BENCHMARKING OF STILL IMAGE WATERMARKING METHODS: PRINCIPLES AND STATE OF THE ART

N. Nikolaidis, V. Solachidis, A. Tefas, V. Arguriou, I. Pitas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece
e-mail: pitas@zeus.csd.auth.gr

Abstract – This paper summarizes the basic principles of still image watermarking benchmarking. A short description of existing benchmarking platforms is also provided.

INTRODUCTION

The development of digital services created new requirements for multimedia security and copyright protection techniques. Watermarking has emerged recently as an important copyright protection tool. Watermarking research evolved with a tremendous speed during the last years [1]. Numerous methods have been presented in the literature and several watermarking software packages have been developed. Up to now, performance evaluation has been carried out in a non-standardized way, with no concrete supporting evidence. With the watermarking technology entering into a more mature era, it is about time that a benchmarking methodology will be devised. This development would benefit both the watermarking technology suppliers, by allowing them to fine-tune their algorithms and providing indications on their position in the watermarking arena, and the technology users, by providing a systematic way of comparing existing solutions and picking the one that satisfies their needs in the best possible way.

This paper will try to address the major considerations that arise when designing a benchmarking system for copyright protection image watermarking methods and summarize the basic benchmarking principles. A brief review of four benchmarking platforms, pointing at the pros and cons of each system is also provided.

BENCHMARKING PRINCIPLES

Ideally, a benchmarking tool should have the ability to highlight the advantages and the weaknesses of the watermarking method under test and allow for easy and efficient method comparison. However, this is not an easy task because it involves examining a set of mutually dependent performance factors (algorithm complexity, decoding/detection performance, and perceptual quality). Thus, one cannot come up with a single figure of merit but rather with a vector of performance indices. An efficient benchmarking method should quantify and present the interactions among the various performance aspects, e.g., the relation between watermark robustness and perceptual quality.

A general consideration that one should bear in mind when dealing with watermarking benchmarking is that watermarking performance depends on the keys that will be used for embedding and detection, the messages that will be embedded and the host images. As a consequence, in order to obtain statistically valid results one should perform multiple trials with a significantly large number of different keys, messages and images of various sizes and

content. Furthermore, in order to ensure that results from different benchmarking trials will be comparable, a common set of parameters and inputs (images, keys, etc) should be used.

Since a major requirement for a watermark is to remain detectable in case of host signal alterations, an important aspect of a benchmarking system is the set of manipulations or attacks that it incorporates. This set should include all attacks and manipulations that the average user or an intelligent pirate can use. Furthermore, as new sophisticated attacks are developed, the benchmark should allow for hassle-free insertion of user-defined attacks.

Combination of results obtained for different images and attacks is sometimes necessary to cope with different performance measurements and come up with a more compact result representation. Such a combination can be based on a set of weights chosen so as to reflect the probability of occurrence of an attack or an image within a certain application scenario. Various levels of information compaction should be allowed.

The parameters that should be taken into account while judging the performance of a watermarking algorithm are the detection performance, the decoding performance, the perceptual quality of the watermarked image and the algorithm complexity. In the context of this paper we make the distinction between watermark detection and message decoding. The term watermark detection is used to denote the ability of the watermarking algorithm to declare the presence or absence of a watermark on an image. As soon as the algorithm declares the image to be watermarked the embedded message (if any) should be decoded. Thus, watermark detection and message decoding should be considered as two distinct steps that are performed in cascade, the message decoding step taking place only if a watermark has been found to reside in the image.

Detection Performance: Watermark detection can be considered as a hypothesis testing problem, the two hypotheses being *H0 : the image under test hosts the watermark under investigation* and *H1 : the image under test does not host the watermark under investigation.* Hypothesis H1 can be further divided into two sub-hypotheses: *H1a : the image under test is not watermarked,* and *H1b : the image under test hosts a different watermark.*

Thus, detection performance can be characterized by the false alarm probability P_{fa} i.e., the probability to detect a watermark in an image that is not watermarked or is watermarked by a different key than the one under investigation, and false rejection probability P_{fr} i.e., the probability of not detecting a watermark in an image that is indeed watermarked. In order to estimate these error probabilities one should conduct experiments involving detection of a watermark W in the following instances:

1. In images that host the watermark W (test related to hypothesis H0 and P_{fr})
2. In images that host a different watermark W' (test related to hypothesis H1b and P_{fa})
3. In images that host no watermark (test related to hypothesis H1a and P_{fa})

Evaluation of P_{fa} using detection with an erroneous watermark (test 2 above) might sometimes be preferable since it corresponds to the worst case scenario.

Watermark detectors can be of two different types. Hard decision detectors generate a binary output (watermark detected, watermark not detected) which results by comparing internally the test statistic of the corresponding hypothesis test against a decision threshold. Soft decision detectors provide as output the test statistic itself i.e., a real number that is related to detection reliability. In this case, thresholding in order to reach a binary decision is done in a separate, subsequent step.

