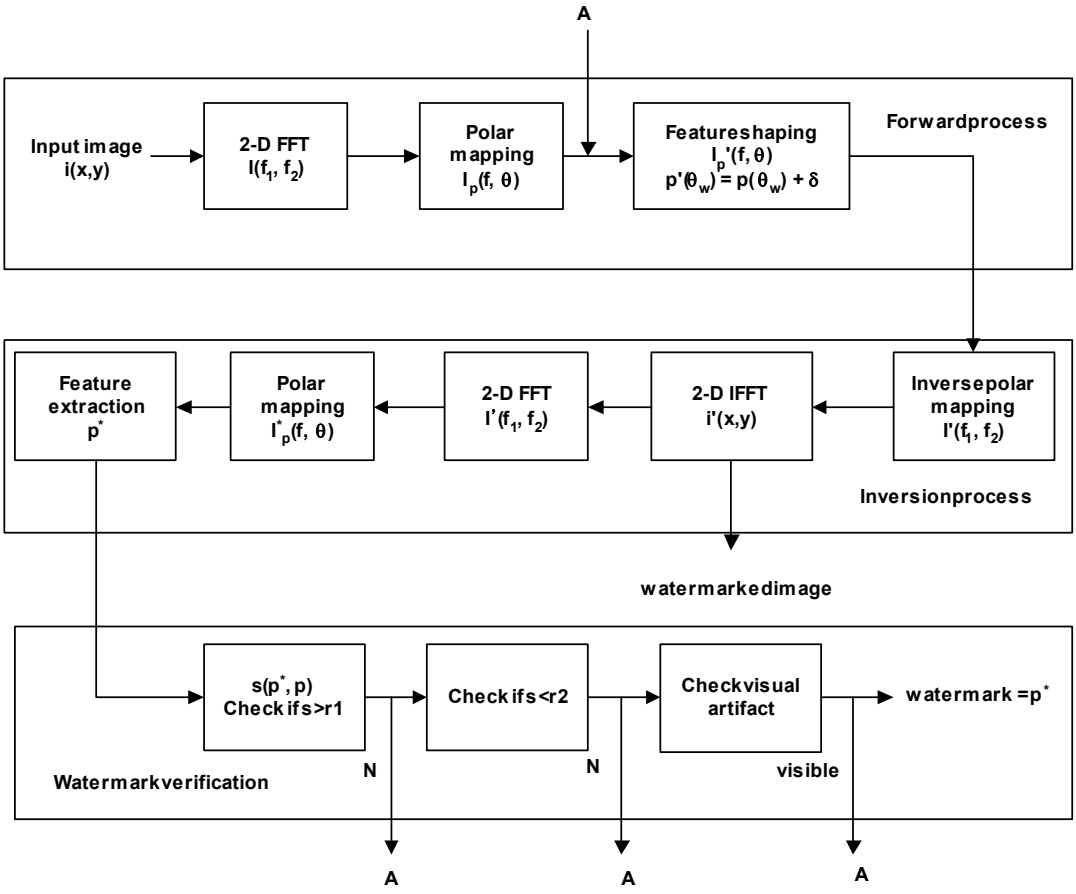

Feature p of sample images
Lena
Fishingboat
Pentagon
p(degree)
θ
150
100
50
0
-50
-100
0
20
40
60
80
100
120
140
160
180

Figure 2.


i(x,y)
2−D FFT
$I(f_1,f_2)$
Polar
mapping
$I_p(f, \theta)$
Feature
extraction
p
Measure
similarity
s(p, w)
watermark = w

(a)


A
Input image
i(x,y)
2-D FFT
$I(f_1, f_2)$
Polar
mapping
$I_p(f, \theta)$
Featureshaping
$I_p'(f, \theta)$
$p'(\theta_w) = p(\theta_w) + \delta$
Forwardprocess
Feature
extraction
p^*
Polar
mapping
$I^*_p(f, \theta)$
2-D FFT
$I'(f_1, f_2)$
2-D IFFT
i'(x,y)
Inversepolar
mapping
$I'(f_1, f_2)$
Inversionprocess
watermarkedimage
$s(p^*, p)$
Check if s>r1
Check if s<r2
Checkvisual
artifact
watermark =p^*
Watermarkverification
N
N
visible
A
A
A

(b)

Figure 3.

