Chapter 1: INTRODUCTION

In recent times, internet is being increasingly used as the platform for distribution of digital multimedia content. The inherent flexibility of Internet facilitates users to transact with one another to create, distribute, store, peruse, subscribe, enhance, modify and trade digital content in various forms like text documents, databases, e-books, still images, audio, video, computer software and games.

The use of an open medium like Internet gives rise to concerns about protection and enforcement of intellectual property rights of the digital content involved in the transaction. In addition, unauthorized replication and manipulation of digital content is relatively trivial and can be done using inexpensive tools, unlike the traditional analog multimedia content. The protection and enforcement of intellectual property rights for digital media has become an important issue. In recent years, the research community has seen much activity in the area of digital watermarking as an additional tool in protecting digital content.

1.1: CRYPTOGRAPHY

Literally, Cryptography is the art of writing in ‘ciphers’; or it is a method of secret communication. In cryptography, the contents of secret message are concealed and only the sender and the receiver of the secret message know the process of extracting the concealed information. Apparently, others can’t easily detect what message is being conveyed. Cryptography is an effective solution to the distribution problem, but in most instances has to be tied to specialized and costly hardware to create tamper-proof devices that avoid direct access to data in digital format. Moreover, most cryptographic protocols are concerned with secured communications instead of ulterior copyright infringements. For instance, access control in set-top-boxes used for digital television demodulation and decoding succeed in avoiding unauthorized access to programs that are being broadcast in scrambled form but fail in precluding further storage and illegal dissemination actions.

1.2: STEGANOGRAPHY

Steganography is a technique for concealed communication. In contrast to cryptography where the content of a communicated message is secret, in steganography the very existence of the message that is communicated is a secret and its presence is known only by parties involved in the communication. Steganography is technique where a secret message is hidden within another unrelated message and then communicated to the other party. Some of the techniques of steganography like use of invisible ink, word spacing patterns in printed documents, coding messages in music compositions, etc., have been used by military intelligence since the times of ancient Greek civilization. In steganography, usually the message itself is of value and must be protected through clever hiding techniques and the "vessel" for hiding the message is not of value. In watermarking, the effective coupling of message to the "vessel", which is the digital content, is of value and the protection of the content is crucial.

Fragile Invisible Steganography Algorithm "Manipulating LSBs"

Goal: To hide image-B in image-A

- Replace one LSB of image-A with the corresponding one MSB of image-B
- Replace two LSBs of image-A with the corresponding two MSBs of image-B
- Compare the results of the two manipulations with the original image-A
- In general, replace 'k' LSBs of image-A with the corresponding 'k' MSBs of image-B, and observe the results.

1.3: DIGITAL WATERMARKING

Watermarking is descendent of steganography which has been in existence for at least a few hundred years. Watermarking is a special technique of steganography where one message is embedded in another and the two messages are related to each other in some way. The most common examples of watermarking are the presence of specific patterns in currency notes which are visible only when the note is held to light and logos in the background of printed text documents. The watermarking techniques prevent forgery and unauthorized replication of physical objects.

Digital watermarking is similar to watermarking physical objects except that the watermarking technique is used for digital content instead of physical objects. In digital watermarking a low-energy signal is imperceptibly embedded in another signal. The low-energy signal is called watermark and it depicts some metadata, like security or rights information about the main signal. The main signal in which the watermark is embedded is referred to as cover signal since it covers the watermark. The cover signal is generally a still image, audio clip, video sequence or a text document in digital format.

Unlike encryption, which does not provide a way to examine the original data in its protected form, the watermark remains in the content in its original form and does not prevent a user from listening to, viewing, examining, or manipulating the content. Also, unlike the idea of steganography, where the method of hiding the message may be secret and the message itself is secret, in watermarking, typically the watermark embedding process is known and the message (except for the use of an optional secret key) does not have to be secret.

Watermarking is the direct embedding of additional information into the original content or host signal. Ideally, there should be no perceptible difference between the watermarked and original signal and the watermark should be difficult to remove or alter without damaging the host signal. In some instances, the amount of information that can be hidden and detected reliably is important. It is easy to see that the requirements of imperceptibility, robustness, and capacity conflict with each other. For instance, a straightforward way to provide an imperceptible watermark is to embed the watermark signal into the *perceptually insignificant* portion of the host data. However, this makes the watermark vulnerable to attack because it is fairly easy to remove or alter the watermark without affecting the host signal.

To provide a robust watermark, a good strategy is to embed the watermark signal into the significant portion of the host signal. This portion of the host data is highly sensitive to alterations, however, and may produce very audible or visible distortions in the host data. Applications for digital watermarking include copyright protection, fingerprinting, authentication, copy control, tamper detection, and data hiding applications such as broadcast monitoring. Watermarking algorithms have been developed for audio, still images, video, graphics, and text.

Visible watermarks which do not interfere with the intelligibility of the host signal have also been developed; while transparent watermarking techniques can be *fragile*, *robust,* or *semi fragile*. Fragile watermarks do not survive lossy transformations to the original host signal and their purpose is tamper detection of the original signal.

There are many effective ways to insert a fragile watermark into digital content while preserving the imperceptibility requirement. Placing the watermark information into the *perceptually insignificant* portions of the data guarantees imperceptibility and provides fragile marking capabilities. For instance, early watermark techniques for still image data propose inserting watermark information into the least significant bits of the pixel values. This results in an imperceptible mark which can detect lossy transformations performed on the watermarked content. For security applications and copyright protection, robust watermarking techniques have been developed. Here the technical challenge is to provide transparency and robustness which are conflicting requirements.

Ideally, an effective, robust watermarking scheme provides a mark that can only be removed when the original content is destroyed as well. The degree of robustness and distortion necessary to alter the value of the original content can vary for different applications. Typically, many of the applications for copyright protection involve relatively high quality original content and the imperceptibility criterion is critical for such applications. In order for a watermarking technique to be robust, the watermark should be embedded in the *perceptually significant* portion of the data.

Some typical distortions or attacks that digital watermarking schemes are expected to survive include re-sampling, rescaling, compression, linear and nonlinear filtering, additive noise, A/D and D/A conversion, and trans-coding. Applications for robust watermarking include copyright protection where each copy gets a unique watermark (commonly referred to as a fingerprint) to identify the end-user so that tracing is possible for cases of illegal use; authentication, where the watermark can represent a signature and copy control for digital recording devices. Within the class of robust watermarking techniques there are several different constraints on encoder and decoder design which depends on the particular application.

Semi-fragile watermarking techniques differentiate between lossy transformations that are "information preserving" and lossy transformations which are "information altering." Lossy transformations include any signal processing step that alters the original signal values and is not invertible. For example, in authentication applications it may be desirable to have a watermark that can distinguish between a lossy transformation such as compression which does not alter the integrity of the content and an alteration which does alter the integrity, such as manipulating or replacing objects within the scene.

There should be no perceptible difference between the watermarked and original signal, and the watermark should be difficult to remove or alter without damaging the host signal.

1.4: APPLICATIONS OF DIGITAL WATERMARKING

Let us look upon some of the scenarios where watermarking is being already used as well as other potential applications. The list given here is by no means complete and intends to give a perspective of the broad range of possibilities that digital watermarking opens.