In the case of hard decision detectors one can use the experimental data to evaluate a single P_{fr} , P_{fa} pair of values. In case of soft decision detectors, the probability distribution function of the detection statistic for both hypotheses H0 and H1b (or H1a) can be evaluated or approximated by an appropriate distribution model. Using these distributions one can evaluate P_{fr} , P_{fa} as a function of the detection threshold T and thus judge the performance of the algorithm under different operating conditions [4]. By plotting the probability of false alarm

versus the probability of false rejection the Receiver Operating Characteristic curve (ROC) results. The ROC curve provides a complete characterization of the system's detection performance and can be used for selecting the appropriate detection threshold. Using the ROC one can also evaluate P_{fa} for fixed P_{fr} , P_{fr} for fixed P_{fa}, and the equal error rate (EER) point, i.e., the point of the ROC where P_{fr} equals P_{fa}.

Decoding Performance: If the watermarking method supports message encryption, the decoding performance of an algorithm can be judged by evaluating the Bit Error Rate (BER) i.e. the mean number of erroneously decoded bits. Since message decoding is assumed to be performed only in case of successful detection, there is a close relation between the decoding and detection performance (false alarm and false rejection probabilities). This becomes obvious when judging the performance of an algorithm with a soft decision detector where one cannot obtain a single BER value but rather a plot of the BER versus P_{fa} or P_{fr}. Another important aspect of a watermarking algorithm that is closely related to its decoding performance is the algorithm payload which can be defined as the maximum number of bits that can be encoded in a fixed amount of data and decoded with a pre-specified BER or alternatively as the amount of data required to host a fixed number of bits so that it can be decoded with a pre-specified BER.

Perceptual Quality: A benchmarking tool should be able to check whether a watermarking method generates watermarks that are imperceptible to the human visual system. If an automated benchmarking platform is to be constructed, perceptual quality should be measured with objective metrics and not by subjective tests, despite the advantages of the latter. Obviously, quantitative measures that correlate better with the image quality (as perceived by human observers) than the widely used SNR and PSNR metrics should be used. However, no globally agreeable, efficient visual quality metric currently exists.

Complexity: Both the complexity of the watermark embedding and the watermark detection/message decoding procedures should be evaluated. The easiest but not the most appropriate way to measure complexity is by recording execution time in a fixed hardware/software suite. Other means of measuring complexity, i.e., by theoretical evaluation of the number of required operations are far more appropriate but impossible to implement within the context of an automated benchmark.

Obviously, the necessity to deal with a multidimensional performance space makes watermarking performance characterization, method comparison and result presentation a complicated task. The requirement for performance evaluation over various attacks and attack parameters (e.g. for various jpeg compression factors) adds one more complexity factor to the problem. A reasonable way to deal with the situation is to derive multiple plots depicting the relation of the various performance factors. Such plots could include detection performance versus perceptual quality, decoding performance versus perceptual quality, detection performance versus attack strength, etc. Since the only aspects of a system that one can have control on are the perceptual quality and the attack strength, fixing these parameters to values typical for the application and measuring the system performance (and proceed to comparisons) with respect to the remaining parameters can be also a way to deal with the multidimensionality problem.

EXISTING BENCHMARKING PLATFORMS

Stirmark (http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/index.html) is the first benchmarking software that has been developed [1,2]. In its current version (v 3.1) Stirmark involves the following attacks: sharpening, GIF and JPEG compression, scaling, cropping, shearing, rotation, column and line removal, flipping and `Stirmark' attack (a combination of slight geometric and intensity distortions). Despite its "historical" importance and the fact that

the accompanying publications raise a number of important benchmarking issues, the current implementation of Stirmark can be characterized as an *attack machine* rather than a benchmark since it lacks the automation and batch processing abilities that a benchmark should posses. The user should provide a number of watermarked images and a "detection" executable that outputs either 1 or 0 (not detected/detected) according to a user-defined rule (e.g. if the application supports message embedding one can construct a detector that reports "watermark detected" when 80% of the message bits have been correctly decoded). Stirmark applies a number of attacks (one at a time) in every watermarked image and calls the detection routine. Thus no option for automatic execution of trials involving different keys or messages is provided (the user can do such trials only by providing images watermarked with different keys). Furthermore, Stirmark fails to address the fact that watermark detection and message decoding are two inter-linked but distinct operations. Detection results can be inserted in a spreadsheet provided along with the software so that successful detection averages (over different images or different attacks) can be calculated. Thus automatic evaluation of performance statistics and performance curves is addressed in a rather primitive way, setting the focus on correct detection probability ($1-P_{fa}$) with no mention on the equally important false alarm probability. Finally no hint on embedding/detection complexity (e.g. by evaluating mean execution time) is provided. A new, drastically improved version of Stirmark is currently under development.