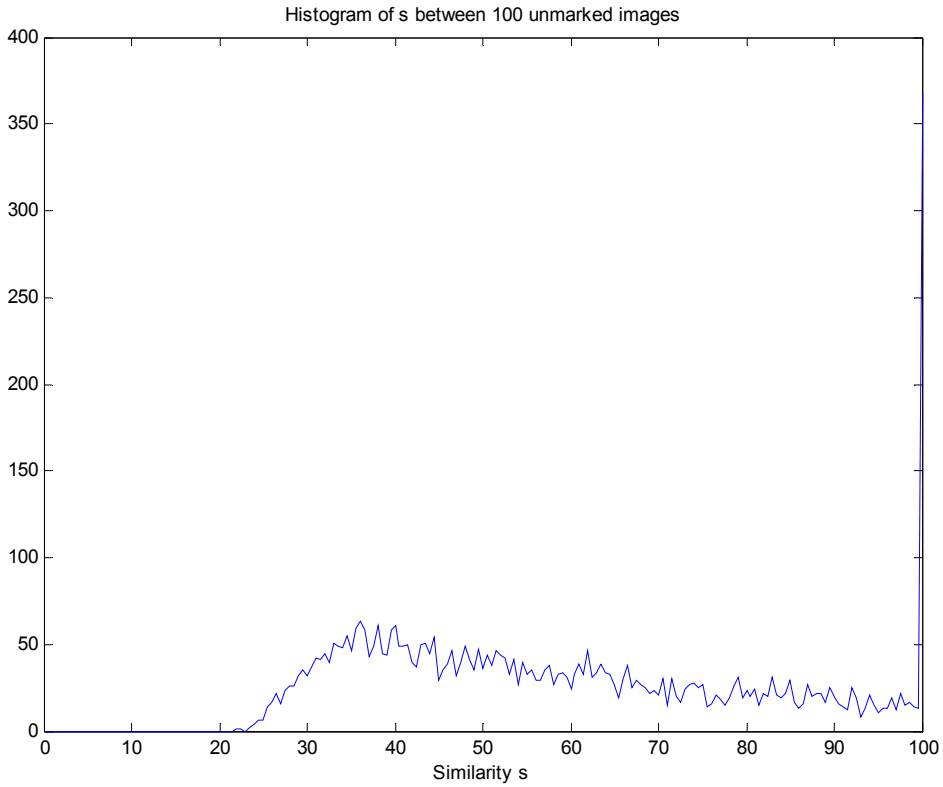
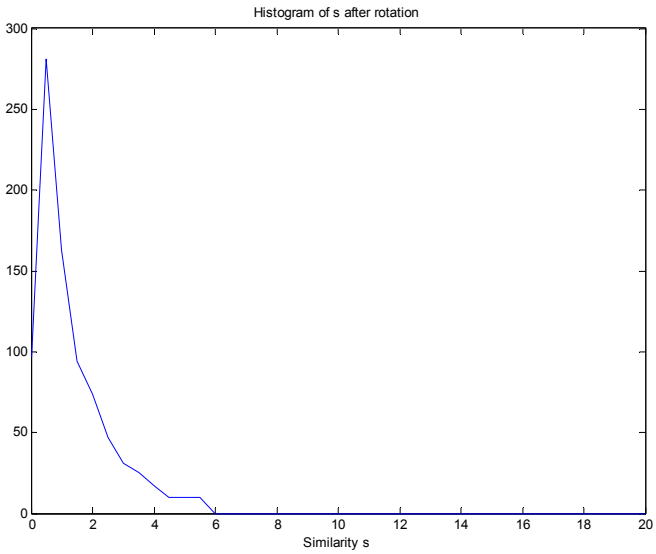
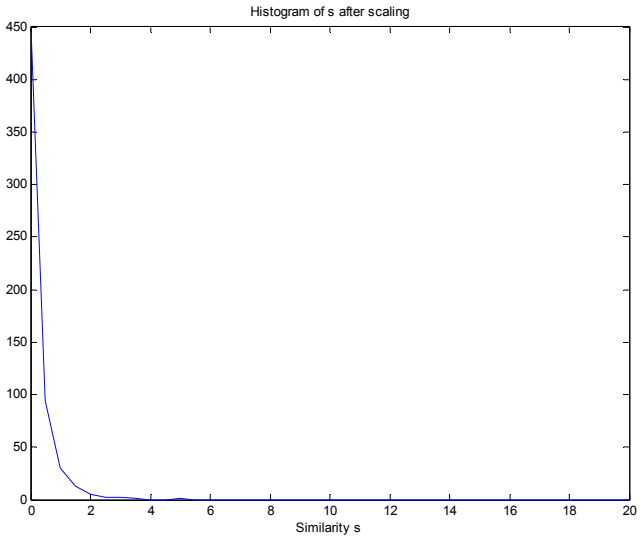

