A Bootstrap Attack on Digital Watermarks in the Frequency Domain

Sam Behseta[1], Charles Lam[2], and Robert L. Webb[3]

[1] Department of Mathematics, California State University,
Fullerton, CA, 92834, USA
statistician@gmail.com
[2] Department of Mathematics, California State University,
Bakersfield, CA, 93311, USA
clam@csub.edu
[3] Department of Computer Science, California Polytechnic State University,
San Luis Obispo, CA, 93407, USA
webb@calpoly.edu

Abstract. In this paper, we propose five simple algorithms to execute a collusion attack given several watermarked documents. Each document considered is a picture represented as a matrix of two dimensional Discrete Cosine Transform (DCT2) coefficients. Our algorithm is independent of media type. Bootstrap methods are used to construct confidence intervals for each DCT2 coefficient and determine its uncertainty. Using simulation studies we show that Bootstrap procedures are highly efficient with respect to the number of iterations and sample size per iteration while maintaining stellar probabilistic coverage, providing results at least as good as averaging or taking the median of signals. Most importantly, a set of simulation studies suggest that the precision of our heuristic methodology increases quickly when the number of watermarked copies are increased, but good probabilistic coverage is achieved with a low number of independently watermarked copies. We conjecture that the Bootstrap methodology will be highly effective in reconstructing the original signal for documents with high redundancy.

1 Introduction

A secure spread spectrum digital watermark is a signal added into a digital document in such a way that the alteration to the final use, usually analog, of the digital signal is minimal. However, the watermark is strategically placed so that common operations on the signal will not destroy it. Since the watermark can be recovered from the host signal if the original signal is available, the watermark can be used to identify the original owner of a specific copy [3]. For a discussion of the overall motivations for digital watermarking, see [3,15].

The questions of security and robustness of these types of watermarks have stimulated a large number of research activities. For example, see [3,6,10,11,19,25]. In this paper, we describe resampling algorithms to determine a confidence interval for the true value of any part of the original signal in a

L. Chen, M.D. Ryan, and G. Wang (Eds.): ICICS 2008, LNCS 5308, pp. 361–375, 2008.

collusion attack. This coverage shows that we can obtain knowledgeable information about the original unwatermarked document, when only a small number of watermarked documents are available. By using a measure of centrality such as the median, we can also produce a document at least as accurate to the original document as averaging signals of all available watermarked documents.

Our algorithms generate two statistics that can be used to determine the original signal: the percentage of altered 2D Discrete Cosine Transform (DCT2) coefficients that are inside a particular confidence interval, and the average span of that confidence interval. We develop methods which can be used to make confidence intervals as narrow as possible without effecting their coverage rate.

The creators of secure spread spectrum digital watermarks attempted to approximate the original signal by averaging all of the available watermarked signals [3]. This method is satisfying because each watermarked signal is created by adding a sequence with low correlation.

Unfortunately, simple averaging does not satisfactorily reconstruct the original image unless the number of attackers is large. The metric to determine the similarity of watermark W with some other watermark W^* is

$$\text{sim}(W, W^*) = \frac{W \cdot W^*}{\sqrt{W \cdot W}}. \tag{1}$$

Let $W_1, W_2, ...W_n$ be n sequences with low correlation. Let $W^* = \frac{W_1+W_2+...+W_n}{n}$, then

$$\text{sim}(W_1, W^*) = \frac{W_1 \cdot \left(\frac{W_1+W_2+...+W_n}{n}\right)}{\sqrt{W_1 \cdot W_1}} \approx \frac{W_1 \cdot W_1}{n\sqrt{W_1 \cdot W_1}} \approx \frac{\text{sim}(W_1, W_1)}{n}.$$

When sequences follow the white noise assumption, then $\text{sim}(W_1, W_1) \approx \sqrt{L}$, where L is the length of the sequence W_1. Therefore, if $\frac{\sqrt{L}}{n}$ is large enough to register a positive match between two watermarks, then simple averaging is insufficient to obscure the original watermark.

If we continue to assume that watermarks are drawn from $N(0, 1)$, and if $\text{sim}(W_1, W^*) = s$, then the probability that W_1 and W^* are unrelated is the same as the probability of choosing a sample from $N(0, 1)$, which is s standard deviations away from the mean. This is demonstrated by the fact that if $W = w_1, w_2, ..., w_L$ is a fixed sequence of numbers from $N(0, 1)$, and $W^* = w_1^*, w_2^*, ..., w_L^*$ is independently generated from samples of $N(0, 1)$, then

$$W \cdot W^* = w_1 w_1^* + w_2 w_2^* + ...w_L w_L^*$$

will follow the distribution $N(0, w_1^2 + w_2^2 + ... + w_L^2)$. Notice that the standard deviation of this distribution is $\sqrt{W \cdot W}$. Therefore, $\text{sim}(W, W^*)$ will follow the $N(0, 1)$ distribution if W^* is independently generated from a fixed W. For the averaging collusion attack to be effective, n must be very large or L must be very small. However, these are generally unreasonable assumptions [3]. The algorithms in this paper address these issues by using a small number of attackers to find a narrow confidence interval with a high probability of containing the

true value of the original signal. We also show that choosing the median of the generated confidence interval produces a result at least as good as the average.

Our work was initially inspired by Cox, et al. [3]. They detailed a watermarking system that is used to determine the original owner of a signal. In this paper, we use the watermarking method described by Zhu, et al. [25], which is a modification of the method presented by Cox [3]. The term "digital watermark" has been used for many other purposes (see Furon [11], Delaigle, et al. [4], Cannons and Moulin [1], Soriano, et al. [17], Wolfgang, Podilchuk and Delp [21] and references therein).