1.4.1: Image Watermarking

Many techniques have been developed for the watermarking of still image data. For grey-level or color-image watermarking, watermark embedding techniques are designed to insert the watermark directly into the original image data, such as the luminance or color components or into some transformed version of the original data to take advantage of perceptual properties or robustness to particular signal manipulations. Requirements for image watermarking include imperceptibility, robustness to common signal processing operations, and capacity. Common signal processing operations which the watermark should survive include compression (such as JPEG), filtering, rescaling, cropping, A/D and D/A conversion, geometric distortions, and additive noise. Capacity refers to the amount of information (or payload) that can be hidden in the host image and detected reliably under normal operating conditions. Many of the watermarking techniques are additive, where the watermark signal is added directly to the host signal or transformed host signal. The watermark may be scaled appropriately to minimize noticeable distortions to the host. Perceptual models may be used to determine and adapt the watermark scale factor appropriately to the host data. The watermark itself is a function of the watermark information, a secret or public key and perhaps the original host data. Some examples of watermark information include a binary sequence representing a serial number or credit card number, a logo, a picture, or a signature.

Many of the current watermarking techniques insert one bit of information over many pixels or transform coefficients and use classical detection schemes to recover the watermark information. These types of watermarking techniques are usually referred to as **spread-spectrum approach**, due to their similarity to spread-spectrum communication systems. For still image watermarking, watermark embedding is applied directly to the pixel values in the spatial domain or to transform coefficients in a transform domain such as the discrete cosine transform (DCT) or discrete wavelet transform (DWT). Watermark detection usually consists of some preprocessing step (which may include removal of the original host signal if it is available for detection) followed by a correlation operator.

Some of the earliest techniques embed m-sequences into the least significant bit (LSB) of the data to provide an effective transparent embedding technique. Random M-sequences are chosen due to their good correlation properties so that a correlation operation can be used for watermark detection. Furthermore, these techniques are computationally inexpensive to implement. In which we reshape the m-sequence into two-dimensional watermark blocks which are added and detected on a block-by-block basis.

1.4.2: Video Watermarking

In this case most considerations made in previous sections hold. However, now the temporal axis can be exploited to increase the redundancy of the watermark. As in the still images case, watermarks can be created either in the spatial or in the DCT domains. In the latter, the results can be directly extrapolated to MPEG-2 sequences, although different actions must +3.be taken for I, P and B frames. Note that perhaps the set of attacks that can be performed intentionally is not smaller but definitely more expensive than for still images.

1.4.3: Audio Watermarking

Again, previous considerations are valid. In this case, time and frequency masking properties of the human ear are used to conceal the watermark and make it inaudible. The greatest difficulty lies in synchronizing the watermark and the watermarked audio file, but techniques that overcome this problem have been proposed.

1.4.4: Hardware/Software Watermarking

This is a good paradigm that allows us to understand how almost every kind of data can be copyright protected. If one is able to find two different ways of expressing the same information, then one bit of information can be concealed, something that can be easily generalized to any number of bits. This is why it is generally said that a perfect compression scheme does not leave room for watermarking. In the hardware context, Boolean equivalences can be exploited to yield instances that use different types of gates and that can be addressed by the hidden information bits. Software can be also protected not only by finding equivalences between instructions, variable names, or memory addresses, but also by altering the order of non-critical instructions. All this can be accomplished at compiler level.

1.4.5: Text Watermarking

This problem, which in fact was one of the first that was studied within the information hiding area can be solved at two levels. At the printout level, information can be encoded in the way the text lines or words are separated (this facilitates the survival of the watermark even to photocopying). At the semantic level (necessary when raw text files are provided), equivalences between words or expressions can be used, although special care has to be taken not to destruct the possible intention of the author.

1.4.6: Labeling

The hidden message could also contain labels that allow for example to annotate images or audio. Of course, the annotation may also been included in a separate file, but with watermarking it results more difficult to destroy or loose this label, since it becomes closely tied to the object that annotates. This is especially useful in medical applications since it prevents dangerous errors.

1.4.7: Fingerprinting

This is similar to the previous application and allows acquisition devices (such as video cameras, audio recorders, etc) to insert information about the specific device (e.g., an ID number) and date of creation. This can also be done with conventional digital signature techniques but with watermarking it becomes considerably more difficult to excise or alter the signature. Some digital cameras already include this feature.

1.4.8: Authentication

This is a variant of the previous application, in an area where cryptographic techniques have already made their way. However, there are two significant benefits that arise from using watermarking: first, as in the previous case, the signature becomes embedded in the message, second, it is possible to create ‘soft authentication’ algorithms that offer a multi-valued ‘perceptual closeness’ measure that accounts for different unintentional transformations that the data may have suffered (an example is image compression with different levels), instead of the classical yes/no answer given by cryptography-based authentication. Unfortunately, the major drawback of watermarking-based authentication is the lack of public key algorithms that force either to put secret keys in risk or to resort to trusted parties.

1.4.9: Copy and Playback Control

The message carried by the watermark may also contain information regarding copy and display permissions. Then, a secure module can be added in copy or playback equipment to automatically extract this permission information and block further processing if required. In order to be effective, this protection approach requires agreements between content providers and consumer electronics manufacturers to introduce compliant watermark detectors in their video players and recorders. This approach is being taken in Digital Video Disc (DVD).

Chapter 2: DIGITAL IMAGE WATERMARKING

2.1: CHARACTERISTIC FEATURES OF WATERMARKING

As mentioned earlier, digital watermarking techniques are useful for embedding metadata in multimedia content. There are alternate mechanisms like using the header of a digital file to store meta-information. However, for inserting visible marks in images & video and for adding information about audio at the beginning or end of the audio clip etc. the digital watermarking technique is appealing, since it provides following main features and does not require out-of-band data as in other mechanisms.

2.1.1: Imperceptibility

The embedded watermarks are imperceptible both perceptually as well as statistically and do not alter the aesthetics of the multimedia content that is watermarked. The watermarks do not create visible artifacts in still images, alter the bit-rate of video or introduce audible frequencies in audio signals. The watermark should be perceptually invisible, or its presence should not interfere with the work being protected.

2.1.2: Robustness

Depending on the application, the digital watermarking technique can support different levels of robustness against changes made to the watermarked content. If digital watermarking is used for ownership identification, then the watermark has to be robust against any modifications. The watermarks should not get degraded or destroyed as a result of unintentional or malicious signal and geometric distortions like analog-to-digital conversion, digital-to-analog conversion, cropping, re-sampling, rotation, dithering, quantization, scaling and compression of the content. On the other hand, if digital watermarking is used for content authentication, the watermarks should be fragile, i.e., the watermarks should get destroyed whenever the content is modified.

The watermark must be difficult (hopefully impossible) to remove. If only partial knowledge is available (for example, the exact location of the watermark in an image is unknown), then attempts to remove or destroy a watermark should result in severe degradation in fidelity before the watermark is lost. In particular, the watermark should be robust in the following areas:

- **Inseparability -** After the digital content is embedded with watermark, separating the content from the watermark to retrieve the original content is not possible.
- **Common Signal Processing -** The watermark should still be retrievable even if common signal processing operations are applied to the data. These include, digital-to-analog and analog-to-digital conversion, re-sampling, re-quantization (including dithering and recompression), and common signal enhancements to image contrast and color, or audio bass and treble, for example.
- **Common Geometric Distortions -** Watermarks in image and video data should also be immune from geometric image operations such as rotation, translation, cropping and scaling.
- **Subterfuge Attacks (Collusion and Forgery) -** In addition, the watermark should be robust to collusion by multiple individuals who each possess a watermarked copy of the data. That is, the watermark should be robust to combining copies of the same data set to destroy the watermarks. Further, if a digital watermark is to be used in litigation, it must be impossible for colluders to combine their images to generate a different valid watermark with the intention of framing a third party.
- **Universality -** The same digital watermarking algorithm should apply to all three media under consideration. This is potentially helpful in the watermarking of multimedia products. Also, this feature is conducive to implementation of audio and image/video watermarking algorithms on common hardware.
- **Unambiguousness -** Retrieval of the watermark should unambiguously identify the owner. Furthermore, the accuracy of owner identification should degrade gracefully in the face of attack.