Histogram of s between 100 unmarked images
400
350
300
250
200
150
100
50
0
0
10
20
30
40
50
60
70
80
90
100
Similarity s

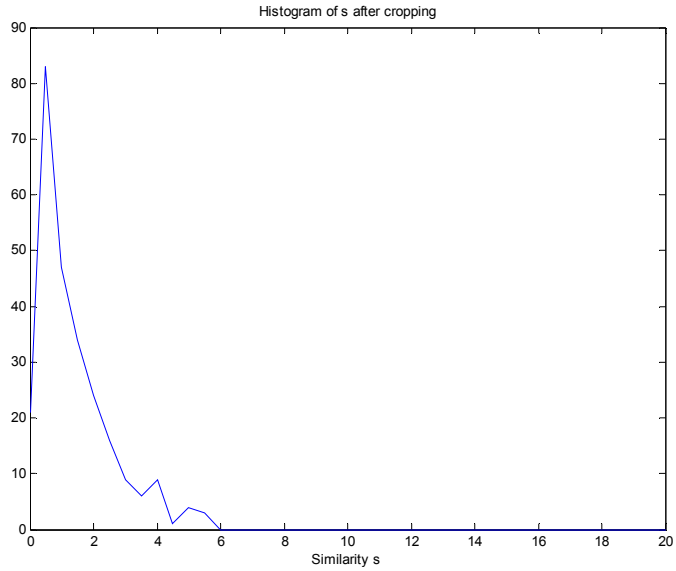
Figure 4.


Histogram of s after rotation
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14
16
18
20
Similarity s

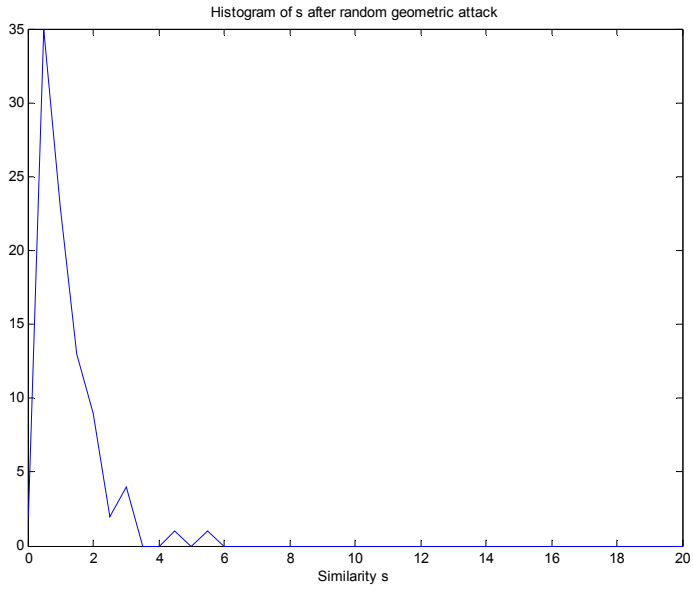
(a)


Histogram of s after scaling
450
400
350
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14
16
18
20
Similarity s