Collusion attacks have been specifically addressed by Doërr and Dugelay [6] where redundancy in the host signal was used to distort the included watermark. Vinod and Bora [19] used similar methods on video signals, where redundancy is especially applicable. These types of geometric attacks can be mitigated using the methods presented in Dong [5].

The specific problem of this paper has been considered by Comesana, et al. [2] Their paper also includes an especially detailed discussion of the difference between the motivations for watermarking presented in [18] and [3]. A theoretical result on the limitations of the method presented in [3] and a possible attack is presented in Ergun, et al [10]. However, there are perceptible negative effects on the host signal when this attack is used.

More recent work on collusion attacks have been done in [23,24,22,20]. In this series of papers, the authors classify collusion attacks into linear and non-linear categories where simple averaging is the most naive of the linear attacks and using the median of the colluding copies is the most naive of the non-linear attacks. Specifically, in [24], the authors introduce a non-linear attack that reduces the similarity metric. Further work on non-linear attacks has been done by [14,13]. The attack presented in our paper is non-linear, but extends previous schemes by using the non-parametric Bootstrap methodology to mitigate possible effects of outliers in the parametric watermark.

The term "watermark" has been used in the literature for other purposes. Included are references to papers explaining these other uses. For a discussion of video watermarking see Podilchuk and Delp [15]. The relationship between video and image watermarking is explored in Doerr and Dugelay [7]. Another application for watermarking is steganography where a detailed signal is hidden in a host signal. Vila-Forcén, et al [18] have used collusion attacks to extract information about the host signal and original signal in this setting.

This paper is organized in the following way: We describe our notations and algorithms in section 2. In section 3, we summarize the extensive empirical simulations we performed using the "Lena" picture as the host signal. The results demonstrate that the output of the algorithms are effective in terms of speed and ability to determine precise confidence intervals with high probability of containing the true value of the original signal. At the end of section 3, we include a discussion of the technical details of the simulation. Section 4 contains a discussion of our methods and also includes some suggestions on future work. Tables containing the numerical results are presented in the appendix.

2 Methods

2.1 The Bootstrap Methodology

Let $X_1^*, ...X_n^*$ be an independently drawn and identically distributed sample from $\hat{F}$, the empirical cumulative distribution function of a dataset $X_1, ..., X_n$. $X_1^*, ...X_n^*$, or the Bootstrap sample, can be acquired by sampling with replacement from $X_1, ..., X_n$. Also, suppose that T_n is an estimator of some unknown parameter θ (for example, $T_n = \bar{X}$). The Bootstrap sample is nonparametric in a sense that since it is obtained from the dataset, it makes no assumptions regarding the underlying statistical model and its parameters. By generating repeated Bootstrap samples, one can obtain a probability distribution for T_n^*, hence being able to assess its uncertainty.

Let $U_F(x) = Pr(T_n \leq x)$ denote the distribution of T_n. By replacing F with $\hat{F}$, we have $U_{\text{boot}}(x) = U_{\hat{F}}(x) = Pr[T_n^* \leq x|(X_1, ..., X_n)]$, the conditional distribution of T_n^* given data. Let ρ_∞ be a metric generated by a sup-norm[1] on the space of all distributions in R^p, for an integer p, representing the dimensionality of the distribution space. Then the following results hold (See Shao and Tu [16] for details):

(1) If T is continuously ρ_∞-Hadamard differentiable, then the Bootstrap estimator U_{boot} is strongly consistent. In other words, $\rho_\infty(U_{\text{boot}}, U_n) \to 0$ (almost surely).
(2) If T is continuously ρ_r-Frechet differentiable, then the Bootstrap estimator U_{boot} is strongly consistent.

These results guarantee the fact that the Bootstrap distribution U_{boot} is consistent for many estimators such as the mean and the median.

2.2 Notation

Let the original signal be represented by an $I \times J$ matrix D_0 such that the value of the (i, j)-th element is $D_{0,i,j}$, where $i = 1, ..., I$, $j = 1, ..., J$. In this context, each element is the value of a DCT2 coefficient. There are also K independently watermarked documents which will be used to make statistical inference with regard to the original signal. These matrices are denoted by D_k, where $k = 1, ..., K$. Similarly, $D_{k,i,j}$ represents the (i, j)-th element of the k-th matrix, also a DCT2 coefficient.

The Bootstrap is performed by sampling from K existing signals. See [8] for a discussion of the theoretical properties of the Bootstrap procedure. The probabilistic framework for determining a confidence interval for $D_{0,i,j}$ is obtained by sampling n many elements with replacement from $D_{1,i,j}, \ldots, D_{K,i,j}$, where $n < K$ followed by calculating some measure of centrality from this sample. This procedure is repeated B times to create B possible values of the given statistic.

[1] $\| h \|_\infty$ is the sup-norm of a function h on R^p, if $\| h \|_\infty = \sup_x |h(x)|$.

For our simulation, the K signals are created by adding samples taken from $N(0,1)$ to some of the elements of the original signal. For some subset $S \subset \{(i,j) : i = 1, ..., I, j = 1, ..., J\}$, we let

$$D_{k,i,j} = D_{0,i,j} + \alpha W_{k,i,j}, (i,j) \in S, \tag{2}$$

for some appropriate constant α, where $W_{k,i,j}$ is a sequence of numbers from $N(0,1)$. The constant α can be viewed as the standard deviation of the added noise.