2.1.3: Security

The digital watermarking techniques prevent unauthorized users from detecting and modifying the watermark embedded in the cover signal. Watermark keys ensure that only authorized users are able to detect/modify the watermark. Finally, the watermark should *withstand multiple watermarking* to facilitate traitor tracing.

In general, a digital watermark should have several different properties. The most important are imperceptibility, robustness and security. Imperceptibility means that the watermarked data should be perceptually equivalent to the original, un-watermarked data. In some applications, the watermark may be perceptible as long as it is not annoying or obtrusive; however, many applications require that the watermark be imperceptible. Security means that unauthorized parties should not be able to detect or manipulate the watermark. Cryptographic methods are typically employed to make watermarks secure. Finally, robustness means that, given the watermarked data, one should not be able to make the watermark undetectable without also destroying the value or usefulness of the data.

Another characteristic of a watermarking scheme is whether or not the original data is available during detection. In some schemes [1], the watermark detector has access to the original data. Hence, interference from the original can presumably be eliminated. Blind schemes do not have the luxury of using the original during watermark detection. They typically apply some pre-processing to the received data to suppress interference from the original.

2.2: DESIGN CONSIDERATIONS AND REQUIREMENTS

Requirements and design of watermarking techniques are impacted by the different types of content in two major ways: imperceptibility and robustness requirements.

The first challenge is designing a watermark embedding algorithm which provides an imperceptible mark, that is, one which does not noticeably degrade the original host signal. Ideally, the marking algorithm should be adapted by using perceptual models appropriate for the different media types. The perceptual models used for representations of continuous tone images are not appropriate for text or graphics.

The other factor for designing watermarking schemes for multimedia is the type of degradations that the watermark is expected to survive and system requirements for media specific applications. For instance, it may be desirable for a still image watermarking technique to be able to survive JPEG compression and photocopying while for some video watermarking applications, it may be important to do watermark embedding and detection in real time on a compressed bit stream.

Moreover, the type of manipulations and the attacker expected computational power heavily depend on the application. Watermarking, like cryptography, also uses secret keys to map information to owners, although the way this mapping is actually performed considerably differs from what is done in cryptography, mainly because the watermarked object should keep its intelligibility. In most watermarking applications embedment of additional information is necessary. This information includes identifiers of the owner, recipient and/or distributor, transaction dates, serial numbers, etc. which play a crucial role in adding value to watermarking products.

2.3: DISTORTIONS AND ATTACKS

In practice, a watermarked object may be altered either on purpose or accidentally, so the watermarking system should still be able to detect and extract the watermark. Obviously, the distortions are limited to those that do not produce excessive degradations, since otherwise the transformed object would be unusable. These distortions also introduce a degradation on the performance of the system. For intentional attacks, the goal of the attacker is to maximize the reduction in these probabilities while minimizing the impact that his/her transformation produces on the object; this has to be done without knowing the value of the secret key used in the watermarking insertion process, which is where all the security of the algorithm lies. Following are some of the best known attacks. Some of them may be intentional or unintentional, depending on the application.

2.3.1: Additive Noise

This may stem in certain applications from the use of D/A and A/D converters or from transmission errors. However, an attacker may introduce perceptually shaped noise (thus, imperceptible) with the maximum unnoticeable power. This will typically force to increase the threshold at which the correlation detector works.

2.3.2: Filtering

Low-pass filtering, for instance, does not introduce considerable degradation in watermarked images or audio, but can dramatically affect the performance, since spread-spectrum-like watermarks have non negligible high-frequency spectral contents.

2.3.3: Cropping

This is a very common attack since in many cases the attacker is interested in a small portion of the watermarked object, such as parts of a certain picture or frames of a video sequence. With this in mind, in order to survive, the watermark needs to be spread over the dimensions where this attack takes place.

2.3.4: Compression

This is generally an unintentional attack which appears very often in multimedia applications. Practically all the audio, video and images that are currently being distributed via Internet have been compressed. If the watermark is required to resist different levels of compression, it is usually advisable to perform the watermark insertion task in the same domain where the compression takes place. For instance, DCT-domain image watermarking is more robust to JPEG compression than spatial-domain watermarking.

2.3.5: Rotation and Scaling

This has been the true battle-horse of digital watermarking, especially because of its success with still images. Correlation based detection and extraction fail when rotation or scaling is performed on the watermarked image because the embedded watermark and the locally generated version do not share the same spatial pattern anymore. Obviously, it would be possible to do exhaustive search on different rotation angles and scaling factors until a correlation peak is found, but this is prohibitively complex. Estimating the two parameters becomes simple when the original image is present, but, although the problem resembles synchronization for digital communications, the techniques applied there fail loudly.

2.3.6: Statistical Averaging

An attacker may try to estimate the watermark and then 'un-watermark' the object by subtracting the estimate. This is dangerous if the watermark does not depend substantially on the data. Note that with different watermarked objects it would be possible to improve the estimate by simple averaging. This is a good reason for using perceptual masks to create the watermark.

2.3.7: Attacks at Other Levels

There are a number of attacks that are directed to the way the watermark is manipulated. For instance, it is possible to circumvent copy control mechanisms discussed below by super-scrambling data so that the watermark is lost or to deceive web crawlers searching for certain watermarks by creating a presentation layer that alters the way data are ordered.

2.4: PREVIOUS WORK

Several previous digital watermarking methods have been proposed. Tanaka *et al.* [9] describe several watermarking schemes that rely on embedding watermarks that resemble quantization noise. Their ideas hinge on the notion that quantization noise is typically imperceptible to viewers. Their first scheme injects a watermark into an image by using a predetermined data stream to guide level selection in a predictive quantizer. The data stream is chosen so that the resulting image looks like quantization noise. A variation on this scheme is also presented, where a watermark in the form of a dithering matrix is used to dither an image in a certain way. There are several drawbacks to these schemes. The most important is that they are susceptible to signal processing, especially re-quantization, and geometric attacks such as cropping. Furthermore, they degrade an image in the same way that predictive coding and dithering can.

Tanaka *et al.*[9] also propose a watermarking method for "color-scaled picture and video sequences". This method applies the same signal transform as the Joint Photographers Expert Group (JPEG) (discrete cosine transform of 8X8 sub-blocks of an image) and embeds a watermark in the coefficient quantization module. While being compatible with existing transform coders, this scheme may be susceptible to re-quantization and filtering and is equivalent to coding the watermark in the LSB's of the transform coefficients.