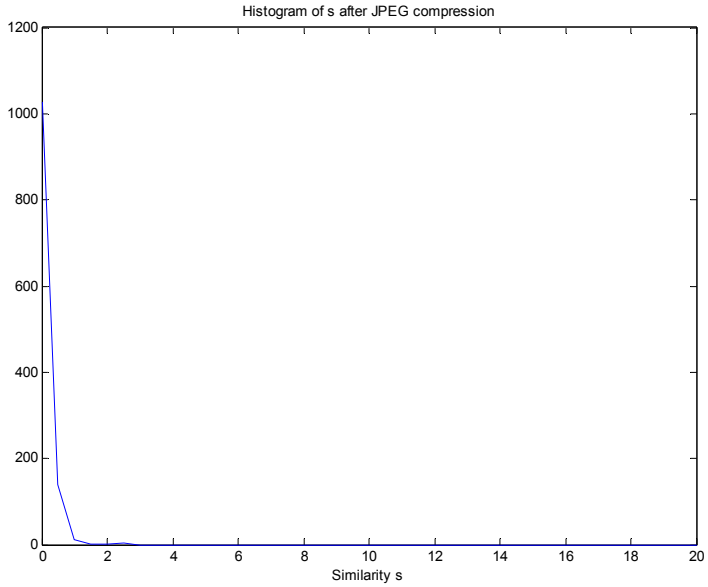
(b)


Histogram of s after cropping
90
80
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
18
20
Similarity s

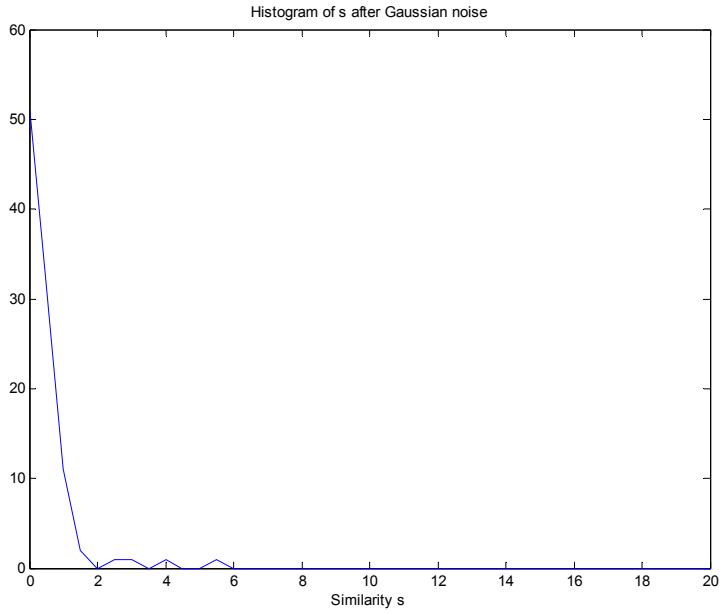
(c)


Histogram of s after random geometric attack
35
30
25
20
15
10
5
0
0
2
4
6
8
10
12
14
16
18
20
Similarity s

(d)


Histogram of s after JPEG compression
1200
1000
800
600
400
200
0
0
2
4
6
8
10
12
14
16
18
20
Similarity s

(e)

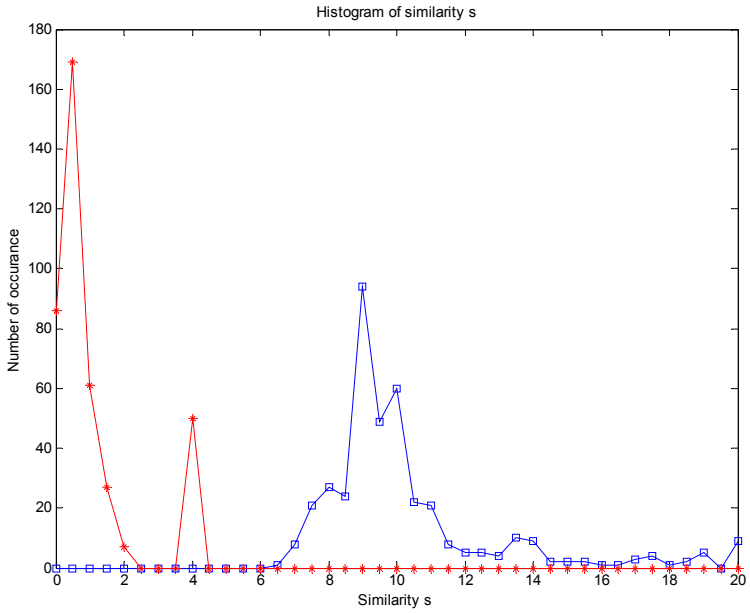

Histogram of s after Gaussian noise
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
18
20
Similarity s

