PUTTING REPRODUCIBLE SIGNAL PROCESSING INTO PRACTICE: A CASE STUDY IN WATERMARKING

M. Barni[1], *F. Pérez-González*[2], *P. Comesaña*[2], *G. Bartoli*[1],

[1]Department of Information Engineering
University of Siena, Italy,
[2]Signal Processing and Communications Department, ETSI Telecom.
University of Vigo, Spain

ABSTRACT

In this paper the authors analyze how the description and presentation of results about an algorithm proposed in the literature should be modified in order to comply with the Reproducible Signal Processing paradigm. We describe the problems one is faced with, by specifically focusing on how the description of the algorithm should be improved with respect to the *classical* approach.

Keywords: Reproducible signal processing, verifiability of results, BNSA, watermarking.

1. INTRODUCTION

Reproducibility lies at the very core of the scientific method: an experiment or test is reproducible if it can be replicated by researchers independent from those that conducted it in the first place. When an experiment is successfully reproduced, chances that it be flawed are reduced. This is the reason why in certain scientific disciplines such as biology, physics or chemistry, great attention is paid not only to the experimental part, but further to experiment replication.

Unfortunately, even though the signal processing research community widely recognizes the importance of the experimental part of any work, very little has been done towards fully *reproducible signal processing*. In many papers, algorithms are claimed to be superior without providing enough empirical support, while in others data supplied in the experimental section are so vague or scarce that reproducibility is simply chimerical. Not to mention when the experiments are purposely crafted to give an empirical instance that allegedly “shows” the virtues of a certain algorithm.

In [1] Barni and Pérez-González have advocated the use of the scientific method in Signal Processing, with special emphasis on the reproducibility of the experimental results. In order to enable *reproducible signal processing* one should carefully describe the complete framework that yielded the experimental results; although this might seem a trivial task, in fact it is a thorny question, as all aspects of both the algorithm and the experimental setup should be characterized beyond ambiguity. This pinpoints the difficulties of reproducible signal processing and highlights the tremendous value of those research groups that devote time and resources to implement other researchers’ algorithms and repeat their experiments or carry new ones to test the validity of such algorithms. We strongly believe that this type of research should also be academically rewarding as it is in most scientific areas.

From a practical point of view, we have developed a methodology for conducting experimental work under the reproducible signal processing (RSP) paradigm. To test it, we chose a paper recently published by the research group in Vigo, who also provided a detailed additional description of the proposed algorithm. The research group in Siena then tried to reproduce the results shown in the original paper. Specifically, we considered a case taken from watermarking, an area in which we are active researchers. We have chosen a specific problem which has recently raised much interest in watermarking, namely, oracle-like attacks. These attacks exploit the binary answers of a watermarking detector in order to remove the watermark with a minimal distortion. Since they play with the sensitivity of the detector to slight changes of the input, they are sometimes also known as *sensitivity attacks*. Although sensitivity attacks have been known for some time, the recent publication of a *blind sensitivity algorithm* (BNSA) [2] has cast doubts on the security of most existing detectors, with immediate questions on their application to copyright protection and even fingerprinting scenarios. The impressive results allegedly achieved by this blind algorithm and their potential practical consequences cry out for the reproduction of the experiments in another research lab. This, the simplicity of the blind algorithm, and the fact that it has already been published constituted our main motivations to test our methodology with this example.

2. THE RSP FRAMEWORK

As mentioned in the introduction, the goal of BNSA is to remove the watermark from watermarked contents without knowing the watermarking algorithm. The BNSA algorithm is not a trivial one, so its implementation will have to deal with some problems, but it is neither too complicated, hence it seems an ideal candidate for testing the problems one has to face with to ensure results reproducibility. The most straightforward approach to RSP requires that the authors share the software implementing the proposed algorithms, and the data set use to test it, with readers. In this way the possibility of reproducing the results shown in the paper and/or testing them on different data sets is automatically achieved. At the same time, relying on software availability also presents some inherent drawbacks, that can be summarized as follows.

Software usability. Which format should be adopted for the shared software? Providing executable files would tie the software to a particular hardware, while at the same time source software needs to be compiled, raising compatibility issues. Last but not least, who

This work was partially funded by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT, and by the Italian Ministry of University and Research under FIRB project RBIN04AC9W.

is going to ensure that the software made available by the authors can be correctly compiled (run) on the most common platforms? Can reviewers be in charge of this time consuming, non-rewarding task?