Koch, Rindfrey, and Zhao [7] propose two general methods for watermarking images. The first method breaks up an image into 8X8 blocks and computes the discrete cosine transform (DCT) of each of these blocks. A pseudorandom subset of the blocks is chosen, then, in each such block, a triple of frequencies is selected from one of 18 predetermined triples and modified so that their relative strengths encode a one or zero value. The 18 possible triples are composed by selection of three out of eight predetermined frequencies within the 8X8 DCT block. The choice of the eight frequencies to be altered within the DCT block is based on a belief that the "middle frequencies have moderate variance," i.e. they have similar magnitude. This property is needed in order to allow the relative strength of the frequency triples to be altered without requiring a modification that would be perceptually noticeable. Superficially, this scheme is similar to our own proposal, also drawing an analogy to spread spectrum communications. However, the structure of their watermark is different from ours, and the set of frequencies is not chosen based on any direct perceptual significance, or relative energy considerations. Further, because the variance between the eight frequency coefficients is small, one would expect that their technique may be sensitive to noise or distortions. This is supported by the experimental results that report that the "embedded labels are robust against JPEG compression for a quality factor as low as about 50%." By comparison, I demonstrate that our method performs well with compression quality factors as low as 5%. A proposal by Koch and Zhao [7] used not triples of frequencies but pairs of frequencies, and was again designed specifically for robustness to JPEG compression. Nevertheless, they state that "a lower quality factor will increase the likelihood that the changes necessary to superimpose the embedded code on the signal will be noticeably visible." In a second method, designed for black and white images, no frequency transform is employed. Instead, the selected blocks are modified so that the relative frequency of white and black pixels encodes the final value. Both watermarking procedures are particularly vulnerable to multiple document attacks. To protect against this, Zhao and Koch propose a *distributed* 8X8 block created by randomly sampling 64 pixels from the image.

However, the resulting DCT has no relationship to that of the true image and consequently may be likely to cause noticeable artifacts in the image and be sensitive to noise.

In addition to direct work on watermarking images, there are several works of interest in related areas [12]. How should a watermark be structured to maximize its robustness? Cox et al. [1] suggest that an image watermark should be restricted to the "perceptually significant" (e.g., large-amplitude) spectral components. Large-amplitude components offer better masking potential and cannot be removed without also degrading the image.

On the other hand, Piva et al. [10] suggest placing the watermark in the middle frequencies. Hsu and Wu explain that, with regard to imperceptibility, the human visual system is less sensitive to high spatial frequencies, but with regard to robustness, processing like compression only preserves low-spatial frequencies. As a compromise, the watermark should lie in the middle frequencies.

Uncertainty about the proper structure of a watermark remains. Part of the difficulty in answering the question is that robustness is easy to postulate but hard to measure. Currently, it is still difficult to quantify the detectability of an attacked watermark and the quality of the attacked data. They are based on selecting a distortion measure, performing a battery of attacks on different watermarks, and measuring quantities such as the probability of error after each attack. They propose a methodology for evaluating robustness experimentally, but they are specific to a given watermarking method and the set of attacks. Moreover, they lack a strong theoretical foundation and development.

Examples of attacks already included compression, linear filtering, geometric transformations, and D/A–A/D conversion. Some extensive lists appear in [1] and [2], but it is impossible to name all of the potential attacks.

2.5: DIGITAL WATERMARKING SYSTEM OVERVIEW

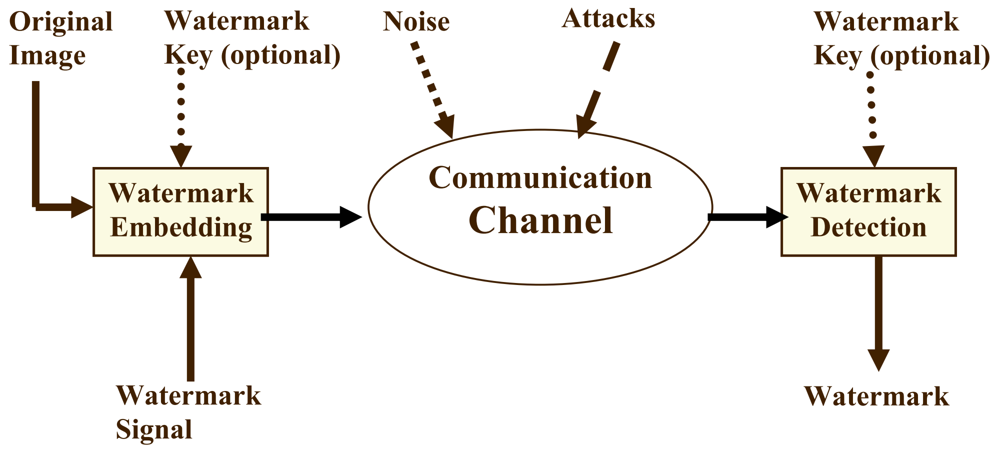


Figure 1: A Common Digital Watermarking System

The digital watermarking system essentially consists of a watermark embedder and a watermark detector (see Figure1). The watermark embedder inserts a watermark onto the cover signal (original image) and the watermark detector detects the presence of watermark signal. Note that an entity called watermark key is used during the process of embedding and detecting watermarks. The watermark key has a one-to-one correspondence with watermark signal (i.e., a unique watermark key exists for every watermark signal). The watermark key is private and known to only authorized parties and it ensures that only authorized parties can detect the watermark. Further, note that the communication channel can be noisy and hostile (i.e., prone to security attacks) and hence the digital watermarking techniques should be resilient to both noise and security attacks.

2.6: STRUCTURE OF A WATERMARKING SYSTEM

Every watermarking system consists at least of two different parts: watermark embedding unit and watermark detection and extraction unit. Figure 2 shows an example of embedding unit for still images. The unmarked image is passed through a perceptual analysis block that determines how much a certain pixel can be altered so that the resulting watermarked image is indistinguishable from the original. This takes into account the human eye sensitivity to changes in flat areas and its relatively high tolerance to small changes in edges. After this so-called *perceptual-mask* has been computed, the information to be hidden is shaped by this mask and spread all over the original image. This spreading technique is similar to the interleaving used in other applications involving coding, such as compact disc storage, to prevent damage of the information caused by scratches or dust. In our case, the main reason for this spreading is to ensure that the hidden information survives cropping of the image. Moreover, the way this spreading is performed depends on the secret key, so it is difficult to recover the hidden information if one is not in possession of this key. Additional key-dependent uncertainty can be introduced in pixel amplitudes (recall that the perceptual mask imposes only an upper limit). Finally, watermark is added to the original image.