(f)

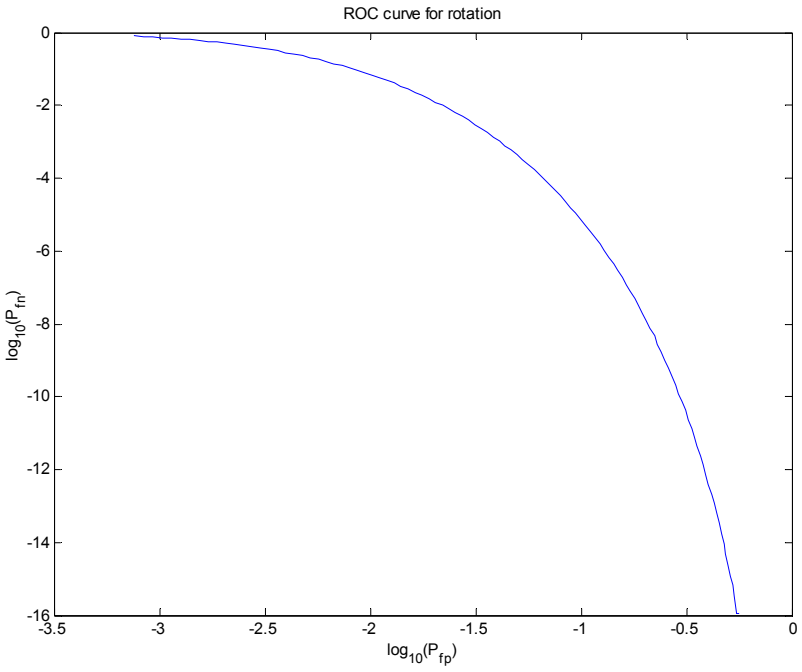
Figure 5.

(a) **(b)**

Figure 6.

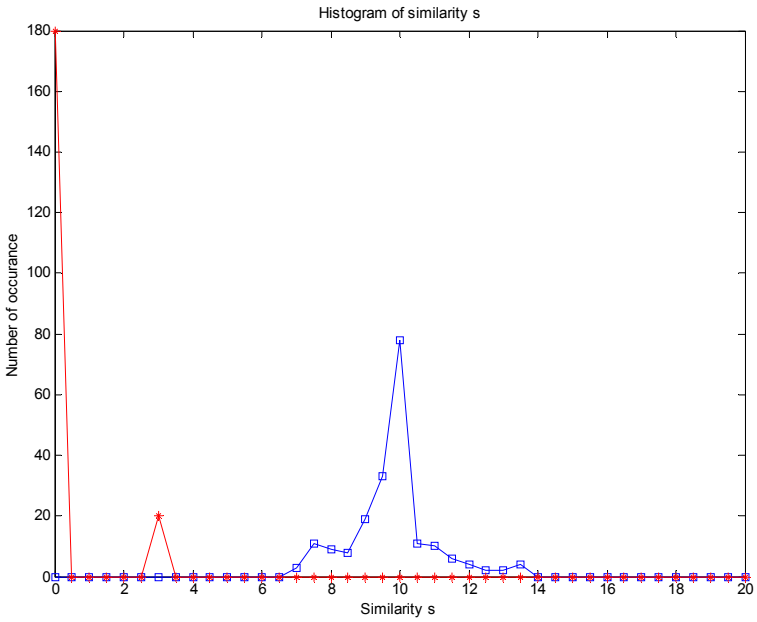

Histogram of similarity s
Number of occurance
Similarity s
0
20
40
60
80
100
120
140
160
180
0
2
4
6
8
10
12
14
16
18
20

(a)