Software readability. In order to be really useful, the shared software should be easy to read. In this way readers can check that the software really implements the algorithm described in the paper and, possibly, test it under different working conditions. Again, it is not clear who is going to check whether the software is properly commented and if it is a faithful implementation of the algorithms presented in the paper.

Licensing problems. Many instances may exist where the authors are not willing to share their software to protect their copyrights and/or to not break licensing agreements. This is the case, for instance, of algorithms implemented by relying on proprietary libraries.

It is evident that regardless of software availability, the algorithms and the experiments presented in the paper should be clearly described in such a way that any reader could re-implement them and re-obtain the same results. By keeping the above observations in mind, the framework we tested and experimented with regard to BNSA can be summarized as follows.

- A block diagram or a pseudo-language description of the proposed algorithm must be included in the paper; the description should be detailed enough to allow readers to re-implement the algorithm with no uncertainty.
- All the parameters needed to run the algorithm are clearly listed in a table and the values used in the experiments detailed.
- The data used to run the experiments are clearly defined or made available to readers and reviewers.

In the following section an RSP pseudo-code description of the BNSA algorithm made according to the above rules is given. Readers are encouraged to compare this description with the original one [2][1]. It goes without saying that any good paper should support the pseudo-code with a standard description of the algorithm and the rationale behind it. Noticeably, readers will find that in some points this paper does not fully comply with the RSP rules stated above. This was unavoidable, since writing an RSP-compliant paper would have required much more room than was available.

3. AN RSP DESCRIPTION OF BNSA

The research group in Vigo provided a pseudo-code description of the BNSA algorithm including the initialization procedure; as will be shown in the next sections, this description plays a major role in determining the performance of the algorithm. Such a pseudo-code is reported below. To save space the pseudo-code is poorly commented, it is our opinion, though, that some comments should be inserted here and there to link the pseudo-code to the overall description of the algorithm.

```
Function $\mathbf{z}$ =BNSA($\mathbf{y}$)
 1. Compute $\nu$ such that $\nu \cdot \mathbf{y}$ is outside the detection region, but close to the boundary:
    (a) $\nu_0 = 0, \nu_1 = 1$
    (b) While $(\nu_1 - \nu_0) > \epsilon_1$
        i. $\nu_2 = (\nu_0 + \nu_1)/2$
        ii. $\mathbf{y}_1 = \nu_2 \cdot \mathbf{y}$
        iii. If detect($\mathbf{y}_1$) $= 1$ then $\nu_1 = \nu_2$, else $\nu_0 = \nu_2$
    (c) $\mathbf{t}_1 = (\nu_0 - 1) \cdot \mathbf{y}$
    (d) $\gamma_1 = \text{energy}(\mathbf{t}_1)$
 2. $\mathbf{t}_2$ = zero-mean Gaussian vector with variance $\sigma_T^2$
 3. $\beta = \text{minNorm}(\mathbf{y}, \mathbf{t}_2)$
 4. $\gamma_2 = \text{energy}(\beta \cdot \mathbf{t}_2)$
 5. We choose the vector which minimizes distortion:
    If $\gamma_1 < \gamma_2$ then $\mathbf{t} = \mathbf{t}_1$, else $\mathbf{t} = \mathbf{t}_2$
 6. $\beta = \text{minNorm}(\mathbf{y}, \mathbf{t})$
 7. $\gamma_{start} = \text{energy}(\beta \cdot \mathbf{t})$
 8. Slightly modify each component of the vector $\mathbf{t}$:
    (a) $\mathbf{t}' = \mathbf{t}$
    (b) $\mathbf{t}'[i] = \mathbf{t}'[i] + \epsilon_2$
    (c) $\beta = \text{minNorm}(\mathbf{y}, \mathbf{t}')$
    (d) $\gamma[i] = \text{energy}(\beta \cdot \mathbf{t}')$
 9. Gradient estimation: $\hat{\nabla}[i] = (\gamma[i] - \gamma_{start})/\epsilon_2$
10. Look for a decreasing step-length:
    (a) $\xi = 10$
    (b) $\mathbf{t}_{new} = \mathbf{t} - \xi \cdot \hat{\nabla}$
    (c) $\beta = \text{minNorm}(\mathbf{y}, \mathbf{t}_{new})$
    (d) $\gamma_{step} = \text{energy}(\beta \cdot \mathbf{t}_{new})$
    (e) While $\gamma_{start} < \gamma_{step}$
        i. $\xi = 0.7 \cdot \xi$
        ii. $\mathbf{t}_{new} = \mathbf{t} - \xi \cdot \hat{\nabla}$
        iii. $\beta = \text{minNorm}(\mathbf{y}, \mathbf{t}_{new})$
        iv. $\gamma_{step} = \text{energy}(\beta \cdot \mathbf{t}_{new})$
11. The resulting signal is $\mathbf{z} = \mathbf{y} + \beta \cdot \mathbf{t}_{new}$
12. If $\mathbf{z}$ meets the desired quality criteria, then it is the solution.
    Otherwise the algorithm is iterated again from point 6 with $\mathbf{t} = \mathbf{t}_{new}$.
```

```
Function $\beta = \text{minNorm}(\mathbf{y}, \mathbf{t}_0)$.
It computes the minimum scaling factor $\beta$ such that $\beta\mathbf{t}_0 + \mathbf{y}$ is outside the detection region:
 1. Normalization of attacking vector: $\mathbf{t} = \mathbf{t}_0/||\mathbf{t}_0||$
 2. If detect$(\mathbf{y} + \mathbf{t}) = 0$ or detect$(\mathbf{y} - \mathbf{t}) = 0$,
    then $\mathbf{v}_{out1} = \mathbf{t}$ and $\mathbf{v}_{in1} = 0$
 3. If detect$(\mathbf{y} + \mathbf{t}) = 1$ and detect$(\mathbf{y} - \mathbf{t}) = 1$:
    (a) While detect$(\mathbf{y} + \mathbf{t}) = 1$ and detect$(\mathbf{y} - \mathbf{t}) = 1$,
        $\mathbf{t} = 2 \cdot \mathbf{t}$
    (b) $\mathbf{v}_{out1} = \mathbf{t}$ and $\mathbf{v}_{in1} = \mathbf{t}/2$
 4. If detect$(\mathbf{y} + \mathbf{t}) = 0$
    (a) $\mathbf{v}_{out} = \mathbf{v}_{out1}$ and $\mathbf{v}_{in} = \mathbf{v}_{in1}$
    (b) While $||\mathbf{v}_{out} - \mathbf{v}_{in}|| > \epsilon_3$:
```