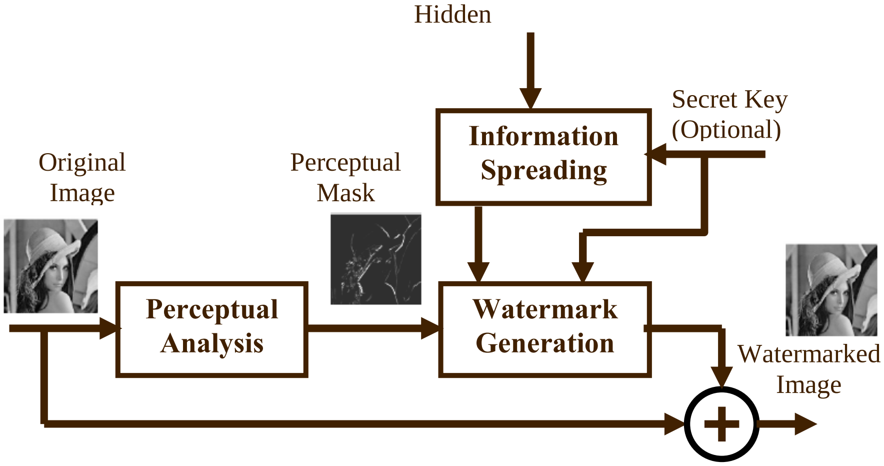


Figure 2: Watermark insertion unit

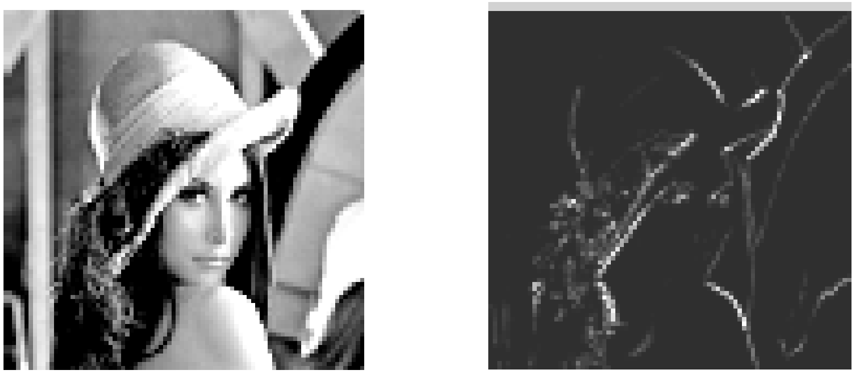


Figure 3: Original ‘Lenna’ image and Perceptual Mask of the image

Figure 3 represents the perceptual mask that results after analyzing the image presented in Figure 3. Higher intensity (i.e., whiter) levels imply that higher perturbations can be made at those pixels without perceptible distortion. Thus, the higher capacity areas for hiding information correspond to edges. These masks are computed by using some known results on how the human eye works in the spatial domain. Different results are obtained when working on other domains, such as the DCT (Discrete Cosine Transform) or Wavelet transform. In fact, when working on the DCT coefficients domain one may take advantage of the relative independence between the maximum allowable perturbations at every coefficient. This is useful when dealing with the mask for watermarking purposes.

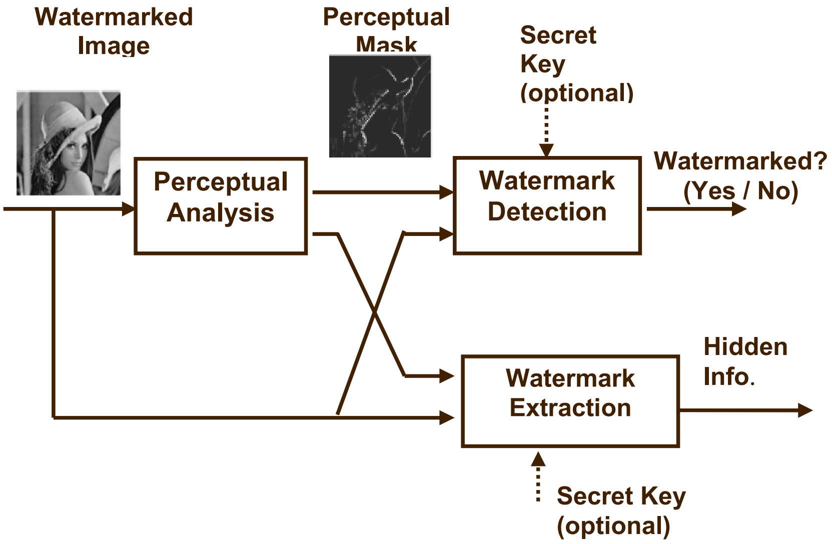


Figure 4: Watermark detection and extraction unit

Above, Figure 4 shows the typical configuration of a watermark detection and extraction unit. Watermark detection involves deciding whether a certain image has been watermarked with a given key. Note then that a watermark detector produces a binary output. Important considerations here are the probability of correct detection PD (i.e., the probability of correctly deciding that a watermark is present) and the probability of false alarm PF (i.e., the probability of incorrectly deciding that an image has been watermarked with a certain key). These two measures allow us to compare different watermarking schemes: One method will be superior if achieves a higher PD for a fixed PF. Note also that for a watermarking algorithm to be useful it must work with extremely low probabilities of false alarm.

Watermark detection is usually done by correlating the watermarked image with a locally generated version of the watermark at the receiver side. This correlation yields a high value when the watermark has been obtained with the proper key. It is possible to improve the performance of the detector by eliminating original image-induced noise with signal processing. It is worthy of remark that some authors, like Cox I.J. in [1], propose using the original image in the detection process.

Once the presence of the watermark has been correctly detected, it is possible to extract the hidden information. The procedure is also generally done by means of a cross-correlation but in this case, an independent decision has to be taken for every information bit with a sign slicer. In fact, I.J. Cox et al. [1] have also shown that this correlation structure has not been well-founded and significant improvements are achievable when image statistics are available. For instance, the widely-used DCT coefficients used in the JPEG and MPEG-2 standards are well approximated by *generalized Gaussian probability density functions* that yield a considerably different extraction scheme. Obviously, when extracting the information the most adequate parameter for comparison purposes is the *probability of bit error* Pb, identical to that used in digital communications. This is not surprising because watermarking creates a hidden (also called *steganographic*) channel on which information is conveyed.

Chapter 3:
SPREAD-SPECTRUM WATERMARKING ALGORITHM (COX'S METHOD)

There are two parts to building a strong watermark: the *watermark structure* and the *embedding strategy*. In order for a watermark to be robust and secure, these two components must be designed correctly. I.J. Cox et al. [1] have provide two key insights that make our watermark both robust and secure: They argue that the watermark be placed explicitly in the perceptually most significant components of the data, and that the watermark be composed of random numbers drawn from a Gaussian ***N***(0,1) distribution.

The stipulation that the watermark be placed in the perceptually significant components means that an attacker must target the fundamental structural components of the data, thereby heightening the chances of fidelity degradation. While this strategy may seem counterintuitive from the point of view of steganography (how can these components hide any signal?), I.J. Cox et al. discovered that the significant components have a *perceptual capacity* that allows watermark insertion without perceptual degradation. Further, most processing techniques applied to media data tend to leave the perceptually significant components intact. While one may choose from a variety of such components, in this procedure we focus on the perceptually significant *spectral* components of the data. This simultaneously yields high perceptual capacity and achieves a uniform spread of watermark energy in the pixel domain.

3.1: STRUCTURE OF THE WATERMARK

Cox et al. now give a high-level overview of their basic watermarking scheme; many variations are possible. In its most basic implementation, a watermark consists of a sequence of real numbers $X = x_1, \cdots, x_n$. In practice, we create a watermark where each value x_i is chosen independently according to ***N(0, 1)***; (where ***N(***μ***,*** σ^2***)*** denotes a normal distribution with mean μ and variance σ^2). We assume that numbers are represented by a reasonable but finite precision and ignore these insignificant round-off errors. This procedure exploits the fact that each component of the watermark is chosen from a normal distribution. Alternative distributions are possible, including choosing x_i uniformly from {1, -1}, {0, 1} or [0, 1]. However, as we discuss in IV-D, using such distributions leaves one particularly vulnerable to attacks using multiple watermarked documents.

3.2: WATERMARK EMBEDDING STRATEGY

The principle underlying watermark structuring strategy is that the mark be constructed from independent, identically distributed (i.i.d.) samples drawn from a Gaussian distribution. Once the significant components are located, Gaussian noise is injected therein. The choice of this distribution gives resilient performance against collusion attacks. The Gaussian watermark also gives Cox's [1] scheme strong performance in the face of quantization, and may be structured to provide low false positive and false negative detection.