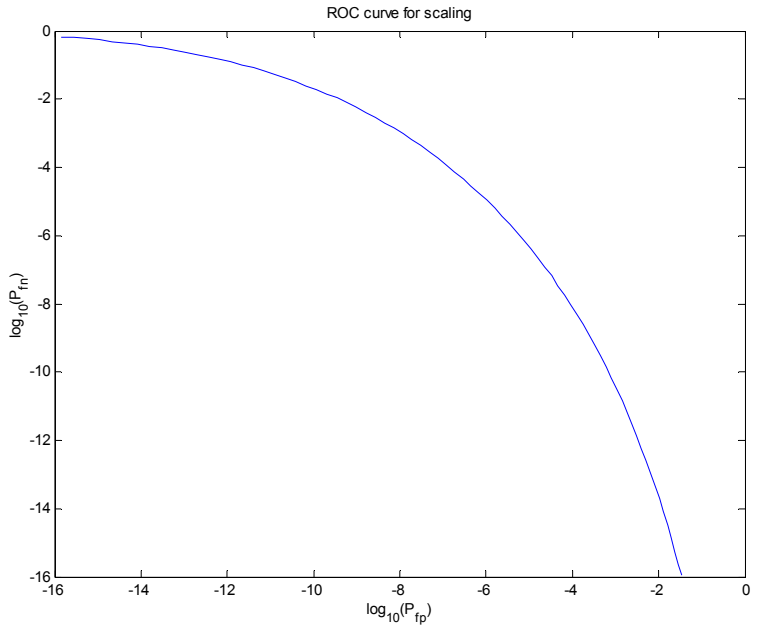

ROC curve for rotation
log10(Pfn)
log10(Pfp)
0
-2
-4
-6
-8
-10
-12
-14
-16
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0

(b)

Figure 7.

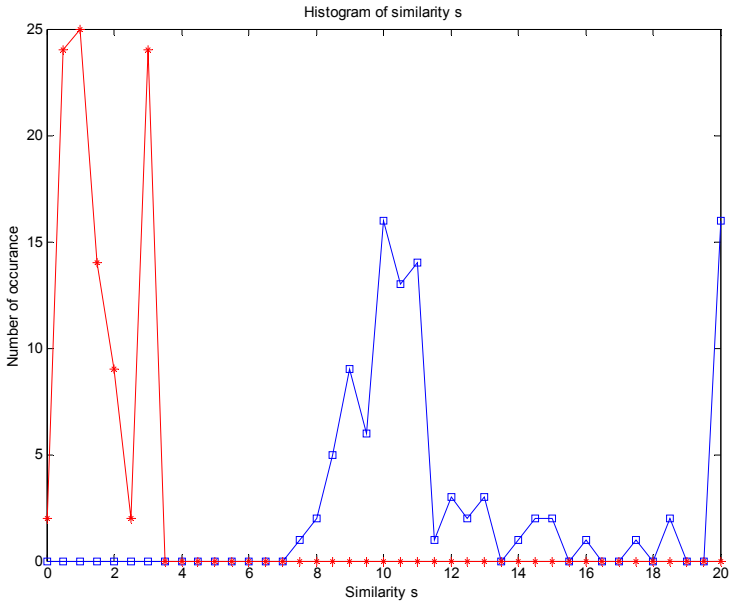

Histogram of similarity s
Number of occurance
Similarity s
0
20
40
60
80
100
120
140
160
180
0
2
4
6
8
10
12
14
16
18
20

(a)