Checkmark (http://watermarking.unige.ch/Checkmark/) [3] can be considered as a successor of Stirmark. Apart from Stirmark attacks, Checkmark incorporates an important number of new attacks that include wavelet compression (jpeg 2000), projective transformations, modeling of video distortions, warping, copy attack, template removal attack, denoising, non-linear line removal, collage attack, down/up sampling, dithering and thresholding. Being a open-source Matlab application, Checkmark allows for the inclusion of new attacks. Furthermore, Checkmark implements new objective quality metrics namely the weighted PSNR and the so-called Watson metric and provides a number of "application templates" i.e. lists of attacks related to a certain application. In the near future, application templates will support application-specific weighted averaging of attacks. Despite the major improvements, the basic operating principles of Checkmark are very similar with those of Stirmark: the user should provide a number of watermarked images and a "detection" executable with a user-defined detection rule. Attacks described in the selected application template are applied in every watermarked image and the detection routine is being called. Thus Checkmark inherits many of Stirmark drawbacks (no option for automatic execution of multiple trials, no evaluation of the false alarm probability, failure to address watermark detection and message decoding separately, no complexity evaluation).

Optimark (http://poseidon.csd.auth.gr/optimark/) [4] is the latest arrival in the benchmarking software scene. It features a graphical user interface and incorporates the same attacks with Stirmark (cascades of attacks are also possible). The user should supply an embedding and a detection/decoding executable. Optimark supports both hard and soft decision detectors. The user selects the test images, the set of PSNR values that the embedding software will operate on (an option for the automatic calculation of the embedding strength that leads to the selected image quality is provided), the range of keys and messages that will be used and the attacks that will be performed. Then Optimark launches an automatic execution of multiple trials using the images, attacks, keys and messages selected by the user. Detection using both correct and erroneous keys (necessary for evaluating false alarms) is performed. Message decoding performance is evaluated separately from watermark detection. Raw results are automatically processed by the benchmark in order to provide a number of performance

metrics and plots (in html format) that include mean embedding and detection time, ROCs, Equal Error Rate, P_{fa} for a user defined, P_{fr}, P_{fr} for a user defined, P_{fa}, (for soft decision detectors), P_{fa} and P_{fr} (for hard decision detectors), Bit Error Rate either as a single value or as a function of P_{fa} , percentage of perfectly decoded messages, payload (for algorithms that support message encoding). Evaluation of the algorithm breakdown limit for a certain attack and a certain performance criterion, i.e., evaluation of the attack severity where algorithm performance exceeds (or falls below) a certain limit is provided. Result can be summarized in multiple levels using a set of user defined weights on the selected attacks and images. Options for both user defined and preset benchmarking sessions are also available. The main drawbacks of Optimark are the lack of expandability with respect to attacks and the use of a simple perceptual quality metric. These drawbacks will be dealt with in the next version of the platform.

Certimark R&D project (http://www.certimark.org) funded by the European Union is currently developing a benchmarking platform using a client-server, web-based structure. Its main characteristic will be its open architecture that will allow for easy integration of new functionalities. The software will include the following features:
- A flexible interface to plug-in watermark embedding and detection software. The user will supply an embedding/detection dynamic link library (dll) and an xml file describing the watermarking parameters (values of key, embedding strength, etc).
- An extensive list of attacks and a flexible interface to plug-in new attack software using a dll xml file pair.
- A flexible interface to plug-in objective perceptual quality metrics.
- A definition of a format for raw test results, to enable development and use of different results processing and analysis tools.

CONCLUSIONS

Despite the recent advances in the field of benchmarking of copyright protection still image watermarking methods the area is still in its early stages of development. A number of practical and theoretical issues need to be solved, hopefully within the near future.

ACKNOWLEDGEMENTS

This work has been partially supported by EU Project IST-1999-10987 CERTIMARK.

References

[1] Stefan Katzenbeisser and Fabien A. P. Petitcolas (eds). *Information hiding techniques for steganography and digital watermarking*. Artech House Books, ISBN 1-58053- 035-4, December 1999.
[2] Fabien A. P. Petitcolas. Watermarking schemes evaluation. *I.E.E.E. Signal Processing*, vol. 17, no. 5, pp. 58–64, September 2000
[3] Shelby Pereira, Sviatoslav Voloshynovskiy, Maribel Madueño, Stéphane Marchand-Maillet and Thierry Pun, Second generation benchmarking and application oriented evaluation, In *Information Hiding Workshop III*, Pittsburgh, PA, USA, April 2001
[4] V. Solachidis, A. Tefas, N. Nikolaidis, S. Tsekeridou, A. Nikolaidis, I.Pitas, ``A benchmarking protocol for watermarking methods'', 2001 *IEEE Int. Conf. on Image Processing (ICIP'01)*, pp. 1023-1026, Thessaloniki, Greece, 7-10 October, 2001