Also, note that the techniques presented herein do not provide proof of content ownership on their own. We focus on the algorithms that insert messages into content in an extremely secure and robust fashion. Nothing prevents someone from inserting another message and claiming ownership. However, it is possible to couple these methods with strong authentication and other cryptographic techniques in order to provide complete, secure and robust owner identification and authentication.

A watermark should be embedded in the cover data's perceptually significant frequency components. Of course, the major problem then becomes how to imperceptibly insert a watermark into perceptually significant components of the frequency spectrum. I.J. Cox [1] presented a watermarking algorithm that is based on ideas from spread spectrum communications and relies on the use of the original image to extract the watermark.

Ultimately, no watermarking system can be made perfect. For example, a watermark placed in a textual image may be eliminated by using optical character recognition technology. However, for common signal and geometric distortions, the experimental results obtained suggest that our system satisfies most of the properties discussed in the introduction, and displays strong immunity to a variety of attacks in a collusion resistant manner.

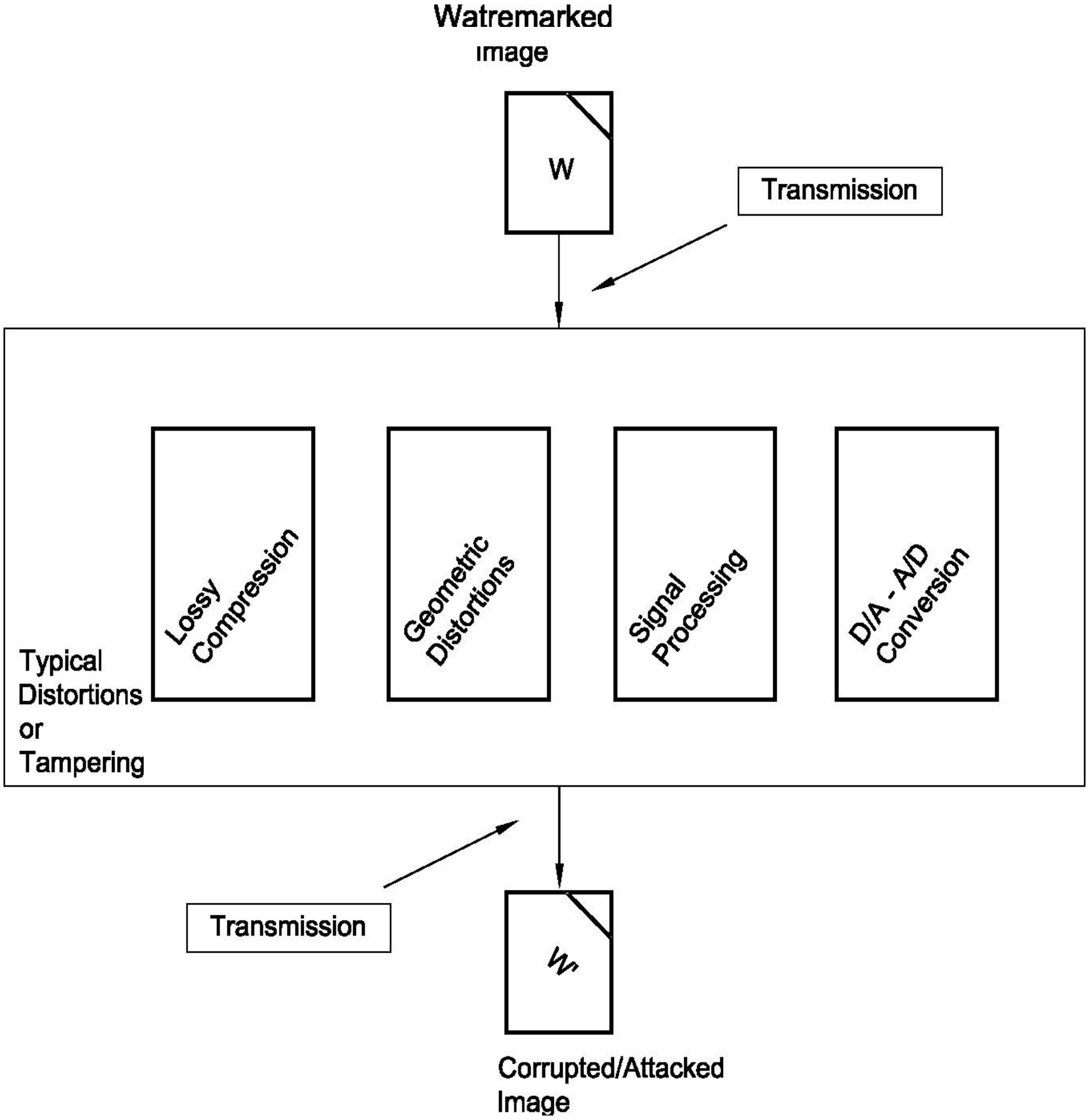


Figure 5: Common processing operations that a digital image could undergo.

3.3: WATERMARKING IN FREQUENCY DOMAIN

In order to understand the advantages of a frequency-based method, it is instructive to examine the processing stages that an image may undergo in the process of copying, and to study the effect that these stages could have on the data, as illustrated in Fig. 5. In the figure, "transmission" refers to the application of any source or channel code, and/or standard encryption technique to the data. While most of these steps are information lossless, many compression schemes (like JPEG) are lossy, and can potentially degrade the data’s quality, through irretrievable loss of information. In general, a watermarking scheme should be resilient to the distortions introduced by such algorithms.

Lossy compression is an operation that usually eliminates perceptually non-salient components of an image or sound. Most processing of this sort takes place in the frequency domain. In fact, data loss usually occurs among the high-frequency components.

After receipt, an image may endure many common transformations that are broadly categorized as geometric distortions or signal distortions. Geometric distortions are specific to images and video, and include such operations as rotation, translation, scaling and cropping. However, an affine scaling (shrinking) of the image leads to a loss of data in the high-frequency spectral regions of the image. Attacks like cropping, or the cutting out and removal of portions of an image, leads to irretrievable loss of image data; which may seriously degrade any spatially based watermark. However, a frequency-based scheme [1] spreads the watermark over the whole spatial extent of the image, and is therefore less likely to be affected by cropping.

Common signal distortions include digital-to-analog and analog-to-digital conversion, re-sampling, re-quantization, including dithering and recompression, and common signal enhancements to image contrast and/or color, and audio frequency equalization. Many of these distortions are nonlinear, and it is difficult to analyze their effect in either a spatial- or frequency-based method. However, the fact that the original image is known allows many signal transformations to be undone, at least approximately. For example, histogram equalization [3], a common nonlinear contrast enhancement method, may be removed substantially.

The watermark must not only be resistant to the inadvertent application of the aforementioned distortions. It must also be immune to intentional manipulation by malicious parties.

3.4: SPREAD SPECTRUM CODING OF WATERMARK

The above discussion illustrates that the watermark should not be placed in perceptually insignificant regions of the image (or its spectrum), since many common signal and geometric processes affect these components. For example, a watermark placed in the high-frequency spectrum of an image can be easily eliminated with little degradation to the image by any process that directly or indirectly performs low-pass filtering. The problem then becomes how to insert a watermark into the most perceptually significant regions of the spectrum in a fidelity preserving fashion. Clearly, any spectral coefficient may be altered, provided such modification is small. However, very small changes are very susceptible to noise.