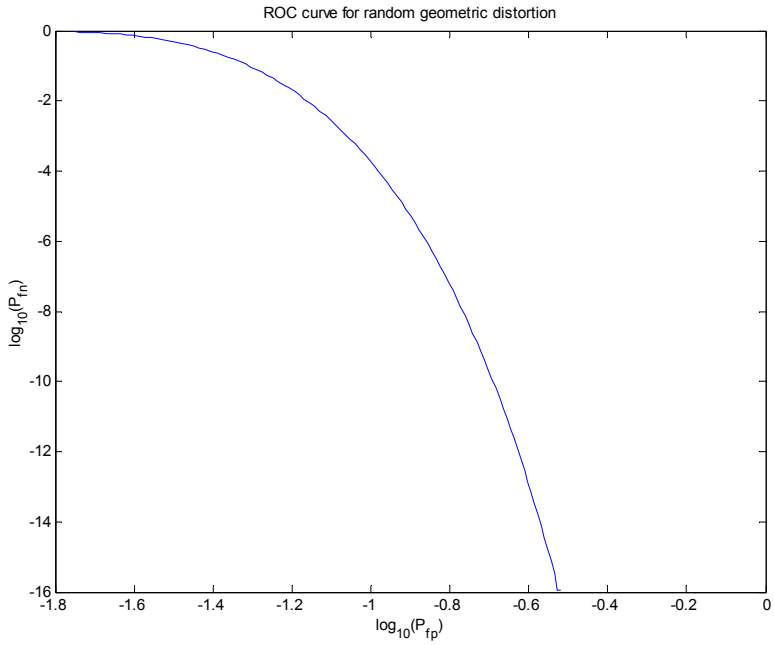

ROC curve for scaling
log10(Pfn)
log10(Pfp)
0
-2
-4
-6
-8
-10
-12
-14
-16
-16
-14
-12
-10
-8
-6
-4
-2
0

(b)

Figure 8.

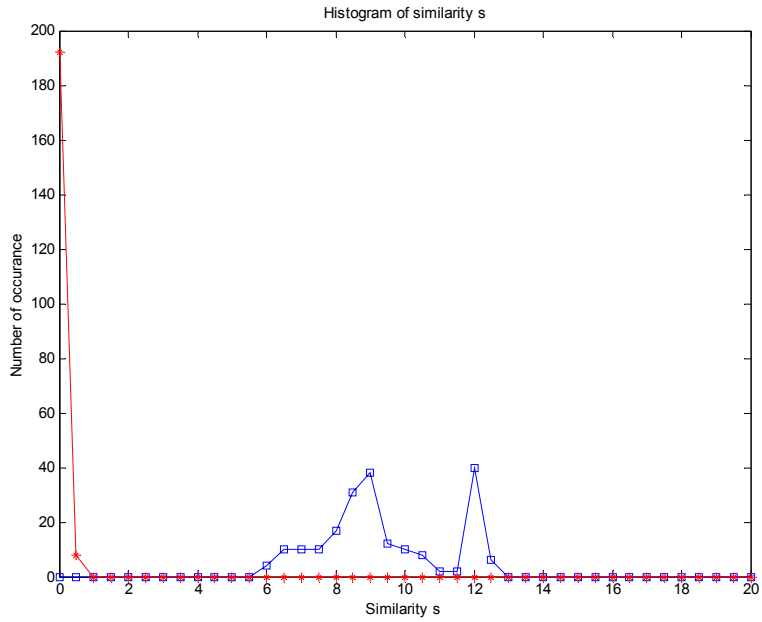

Histogram of similarity s
Number of occurance
Similarity s
0
5
10
15
20
25
0
2
4
6
8
10
12
14
16
18
20

(a)

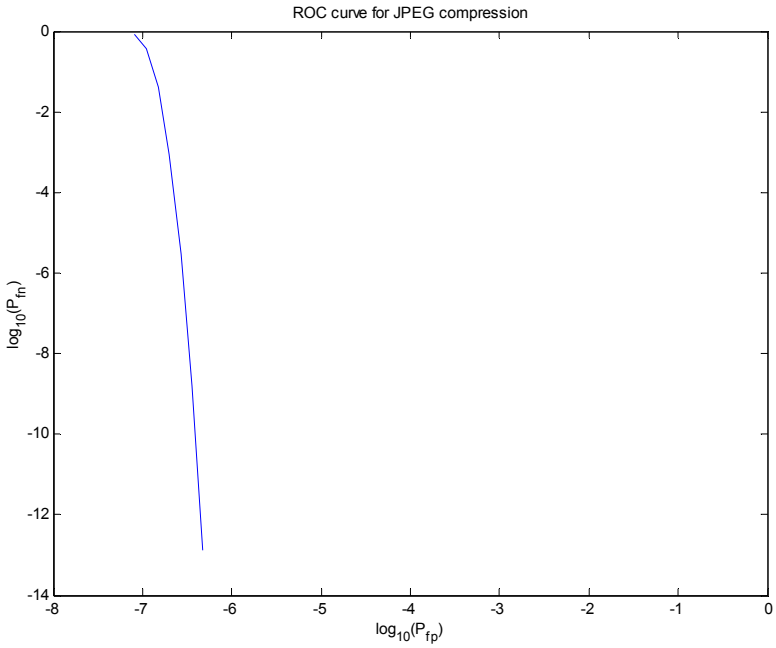

ROC curve for random geometric distortion
log10(Pfn)
log10(Pfp)
0
-2
-4
-6
-8
-10
-12
-14
-16
-1.8
-1.6
-1.4
-1.2
-1
-0.8
-0.6
-0.4
-0.2
0

(b)

Figure 9.