Using a simulation technique described in the next section, we calculate confidence intervals to determine the average span of each interval and the empirical percentage of the times that the original signal is contained in the interval.

2.3 Algorithms

We propose five simple algorithms to calculate confidence intervals for the original signal. There are three parameters shared by all proposed algorithms: the number of watermarked signals (K), the number of Bootstrap iterations (B), and the number of samples per iteration (n), hereafter referred to as "sample size". The basic framework of all algorithms is described in Algorithm 0.

Let (i,j) be a typical element of the signal. There are K associated values of this element, one in each of $D_{k,i,j}$, where $k = 1, ..., k$. In this paper, we sample n of the elements with replacement. Next, we calculate two types of measurements. First, a metric that reflects the distance to a pre-assigned measure of centrality. Second a central statistic such as the mean. Finally, this process is repeated B times to generate B bootstrapped observations. Let $C_{i,j}$ be the list of B values in increasing order. Let $C_{i,j}^{(p)}, p \in (0,1)$ be the value in $C_{i,j}$ so that $[pB]$ of the values are smaller than $C_{i,j}^{(p)}$. The median of those B values ($C_{i,j}^{(0.5)}$) is the bootstrapped median.

Additionally, we calculate associated confidence intervals. Let $\gamma \in (0,1)$. The Bootstrap $(1-\gamma)$-confidence interval for (i,j) is defined as

$$(C_{i,j}^{(\gamma/2)}, C_{i,j}^{(1-\gamma/2)}). \tag{3}$$

The average confidence span is defined as

$$\frac{\sum_{(i,j)\in S} \left(C_{i,j}^{(1-\gamma/2)} - C_{i,j}^{(\gamma/2)} \right)}{|S|\alpha}, \tag{4}$$

where S is the subset of the signal that has been modified by the watermarking process. The appearance of α in the denominator is motivated by the fact that, in the initial watermark process, the signals were magnified by a scale of α. Since we know the original signal, we can also calculate the percentage of the confidence intervals that actually capture $D_{0,i,j}$, the value of the original signal. For any $(1-\gamma)$-confidence interval, this can formally be stated as

$$\frac{\sum_{(i,j)\in S} V_{i,j}^{(1-\gamma)}}{|S|}, \tag{5}$$

where

$$V_{i,j}^{(1-\gamma)} = \begin{cases} 1 & \text{when } D_{i,j} \in (C_{i,j}^{(1-\gamma/2)}, C_{i,j}^{(\gamma/2)}) \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 0: The General Approach

(0.1) For the element (i, j), sample with replacement n many observations with $n < K$.
(0.2) Define a Metric M which can be implemented pointwise on each of the IJ elements. Thus, the metric can be presented as $M_{i,j}$, for $i = 1, ..., I$ and $j = 1, ..., J$.
(0.3) Calculate a statistic $R_{i,j}$ based on the implemented metric on the set of n sampled elements.
(0.4) Repeat steps (2), (3), and (4), B many times.
(0.5) Consider an appropriate quantile or an appropriate measure of centrality of the probability distribution of $R_{i,j}^B$, along with a variability measure associated with it. The superscript B here is to emphasize the number of the Bootstrap repetitions.
(0.6) Form a pointwise $1-\gamma$ percentile level confidence interval for the statistic. In other words, form a confidence interval for the element (i, j) with $C_{i,j}^{(\gamma/2)}$, and $C_{i,j}^{(1-\gamma/2)}$ as its lower and upper bounds respectively.
(0.7) Calculate the pointwise width (span) of the confidence interval of step (7).
(0.8) Calculate the percentage of the confidence intervals covering the actual element of the signal D'.

Algorithm 1: Detection Via Boostrapping the Mean

(1.3) Take $R_{i,j} = \frac{\sum_l D_{l,i,j}}{n}$, where l represents the bootstrapped sample.
(1.5) Consider Median$(R_{i,j}^B)$ as the measure of centrality.

Algorithm 2: Detection Via Boostrapping the Median

(2.3) Take $R_{i,j} = \text{Median}_l(D_{l,i,j})$.
(2.5) Consider Median$(R_{i,j}^B)$ as the measure of centrality.

Algorithm 3: Detection Via Boostrapping the Geometric Mean

(3.3) Here, we take $R_{i,j} = \sqrt[n]{\prod_l (D_{l,i,j})}$.
(3.5) Consider Median$(R_{i,j}^B)$ as the measure of centrality.

Algorithm 4: Detection Via Closeness to Average

(4.2) Define $M_{i,j} = \inf_l |D_{l,i,j} - \frac{\sum_{l=1}^n D_{l,i,j}}{n}|$.
(4.5) Consider Median$(R_{i,j}^B)$ as the measure of centrality.

Algorithm 5: Pairwise Comparisons

(5.2) Calculate $u = 1, ..., \binom{n}{2}$ many distances $|D_{l,i,j} - D_{m,i,j}|$, where $1 \leq m < l \leq n$. Order the distances and label them as $M_{i,j}(1), ..., M_{i,j}(u)$ where $M_{i,j}(1)$ represents the smallest distance. Let $\left(D_{l,i,j}(r), D_{m,i,j}(r)\right)$ correspond to the pair of values belong to $M_{i,j}(r)$.

(5.3) If $n \geq 5$, let $R_{i,j} = \frac{\sum_{r=1}^{3}\left(D_{l,i,j}(r)+D_{l,i,j}(r)\right)}{6}$.