[1] For sake of brevity we can not report the original description of the algorithm. We only want to stress out that though the paper that introduced BNSA is widely recognized as a good one according to the standard parameters used by the signal processing community, it was not detailed enough to allow the exact reproduction of the experimental results.

```
        i. $\mathbf{v}_{middle} = (\mathbf{v}_{out} + \mathbf{v}_{in})/2$
        ii. If detect$(\mathbf{y} + \mathbf{v}_{middle}) = 1$,
            then $\mathbf{v}_{in} = \mathbf{v}_{middle}$, else $\mathbf{v}_{out} = \mathbf{v}_{middle}$
    (c) $\mathbf{v}_{+} = \mathbf{v}_{out}$
5. If detect$(\mathbf{y} - \mathbf{t}) = 0$
    (a) $\mathbf{v}_{out} = -\mathbf{v}_{out1}$ and $\mathbf{v}_{in} = -\mathbf{v}_{in1}$
    (b) While $||\mathbf{v}_{out} - \mathbf{v}_{in}|| > \epsilon_3$:
        i. $\mathbf{v}_{middle} = (\mathbf{v}_{out} + \mathbf{v}_{in})/2$
        ii. If detect$(\mathbf{y} + \mathbf{v}_{middle}) = 1$,
            then $\mathbf{v}_{in} = \mathbf{v}_{middle}$, else $\mathbf{v}_{out} = \mathbf{v}_{middle}$
    (c) $\mathbf{v}_{-} = \mathbf{v}_{out}$
6. If detect$(\mathbf{y} + \mathbf{t}) = 1$, then $\mathbf{v} = \mathbf{v}_{-}$
7. If detect$(\mathbf{y} - \mathbf{t}) = 1$, then $\mathbf{v} = \mathbf{v}_{+}$
8. If detect$(\mathbf{y} + \mathbf{t}) = 0$ and detect$(\mathbf{y} - \mathbf{t}) = 0$:
    • If $||\mathbf{v}_{+}|| < ||\mathbf{v}_{-}||$, then $\mathbf{v} = \mathbf{v}_{+}$, else $\mathbf{v} = \mathbf{v}_{-}$
9. The minimum-normed scaling factor $\beta$ needed to obtain an
   un-watermarked signal when $\beta \cdot \mathbf{t}_0$ is added to $\mathbf{y}$ corresponds
   to the ratio between any component of $\mathbf{v}$ and $\mathbf{t}_0$, therefore
   $\beta = \mathbf{v}[i]/\mathbf{t}_0[i]$.
```

Parameter	ϵ_1	ϵ_2	ϵ_3	σ_T^2
Value	10^{-6}	10^{-3}	10^{-8}	10^{-4}

Table 1. Values of the parameters used in the pseudo-code.