To solve this problem, the frequency domain of the image at hand is viewed as a "*communication channel*" and correspondingly, the watermark is viewed as a "*signal that is transmitted through it"*. Attacks and unintentional signal distortions are thus treated as "*noise"* that the immersed signal must be immune to. While we use this methodology to hide watermarks in data, the same rationale can be applied to sending any type of message through media data.

Cox et al. originally conceived this approach by analogy to spread spectrum communications [8]. In spread spectrum communications, one transmits a narrowband signal over a much larger bandwidth such that the signal energy present in any single frequency is undetectable. Similarly, the watermark is spread over very many frequency bins so that the energy in any one bin is very small and certainly undetectable. Nevertheless, because the watermark verification process knows the location and content of the watermark, it is possible to concentrate these many weak signals into a single output with high signal-to-noise ratio (**SNR**). However, to destroy such a watermark would require noise of high amplitude to be added to *all* frequency bins.

Spreading the watermark throughout the spectrum of an image ensures a large measure of security against unintentional or intentional attack: First, the location of the watermark is not obvious. Furthermore, frequency regions should be selected in a fashion that ensures severe degradation of the original data following any attack on the watermark.

A watermark that is well placed in the frequency domain of an image will be practically impossible to see. This will always be the case if the "*energy"* in the watermark is sufficiently small in any single frequency coefficient. Moreover, it is possible to increase the energy present in particular frequencies by exploiting knowledge of masking phenomena in the human auditory and visual systems. Perceptual masking refers to any situation where information in certain regions of an image or a sound is occluded by perceptually more prominent information in another part of the scene. In digital waveform coding, this frequency domain (and, in some cases, time/pixel domain) masking is exploited extensively to achieve low bit rate encoding of data. It is known that both the auditory and visual systems attach more resolution to the high-energy, low-frequency, spectral regions of an auditory or visual scene. Further, spectrum analysis of images and sounds reveals that most of the information in such data is located in the low-frequency regions.

In principle, any frequency domain transform can be used. However, in the experiments we use a Fourier domain method based on the DCT [3]. Each coefficient in the frequency domain has a *perceptual capacity*, that is, a quantity of additional information can be added without any (or with minimal) impact to the perceptual fidelity of the data. To determine the perceptual capacity of each frequency, one can use models for the appropriate perceptual system or simple experimentation. In practice, in order to place a length ***n*** watermark into an *NxN* image, we computed the *NxN* DCT of the image and placed the watermark into the ***n*** highest magnitude coefficients of the transform matrix, excluding the DC component. For most images, these coefficients will be the ones corresponding to the low frequencies.

In the next section, is provided a high level discussion of the watermarking procedure, describing the insertion, detection and extraction of the watermark.

3.5: SPREAD SPECTRUM WATERMARKING PROCEDURE

3.5.1: Inserting the Watermark

Fig.6 illustrates the general procedure for frequency domain watermarking. Upon applying a frequency transformation to the data, a *perceptual mask* is computed that highlights perceptually significant regions in the spectrum that can support the watermark without affecting perceptual fidelity. The watermark signal is then inserted into these regions in a manner described in Section 3.4. The precise magnitude of each modification is only known to the owner.

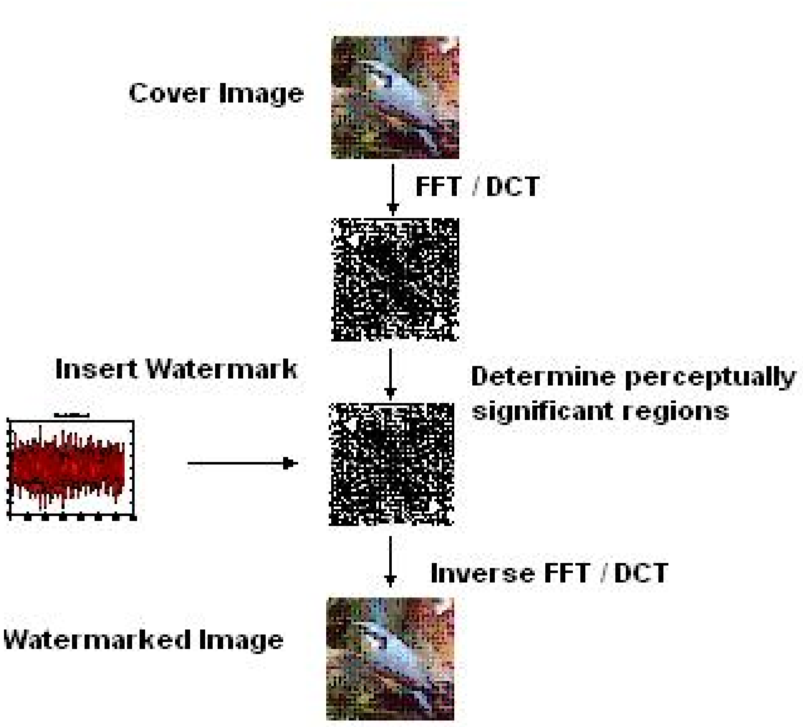


Figure 6: Stages of watermark insertion process.

By contrast, an attacker may only have knowledge of the possible range of modification. To be confident of eliminating a watermark, an attacker must assume that each modification was at the limit of this range, despite the fact that few such modifications are typically this large. As a result, an attack creates visible (or audible) defects in the data. Similarly, unintentional signal distortions due to compression or image manipulation, must leave the perceptually significant spectral components intact, otherwise the resulting image will be severely degraded. This is why the watermark is robust.

When we insert X into V to obtain V' we specify a scaling parameter α, which determines the extent to which X alters V. Three natural formulae for computing are

$$v'_i = v_i + \alpha x_i \quad (3.1)$$

$$v'_i = v_i(1 + \alpha x_i) \quad (3.2)$$

$$v'_i = v_i(e^{\alpha x_i}). \quad (3.3)$$

Equation (3.1) is always invertible, and (3.2) and (3.3) are invertible if $v_i \neq 0$, which holds in all of our experiments. Given V^*, we can therefore compute the inverse function to derive X^* from V^* and V. Equation (3.1) may not be appropriate when the values vary widely. If $v_i = 10^6$, then adding 100 may be insufficient for establishing a mark, but if $v_i = 10$ adding 100 will distort this value unacceptably. Insertion based on (3.2) or (3.2) are more robust against such differences in scale. We note that (3.2) and (3.2) give similar results when αx_i is small. Also, when v_i is positive, then (3.2) is equivalent to $\lg(v'_i) = \lg(v_i) + \alpha x_i$, and may be viewed as an application of (3.1) to the case where the logarithms of the original values are used.

3.5.2: Determining Scaling Parameter (α)

A single scaling parameter α may not be applicable for perturbing all of the values v_i, since different spectral components may exhibit more or less tolerance to modification. More generally one can have multiple scaling parameters $\alpha_1, \cdots, \alpha_n$ and use update rules such as $v'_i = v_i(1 + \alpha_i x_i)$. We can view α_i as a relative measure of how much one must alter v_i to alter the perceptual quality of the document. A large α_i means that one can perceptually “get-away” with altering v_i, by a large factor without degrading the document.

There remains the problem of selecting the multiple scaling values. In some cases, the choice of α_i may be based on some general assumption. E.g. equation (3.2) is a special case of the generalized equation (3.1) $(v'_i = v_i + \alpha_i x_i)$, for $\alpha_i = \alpha v_i$. Essentially, equation (3.2) makes the reasonable assumption that a large value is less sensitive to additive alterations than a small value. In all our experiments we simply use equation (3.2) with a scaling parameter ***α* = 0.1**.