If $n = 4$, let $R_{i,j} = \frac{\sum_{r=1}^{2}\left(D_{l,i,j}(r)+D_{l,i,j}(r)\right)}{4}$.

If $n \leq 3$, let $R_{i,j} = \frac{\left(D_{l,i,j}(1)+D_{m,i,j}(1)\right)}{2}$.

(5.5) Consider Median$(R_{i,j}^{B})$ as the measure of centrality.

Note that the watermark sequence is inserted into a subset of the original signal. This fact can be used to improve the efficiency of the simulation. Specifically, the Bootstrap procedure could be performed on the entire set of K watermarked signal. However, this will increase the computational complexity of the simulation. Restricting our analysis to this subset will not change the results as the rest of the signal is not important to the watermark analysis. This restriction will greatly decrease the time required for each simulation.

3 Results

3.1 Simulation

To investigate the performance of the proposed algorithms, we perform a series of simulations. These simulations produce two outputs. The first output is the pointwise coverage of the confidence intervals. Our empirical findings are consistent with the Bootstrap theory [9,16]. Additionally, we determine the average span of each confidence interval. This output is used to determine which algorithms generate tighter bounds while still maintaining high coverage.

We organize the results around three parameters shared by all the algorithms: 1–the number of Bootstrap iterations B, 2–the sample size n, and 3–the number of watermarked copies K. Our simulations show that all algorithms remain consistent for different combinations of parameter values (B=50, 100, 1000, n=3, 5, 7). Additionally, we found that increasing K greatly decreases the average span and increases the coverage. Most importantly, the average span of the confidence intervals remains small while the coverage of the confidence intervals increases. This suggests that a collusion attack might be feasible with a small number of colluding documents.

We perform simulations to calculate the coverage and the average span of each confidence interval with $K = 10, 15, 20$ independently watermarked copies using sample sizes $n = 3, 5, 7$. Each simulation was repeated for $B = 50, 100, 1000$ iterations to ensure the stability of the result. The selected results are presented in the Appendix.

(a) (b) (c)

Fig. 1. (a) The original Lena picture (b) the significant parts (c) a watermarked version

We convert the "Lena" picture (Figure 1(a)) into a signal of DCT2 coefficients. We watermark the lowest frequency coefficients, excluding the constant coefficient. In Figure 1(b), we show the analog representation of the coefficients that have been watermarked. In Figure 1(c), we include the picture with the watermark inserted. In Figure 2, we demonstrate the reconstructed picture obtained via replacing each DCT2 coefficient with the median of algorithms 1,2, and 5 respectively.

To quantify the reconstructive quality of Figures 2(a), 2(b), and 2(c), we calculated the sequence W^* by reversing the watermarking process with the reconstructed signal D^* using

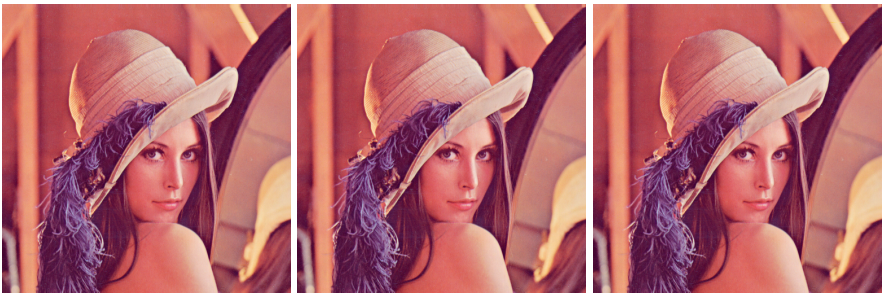
$$W^* = \frac{D^* - D_0}{\alpha}. \tag{6}$$

This recovered watermark is then compared to one of the original watermarks W_i, $1 \leq i \leq K$, using the similarity metric in equation (1). The similarity metric reveals that almost all of our algorithms perform the same. For example, comparing a watermarked image with itself using the above equations yield the result of 146.87. When this is done repeatedly, all results are clustered around $147.6 = \sqrt{21777}$ where 21777 is L, the length of the watermark. When two independently created watermarked pictures are compared using this algorithm, the results are clustered around zero [3]. When the median of the confidence intervals is used to construct W^*, then

$$\mathrm{sim}(W_i, W^*) \approx \frac{147.6}{k},$$

where W_i is the watermark associated with one of the documents used to create W^*.

All of these simulations have been performed using the optimization described at the end of section 2 on freely available software. In terms of complexity, with a watermark of size L on a Bootstrap of B repetitions per signal and n samples per repetition, the algorithm runs in time $O(LBn)$.



Algorithm 1 Algorithm 2 Algorithm 5

Fig. 2. The reconstructed image produced from 100 Bootstrap iterations with a sample size of 7 and a population of 20 using algorithms 1, 2, and 5

3.2 Numerical Results

The tables in the Appendix detail the output of the previously described algorithms. Each table represents a different algorithm with an indicated number of watermarked pictures as an input. Tables have not been included for algorithms three and four because these algorithms produced results that were indistinguishable from algorithm one. Each algorithm was tested with 10, 15, and 20 watermarked documents. Within each table, the columns are organized in categories of 50, 100, and 1000 Bootstrap iterations and subcategories of 3, 5, and 7 samples per Bootstrap iteration. Each row indicates the percentile confidence interval with confidence levels 30, 50, 70, 80, 90, 95, and 99. For example, in table A1, these results refer to algorithm one performed with ten independently watermarked documents. In column 100, sub-column 5, and row 90, we see that the 90 percentile confidence interval contained the correct value 92.9% of the time and the average span of the intervals was 1.26. For a formal definition of the percentile confidence interval, see equation (3).