4. DIFFICULTIES

In order to test if the description in Section 3 is detailed enough to reproduce the results given in [2], the research group in Siena implemented the BNSA algorithm by relying on the pseudo-code description. The description of the initialization procedure was particularly helpful, since the original paper did not give much information about it. This seems to be a recurrent problem in many papers, that tend to focus on the core part of the algorithms without giving enough details with regard to initialization and/or stop conditions.

The implementation of the BNSA algorithm did not raise any particular problem, hence proving the validity of the pseudo-code description. However, several ambiguities were present mainly related to the exact definition of the experimental conditions and the way the watermarking algorithms attacked by BNSA were implemented. A brief summary of the difficulties and ambiguities we had to be faced with are summarized below. Note that we were trying to reproduce the results reported in Figure 1 of the original paper [2].

- The pseudo-code description considers a version of BNSA where an approximation of the gradient of the objective function to be minimized is used, while in the original paper the minimization procedure also relied on the estimation of the Hessian. This misunderstanding was due to the fact that several versions of BNSA were used in [2] to obtain the results on synthetic data and on real images. During the experiments reported here the gradient-based version of BNSA was used for synthetic data, whereas in [2] a version using also a diagonal approximation of the Hessian was used.
- Despite the detailed pseudo-code a few ambiguities were still present with regard to initialization. How are singularities like $0/0$ treated by the detector? Moreover, a literal implementation of the algorithms described in the previous section may result in infinite initialization loops. For instance, in the SS$_{angle}$ case scaling the values of the marked sequence towards zero results in an infinite loop in step 1. How were these situations avoided in the original tests?
- The watermarking methods used for the tests, namely the spread spectrum watermarking with correlation detection (SS [3]), spread spectrum watermarking with correlation coefficient detection (SS$_{angle}$ [4]), JANIS ([5]), and spread spectrum watermarking for Generalized Gaussian hosts (GG [6]), were not explicitly described. To reproduce results we had to resort to the original papers, thus raising some interpretation problems. For instance, in [4] the correlation-coefficient detector is used in conjunction with multiplicative watermarking, whereas in [2] it was used for additive watermarking.
- A major parameter heavily impacting the performance of any watermarking scheme is the false detection probability, i.e. the probability that the watermark is found in a non-marked content. Though such a parameter was correctly supplied, no hint was given with regard to the way the false detection probability was estimated, hence rasing some ambiguities in the way the detection threshold is computed by relying on the false alarm rate. Given that several different equations can be used stemming from different statistical assumptions, a further interaction between the two research groups in Siena and Vigo was necessary. In particular we found that in the original implementation the statistical parameters determining the detection threshold were updated only once every BNSA run, whereas in the experiments carried out by the group in Siena, such parameters were updated each time the detector was run.
- The results of the original plot in [2] are averaged over 100 trials and given in dB. How was the average over the different experiments performed? By averaging the powers in natural units, or averaging powers in dBs? Even in this case, interaction between the two groups was necessary to solve the ambiguity.

4.1. Results

In this section we describe the results obtained by the Siena group and compare them with those originally obtained by the Vigo group and reported in [2]. Specifically we focused on the asymptotic behavior of the BNSA algorithm, hence some discrepancies between the initial part of the graphs are still visible (see figures below). The thorough analysis we carried out, however, permitted us to understand the reasons for such differences and provided us with very useful insights about the performance of the BNSA algorithm.

In Figures 1 and 2 the original and reproduced results are given respectively. Such figures report the average power of the attack necessary to remove the watermark from the host sequence as a function of the number of iterations of the BNSA algorithm. Since convergence is reached after a few iterations we plotted the results only until the $6^t h$ iteration. The results have been obtained by averaging (before passing into the dB domain) the attack power necessary to erase the watermark from 100 randomly generated sequences. Sequences were 2048 samples long. As specified in the original paper, the random sequences were normally distributed with zero mean and unitary variance. Only for the GG case the host sequence was gen-

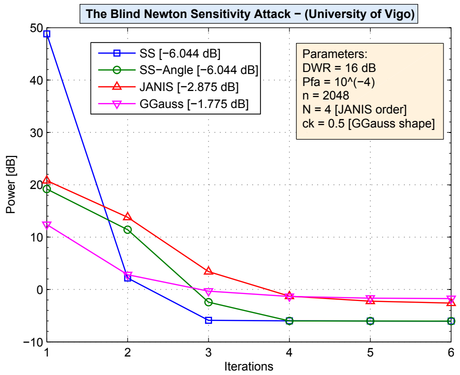

Fig. 1. Original results. The number in squared brackets reports the attacking power after convergence is reached.