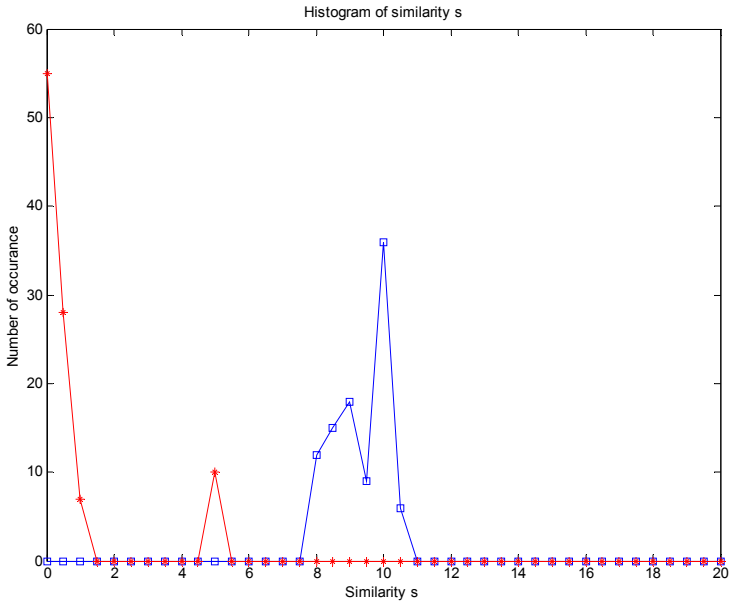

Histogram of similarity s
Number of occurance
Similarity s
0
20
40
60
80
100
120
140
160
180
200
0
2
4
6
8
10
12
14
16
18
20

(a)

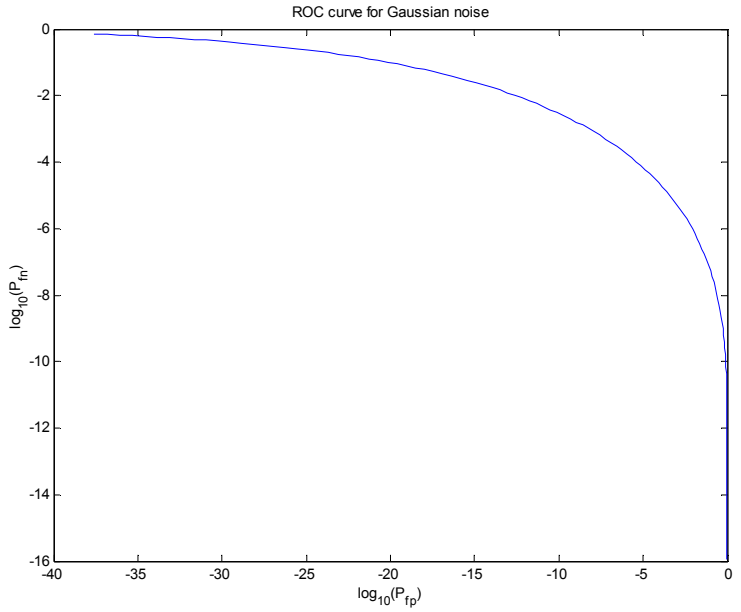

ROC curve for JPEG compression
log10(Pfn)
log10(Pfp)
0
-2
-4
-6
-8
-10
-12
-14
-8
-7
-6
-5
-4
-3
-2
-1
0

(b)

Figure 10.


Histogram of similarity s
Number of occurance
Similarity s
0
10
20
30
40
50
60
0
2
4
6
8
10
12
14
16
18
20

(a)


ROC curve for Gaussian noise
log10(Pfn)
log10(Pfp)
0
-2
-4
-6
-8
-10
-12
-14
-16
-40
-35
-30
-25
-20
-15
-10
-5
0

(b)