All of the algorithms perform well even with a small number of watermarked pictures and a low number of Bootstrap iterations. However, there are differences worth discussing. The primary result concerning the coverage is that the number of Bootstrap iterations quickly leads to diminishing returns. Running the Bootstrap procedure 1000 times does not yield better results than doing it 100 times. Most differences are much less than 10%. The difference between 100 Bootstrap iterations and 50 iterations is greater, but low iteration results are still impressive.

The number of independently watermarked documents does play a large role in the process. The most significant differences occur at the lower confidence intervals. For example, with 10 watermarked documents, using algorithm 1 with 100 Bootstrap iterations and a sample size of 5, the middle half of the samples contained the true value 59.6% of the time. In the same situation with 20 documents, the result improves to 75.8%.

The span also varies greatly with the different parameters of each algorithm. The results were similar to those of the percentage coverage, so there will not be

as great an emphasis on each of the different effects of the parameters. The most important result is that the number of Bootstrap iterations does not change the span much and that the size of the population of watermarked pictures does have a positive correlation with the span.

Clearly, the precision of the result will decrease as the percentile confidence interval increases as there is a positive correlation with average span. Pragmatically, one would like to strike a balance between the percentile confidence interval and the precision. As shown in table A9, at lower percentile confidence intervals with small span, algorithm 5 still produces satisfactory results. This trend continues to hold with fewer Bootstrap iterations and smaller numbers of watermarked documents.

In the end, the objective is to get a high coverage and a low span. To quantify this goal we look at the ratio of the coverage to the span. Referencing the tables in the Appendix, it is clear upon inspection that algorithm 1 produces the greatest ratio of coverage to span. This ratio increases greatly with the population of watermarked pictures because algorithm 1 produces smaller spans with larger populations of watermarked pictures in each Bootstrap iteration, and larger samples will produce a better ratio. For specific examples refer to table A7.

The median of the outputs from each Bootstrap iteration forms the reconstructed image. When watermarked, the PSNR of an image is about 38.4dB. After the process is completed all algorithms produce an image with PSNR above 40dB. These PSNR values are close to the values obtained from the simple mean or median algorithm. However, using the similarity metric, we find that our proposed methods perform at least as good as the simple mean or median

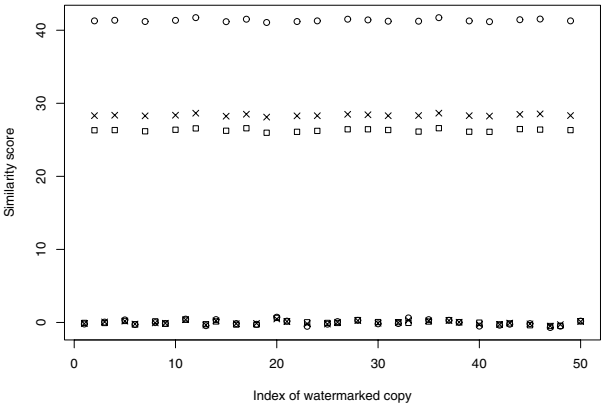


Fig. 3. Similarity values of the reconstructed image with 20 colluding images when the noise distribution is skewed to the right. The similarity values for the image reconstructed with the simple mean are represented with circles. The similarity values for the image reconstructed with the median are represented with crosses. The similarity values for the image reconstructed with algorithm 5 are represented with boxes. The similarity values with 30 other independently watermarked images are shown for comparison.

attacks. For example, when the colluders possess images in which some of the watermarked signals contain outliers, our method improves the similarity results significantly. This is due to the fact that our proposed statistical procedure is non-parametric. Consequently, the Bootstrap-based algorithms perform better than the median or the mean attack specifically in the presence of extreme values [9]. To elaborate, we generate noise from a right-skewed Gamma$(1, 1/5)$ distribution. Under this scenario, the attack in algorithm 5 improves the similarity metric by approximately 36% when compared to the mean and by approximately 7% when compared to the median (see figure 3).

4 Discussion

In this paper, we used the simple equal-tail two-sided Bootstrap percentile confidence interval which has second order accuracy. There have been theoretical improvements to this method. Examples include the Bootstrap bias-corrected percentile, the Bootstrap accelerated bias-corrected percentile and the hybrid Bootstrap confidence intervals (See Shao [16] for a comprehensive theoretical discussion).

We demonstrate the use of resampling to acquire measures of variation associated with a collusion. One should recognize that in principle, the reconstruction of the original document can be done with simple averaging [3]. However, in the process of Bootstrap, we obtain more information about the original unwatermarked document. We believe that Bootstrap resampling methods presented here could open up avenues of research by providing methods for quantifying the probability that the signal is contained in a given interval. This will allow researchers to use many nonparametric methods, resampling being only one of them.

We observe that the Bootstrap works with small sample sizes, low number of iterations and modest number of watermarked copies. Most importantly, while keeping a high PSNR value, the Bootstrap method lowers the value of the similarity metric in comparison with frequently used attacks such as the mean and the median. These are reassuring results. We discuss the statistical inference using the multiple testing adjustment and the use of alternative confidence intervals in another paper. We are currently investigating the use of the Bootstrap on collusion attacks on watermarked motion pictures, where there is a high amount of signal redundancy. We conjecture that the Bootstrap methodology will be more effective at reconstructing the original signal in these cases.

Acknowledgements

We would like to thank Christopher Gutierrez, Joseph Sutton, and Max Velado for their help in the preparation of this paper.