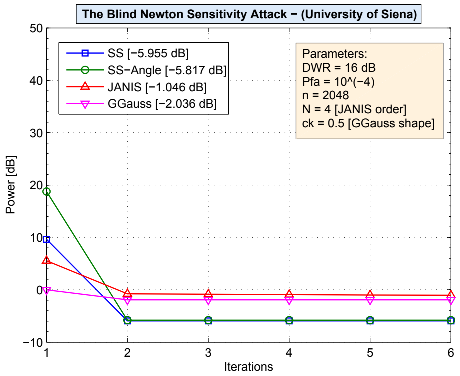

Fig. 2. Reproduced results. The number in squared brackets reports the attacking power after convergence is reached.

erated according to the generalized Gaussian distribution with shape parameter $c_k = 0.5$.

Upon inspection of the results, several differences among the curves immediately appear. When reproducing the results we found that the BNSA always converges in one step. This contrasts with the behavior reported in the original paper where in some cases power convergence required more steps. The explanation of this fact is that the results in Figure 2 were obtained by using only an approximation of the gradient of the objective function, without considering any approximation of the Hessian. As to the attack power reached after convergence, we found that the results obtained for the SS, the SS$_{angle}$ and the GG cases are virtually identical, the few observed differences being possibly due to statistical fluctuations (however this assumption should be verified theoretically). On the contrary, a significant difference is observed when attacking the JANIS system. To understand why, the Vigo group ran again its software by using only the gradient approximation and disregarding the Hessian. The results they obtained are much closer to those obtained by the Siena group, hence proving that the discrepancy between the original and the reproduced results is again due to the use of the gradient-based version of BNSA.

Another important difference can be noticed by looking at the starting point of the curves, i.e. by considering the output of the initialization procedure. In the end we found that the discrepancies were due to the implementation of the detection algorithms, specifically to the way and how often the statistical parameters necessary to fix the detection thresholds were estimated. The group in Siena estimated them each time the detector is run, whereas in the original implementation such parameters were estimated only once for each BNSA run. The impact of this different implementation is mainly visibile at the beginning when the attack noise is high (and the statistical parameters need to be refreshed often). On the other side at the right end of the plot, when the attack noise is low the different estimation strategy has a lower impact, hence justifying why we were able to obtain the same asymptotic results.

5. CONCLUSIONS

The main lessons we learned can be summarized as follows.

RSP is extremely insightful. We now know much more about the BNSA algorithm than we knew in advance, especially with regard to the impact of the initialization procedure and the particular approximation used to implement the gradient descent algorithm.

RSP relies on previous RSP. Ambiguities in the definition of previous algorithms (the GG or JANIS detectors in our case), are carried over to future uses. If a researcher is not being consistent with the RSP paradigm, he/she is making difficult its future application.

RSP is tough. We knew it, but possibly RSP is harder than expected. This strengthens our conviction that the birth of research groups expressly devoted to this kind of research should be encouraged. As already proposed in [1], some first steps into this direction include the reservation of a section of SP journals to RSP-compliant papers, and tightening the quality requirements applied to the experimental part of papers.

RSP is space consuming. Papers written according to the RSP paradigm are likely to be considerably longer than their classical counterpart (in fact we were not able to write a fully reproducible paper in the four pages allowed by the ICASSP format). Is RSP only for archival journals and not suited for conference proceedings?

6. REFERENCES

[1] M. Barni and F. Pérez-González, “Pushing science into signal processing,” *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 119–120, July 2005.

[2] P. Comesaña, L. Pérez-Freire, and F. Pérez-González, “The blind newton-sensitivity attack,” in *In Edward J. Delp III and Ping W. Wong, editors, Security, Steganography, and Watermarking of Multimedia Contents VIII*, S. Jos, CA, January 2006.

[3] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*, Marcel Dekker, 2004.

[4] 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.

[5] T. Furon, B. Macq, N. Hurley, and G. Silvestre, “JANIS: Just Another n-order Side-Informed Watermarking Scheme,” in *Proc. ICIP 2002*, Rochester, NY, USA, 22-25 September 2002.

[6] J. R. Hernandez, M. Amado, and F. Pérez-González, “DCT-domain watermarking techniques for still images: detector performance analysis and a new structure,” *IEEE Transactionson Image Processing*, vol. 9, no. 1, pp. 55–68, Jan. 2000.