References

1. Cannons, J., Moulin, P.: Design and Statistical Analysis of a Hash-Aided Image Watermarking System. IEEE Transactions on Image Processing 13(10), 1393–1408 (2004)
2. Comesaña, P., Pérez-Freire, L., Pérez-González, F.: The Return of the Sensitivity Attack. In: Barni, M., Cox, I., Kalker, T., Kim, H.-J. (eds.) IWDW 2005. LNCS, vol. 3710, pp. 260–274. Springer, Heidelberg (2005)
3. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing 6(12), 1673–1687 (1997)
4. Delaigle, J.F., De Vleeschouwer, C., Macq, B.: Watermarking algorithm based on a human visual model. Signal Processing 66, 319–335 (1998)
5. Dong, P., Brankov, J.G., Galatsanos, N.P., Yang, Y., Davoine, F.: Digital Watermarks Robust to Geometric Distortions. IEEE Transactions on Image Processing 14(12), 2140–2150 (2005)
6. Doërr, G., Dugelay, J.: Countermeasures for Collusion Attacks Exploiting Host Signal Redundancy. In: Barni, M., Cox, I., Kalker, T., Kim, H.-J. (eds.) IWDW 2005. LNCS, vol. 3710, pp. 216–230. Springer, Heidelberg (2005)
7. Doërr, G., Dugelay, J.: Security Pitfalls of Frame-by-Frame Approaches to Video Watermarking. IEEE Transactions on Signals Processing 52(10), 2955–2964 (2004)
8. Efron, B.: The Jackknife, the Bootstrap and other Resampling Plans. Society for Industrial and Applied Mathematics, Philadelphia (1982)
9. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall, New York (1993)
10. Ergun, F., Kilian, J., Kumar, R.: A Note on the Limits of Collusion-Resistant Watermarks. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 140–149. Springer, Heidelberg (1999)
11. Furon, T.: A Survey of Watermarking Security. In: Barni, M., Cox, I., Kalker, T., Kim, H.-J. (eds.) IWDW 2005. LNCS, vol. 3710, pp. 201–215. Springer, Heidelberg (2005)
12. Khayam, S.A.: The Discrete Cosine Transform (DCT): Theory and Application. Michigan State University (2003)
13. Kiyavash, N., Moulin, P.: A Framework for Optimizing Nonlinear Collusion Attacks on Fingerprinting Systems. In: 40^{th} Annual Conference on Information Sciences and Systems, pp. 1170–1175 (2006)
14. Kiyavash, N., Moulin, P.: On Optimal Collusion Strategies for Fingerprinting. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, p. V (2006)
15. Podilchuk, C.I., Delp, E.J.: Digital Watermarking: Algorithms and Applications. Signal Processing Magazine 18(4), 33–46 (2001)
16. Shao, J., Tu, D.: The Jackknife and Bootstrap. Springer, New York (2005)
17. Soriano, M., Fernandez, M., Cotrina, J.: Fingerprinting Schemes: Identifying the Guilty Sources Using Side Information. In: Barni, M., Cox, I., Kalker, T., Kim, H.-J. (eds.) IWDW 2005. LNCS, vol. 3710, pp. 231–243. Springer, Heidelberg (2005)
18. Vila-Forcén, J.E., Voloshynovskiy, S., Koval, O., Pérez-González, F., Pun, T.: Practical Data-Hiding: Additive Attacks Performance Analysis. In: Barni, M., Cox, I., Kalker, T., Kim, H.-J. (eds.) IWDW 2005. LNCS, vol. 3710, pp. 244–259. Springer, Heidelberg (2005)

19. Vinod, P., Bora, P.K.: A New Inter-Frame Collusion Attack and a Countermeasure. In: Barni, M., Cox, I., Kalker, T., Kim, H.-J. (eds.) IWDW 2005. LNCS, vol. 3710, pp. 147–157. Springer, Heidelberg (2005)
20. Wang, Z., Wu, M., Zhao, H., Liu, K.: Resistance of Orthogonal Gaussian Fingerprints to Collusion Attacks. In: Proceedings of International Conference on Multimedia and Expo., vol. 1, pp. 617–620 (2003)
21. Wolfgang, R.B., Podilchuk, C.I., Delp, E.J.: Perceptual Watermarks for Digital Images and Video. Proceedings of IEEE 87(7), 1108–1126 (1999)
22. Wu, M., Trappe, W., Wang, Z., Liu, K.: Collusion-resistant Fingerprinting for Multimedia. IEEE Signal Processing Magazine 21(2), 15–27 (2004)
23. Zhao, H., Wu, M., Wang, Z., Liu, K.: Forensic Analysis of Nonlinear Collusion Attacks for Multimedia Fingerprinting. IEEE Transactions on Image Processing 14(5), 646–661 (2005)
24. Zhao, H., Wu, M., Wang, Z., Liu, K.: Nonlinear Collusion Attacks on Independent Fingerprints for Multimedia. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 664–667 (2003)
25. Zhu, W., Xiong, Z., Zhang, Y.: Multiresolution Watermarking for Images and Video. IEEE Transactions on Circuits and Systems for Video Technology 9(4), 545–550 (1999)

Appendix: Selected Results

Table A1. The table of coverage and span for the Mean Algorithm (1) with a population of size 10. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	**5**	**7**	**3**	**5**	**7**	**3**	**5**	**7**
30	.467,.386	.376,.282	.312,.226	.472,.378	.378,.280	.321,.222	.481,.376	.373,.272	.319,.217
50	.703,.682	.594,.506	.514,.415	.719,.691	.596,.513	.514,.418	.722,.690	.601,.515	.522,.419
70	.869,1.07	.775,.804	.699,.664	.882,1.07	.788,.800	.709,.660	.891,1.07	.794,.803	.717,.663
80	.932,1.32	.851,.995	.790,.828	.936,1.32	.862,.995	.793,.828	.942,1.33	.866,1.00	.801,.827
90	.969,1.62	.921,1.24	.867,1.04	.972,1.65	.929,1.26	.879,1.05	.977,1.68	.936,1.28	.891,1.06
95	.983,1.98	.956,1.53	.923,1.29	.983,1.93	.959,1.49	.926,1.25	.988,1.96	.966,1.51	.933,1.27
99	.989,2.12	.968,1.65	.940,1.38	.991,2.19	.975,1.71	.952,1.44	.996,2.46	.986,1.92	.971,1.62

Table A2. The table of coverage and span for the Median Algorithm (2) with a population of 10. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	**5**	**7**	**3**	**5**	**7**	**3**	**5**	**7**
30	.458,.465	.376,.356	.318,.287	.461,.459	.350,.321	.281,.243	.466,.465	.244,.201	.255,.200
50	.698,.814	.610,.648	.544,.562	.698,.795	.634,.667	.571,.597	.657,.696	.657,.693	.645,.695
70	.872,1.28	.788,1.02	.713,.851	.881,1.27	.805,1.04	.717,.831	.886,1.25	.855,1.17	.649,.707
80	.932,1.60	.873,1.27	.814,1.08	.933,1.61	.883,1.26	.834,1.13	.943,1.69	.889,1.25	.886,1.26
90	.969,2.02	.936,1.61	.893,1.37	.978,2.01	.942,1.64	.900,1.36	.979,1.94	.966,1.83	.888,1.26
95	.987,2.47	.969,2.02	.945,1.72	.990,2.45	.971,1.94	.945,1.67	.992,2.78	.978,1.94	.968,1.89
99	.992,2.64	.978,2.17	.962,1.86	.996,2.81	.984,2.26	.971,1.94	.999,3.01	.996,2.82	.977,1.99

Table A3. The table of coverage and span of the Pairwise Comparison Algorithm (5) with a population of size 10. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	5	7	3	5	7	3	5	7
30	.596,.695	.502,.442	.511,.466	.605,.687	.512,.435	.525,.458	.650,.689	.524,.431	.528,.450
50	.833,1.22	.747,.788	.756,.822	.850,1.21	.761,.786	.771,.826	.884,1.24	.770,.793	.780,.825
70	.952,1.89	.902,1.25	.910,1.30	.967,1.88	.921,1.24	.925,1.30	.977,1.93	.928,1.25	.930,1.30
80	.983,2.37	.952,1.55	.961,1.62	.986,2.38	.964,1.55	.963,1.62	.983,2.27	.967,1.56	.972,1.62
90	.995,2.80	.982,1.98	.984,2.06	.998,2.93	.986,2.00	.987,2.08	.999,3.01	.989,2.04	.990,2.11
95	.998,2.95	.993,2.43	.993,2.49	.998,3.00	.994,2.41	.995,2.47	.999,3.01	.997,2.44	.998,2.51
99	.998,2.99	.996,2.60	.996,2.64	.999,3.01	.996,2.75	.997,2.78	.999,3.01	.999,3.00	.999,3.00

Table A4. The table of coverage and span for the Mean Algorithm (1) with a population of 15. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	5	7	3	5	7	3	5	7
30	.549,.428	.456,.330	.396,.277	.567,.429	.468,.330	.403,.278	.582,.430	.472,0.331	.407,0.276
50	.807,.748	.701,.576	.625,.484	.821,.749	.713,.577	.636,.485	.838,.752	.723,0.578	.641,.487
70	.943,1.14	.878,.880	.819,.741	.951,1.15	.887,.882	.827,.743	.958,1.15	.897,0.885	.837,.746
80	.983,1.41	.954,1.08	.923,.915	.987,1.41	.960,1.09	.930,.916	.989,1.42	.966,1.09	.937,.920
90	.992,1.77	.973,1.38	.951,1.16	.994,1.79	.978,1.39	.957,1.17	.996,1.80	.982,1.39	.962,1.18
95	.997,2.19	.989,1.71	.977,1.45	.998,2.10	.990,1.63	.978,1.38	.998,2.11	.992,1.64	.983,1.39
99	.997,2.18	.989,1.71	.977,1.45	.999,2.29	.994,1.79	.985,1.52	.999,2.44	.996,1.92	.992,1.62

Table A5. The table of coverage and span for the Median Algorithm (2) with a population of 15. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	5	7	3	5	7	3	5	7
30	.544,.472	.456,.370	.400,.306	.554,.464	.455,.356	.405,.299	.555,.469	.401,.287	.392,.280
50	.785,.826	.693,.657	.630,.553	.799,.835	.698,.653	.635,.566	.827,.884	.690,.618	.688,.616
70	.920,1.30	.866,1.03	.813,.877	.946,1.31	.890,1.03	.835,.888	.962,1.37	.882,.980	.877,.967
80	.969,1.62	.935,1.28	.895,1.10	.977,1.63	.941,1.29	.904,1.10	.982,1.68	.961,1.36	.890,1.01
90	.990,2.06	.974,1.64	.953,1.40	.994,2.08	.977,1.65	.960,1.40	.994,2.09	.987,1.77	.963,1.37
95	.997,2.54	.990,2.05	.979,1.75	.998,2.48	.991,1.97	.980,1.69	.9992,2.45	.993,1.90	.989,1.81
99	.998,2.73	.993,2.21	.987,1.90	.9996,2.88	.997,2.31	.991,1.97	1.00,3.40	.999,2.50	.998,2.31

Table A6. The table of coverage and span of the Pairwise Comparison Algorithm (5) with a population of 15. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	5	7	3	5	7	3	5	7
30	.682,.712	.583,.437	.589,.455	.695,.697	.593,.430	.603,.450	.704,.661	.607,.426	.608,.439
50	.901,1.23	.821,.779	.826,.807	.915,1.24	.840,.781	.846,.812	.934,1.27	.851,.782	.854,.810
70	.982,1.93	.951,1.23	.951,1.28	.988,1.93	.963,1.24	.964,1.27	.991,1.86	.971,1.23	.972,1.28
80	.995,2.39	.982,1.54	.981,1.59	.997,2.39	.987,1.54	.988,1.59	.999,2.47	.989,1.54	.989,1.59
90	.999,2.93	.994,1.96	.994,2.03	.9997,3.07	.996,1.98	.998,2.05	1.00,3.37	.997,2.01	.998,2.07
95	.9997,3.25	.998,2.47	.998,2.52	1.00,3.34	.999,2.39	.999,2.46	1.00,3.41	.9992,2.43	.9996,2.51
99	1.00,3.34	.999,2.66	.999,2.71	1.00,3.40	.9997,2.79	.9995,2.84	1.00,3.41	1.00,3.30	1.00,3.31

Table A7. The table of coverage and span for the Mean Algorithm (1) with a population of size 20. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	5	7	3	5	7	3	5	7
30	.606,.396	.500,.292	.430,.235	.622,.392	.508,.287	.434,.232	.630,.385	.510,.283	.434,.229
50	.853,.702	.754,.527	.667,.433	.859,.713	.758,.533	.671,.436	.873,.712	.770,.535	.685,.438
70	.961,1.11	.908,.840	.849,.695	.973,1.10	.918,.835	.866,.690	.978,1.11	.928,.835	.877,.692
80	.985,1.37	.959,1.04	.917,.866	.991,1.37	.963,1.04	.925,.865	.993,1.38	.968,1.04	.931,.865
90	.996,1.71	.983,1.31	.961,1.09	.998,1.73	.988,1.32	.969,1.10	.998,1.76	.992,1.34	.977,1.12
95	.999,2.10	.994,1.62	.982,1.35	.9991,2.04	.996,1.57	.988,1.31	.9994,2.07	.998,1.59	.992,1.33
99	.9996,2.25	.997,1.74	.990,1.46	.9997,2.34	.998,1.82	.994,1.52	1.00,2.63	.9995,2.04	.999,1.72

Table A8. The table of coverage and span for the Median Algorithm (2) with a population of 20. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	5	7	3	5	7	3	5	7
30	.602,.475	.508,.367	.446,.307	.612,.463	.515,.360	.455,.305	.613,.463	.491,.327	.486,.320
50	.834,.828	.754,.654	.695,.558	.848,.831	.759,.658	.693,.554	.876,.841	.741,.611	.719,.574
70	.957,1.31	.912,1.03	.870,.881	.969,1.31	.927,1.03	.884,.881	.978,1.39	.942,1.08	.881,.846
80	.988,1.63	.963,1.29	.934,1.10	.991,1.62	.968,1.29	.937,1.10	.994,1.66	.974,1.32	.951,1.12
90	.996,2.06	.987,1.64	.972,1.40	.998,2.08	.992,1.65	.979,1.41	.9993,2.17	.996,1.73	.987,1.43
95	.9992,2.57	.996,2.05	.990,1.76	.9996,2.48	.997,1.99	.991,1.69	.9998,2.60	.999,2.04	.997,1.76
99	1.00,2.77	.998,2.22	.997,1.90	1.00,2.88	.9992,2.31	.996,1.97	1.00,3.39	1.00,2.67	.9993,2.23

Table A9. The table of coverage and span of the Pairwise Comparison Algorithm (5) with a population of size 20. Span is indicated second and is expressed in terms of multiples of the standard deviation.

	50			100			1000		
	3	5	7	3	5	7	3	5	7
30	.722,.714	.625,.431	.639,.448	.752,.700	.645,.428	.656,.441	.777,.687	.661,.420	.659,.430
50	.935,1.24	.867,.768	.869,.793	.949,1.24	.883,.771	.882,.793	.957,1.24	.899,.771	.897,.794
70	.990,1.93	.969,1.22	.971,1.25	.996,1.94	.978,1.22	.977,1.26	.997,1.92	.985,1.22	.985,1.26
80	.998,2.39	.991,1.51	.992,1.57	.9993,2.40	.995,1.52	.994,1.57	.9995,2.38	.996,1.52	.996,1.57
90	.9998,2.98	.997,1.93	.998,2.00	1.00,3.07	.999,1.95	.999,2.01	1.00,3.05	.9994,1.97	.9997,2.04
95	1.00,3.39	.9993,2.44	.9992,2.50	1.00,3.48	.9994,2.35	.9998,2.42	1.00,3.66	.9997,2.38	.9998,2.46
99	1.00,3.52	.9994,2.64	.9997,2.70	1.00,3.63	.9998,2.77	1.00,2.82	1.00,3.66	1.00,3.22	1.00,3.27