3.5.3: Choosing the Length (n), of the Watermark

The choice of dictates the degree to which the watermark is spread out among the relevant components of the image. In general, as the number of altered components are increased the extent to which they must be altered decreases. For a more quantitative assessment of this tradeoff, we consider watermarks of the form $v'_i = v_i + \alpha x_i$ and model a white noise attack by $v^*_i = v'_i + r_i$ where r_i are chosen according to independent normal distributions with standard deviation σ. For the watermarking procedure described below, one can recover the watermark when α is proportional to $\sigma/\sqrt{n}$.

Note that the number of bits of information associated with the watermark can be arbitrary—the watermark is simply used as an index to a database entry associated with the watermark.

3.5.4: Extracting the Watermark

The procedure for extraction and decoding of the watermark is shown in figure 11. We extract from each image or document $\boldsymbol{D}$ a sequence of values $\boldsymbol{V}$ = $v_1, \cdots, v_n$, into which we insert a watermark $\boldsymbol{X}$ = $x_1, \cdots, x_n$, to obtain an adjusted sequence of values $V' = v'_1, \cdots, v'_n$. V' is then inserted back into the document in place of $\boldsymbol{V}$ to obtain a watermarked document D'. One or more attackers may then alter D', producing a new document $\boldsymbol{D^*}$. Given $\boldsymbol{D}$ and $\boldsymbol{D^*}$, a possibly corrupted watermark $\boldsymbol{X^*}$ is extracted and is compared to X for statistical significance. We extract X^* by first extracting a set of values $V^* = v_1^*, \ldots, v_n^*$ from D^* (using information about D) and then generating X^* from V^* and V. Frequency-domain based methods for extracting V^* and V and inserting V' are given in Section 3.4.

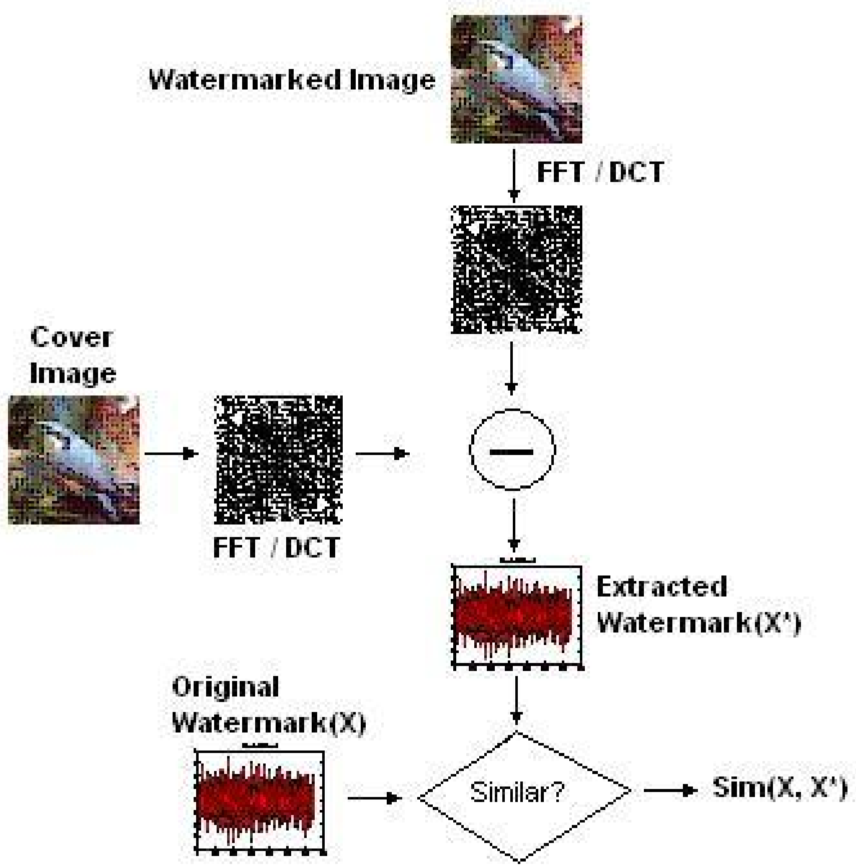


Figure 11: Extraction and decoding of the Watermark.

3.5.5: Evaluating the Similarity of Watermarks

It is highly unlikely that the extracted mark X^* will be identical to the original watermark X . Even the act of re-quantizing the watermarked document for delivery will cause X^* to deviate from X. We measure the similarity of X^* and X by

$$\mathrm{sim}(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X^*}} \tag{3.4}$$

Many other measures are possible, including the standard correlation coefficient. To decide whether ***X*** and ***X**** match, one determines whether **sim*(X*,X)*** > ***T***, where ***T*** is some threshold. Setting the detection threshold is a classical decision estimation problem in which we wish to minimize both the rate of false negatives (missed detections) and false positives (false alarms). I.J. Cox et al. [1] have chosen this measure so that it is particularly easy to determine the probability of false positives.

3.5.5(a): Computing the Probability of False Positives:

There is always the possibility that ***X*** and ***X**** will be very similar purely by random chance; hence, any similarity metric will give “significant” values that are spurious. We analyze the probability of such false positives as follows. Suppose that the creators of document ***D**** had no access to ***X*** (either through the seller or through a watermarked document). Then, even conditioned on any fixed value for ***X****, each x_i will be independently distributed according to ***N***(0, 1). That is, ***X*** is independent of ***X****.

The distribution on ***X*. X*** may be computed by first writing it as $\sum_{i=1}^{n} x_i^* x_i$, where x_i^* is a constant. Using the well-known formula for the distribution of a linear combination of variables that are independent and normally distributed, ***X*. X*** will be distributed according to

$$N\left(0, \sum_{i=1}^{n} x_i^{*2}\right) = N(0, X^* \cdot X^*),$$

Thus, **sim*(X*,X)*** is distributed according to ***N***(0,1). We can then apply the standard significance tests for the normal distribution. E.g., if ***X**** is created independently from ***X*** then the probability that **sim*(X*,X)* > 6,** [1]; is the probability of a normally distributed random variable exceeding its mean by more than six standard deviations.

Hence, for a small number of documents, setting the threshold $\boldsymbol{T}$ at equal to six will cause spurious matching to be extremely rare. Of course, the number of tests to be performed must be considered in determining what false positive probability is acceptable. For example, if one tests an extracted watermark $\boldsymbol{X^*}$ against 10^6 watermarks, then the probability of a false positive is increased by a multiplicative factor of 10^6 as well. I note that our similarity measure and the false-positive probability analysis does not depend on ***n***, the size of the watermark. However, ***n*** implicitly appears, since for example, **sim**$(\boldsymbol{X^*},\boldsymbol{X})$ is likely to be around $\sqrt{\boldsymbol{n}}$ when $\boldsymbol{X}$ is generated in the prescribed manner. As a rule of thumb, larger values of ***n*** tend to cause larger similarity values when $\boldsymbol{X^*}$ and $\boldsymbol{X}$ are genuinely related (e.g., $\boldsymbol{X^*}$ is a distorted version of $\boldsymbol{X}$), without causing larger similarity values when $\boldsymbol{X}$ and $\boldsymbol{X^*}$ are independent. This benefit must be balanced against the tendency for the document to be more distorted when ***n*** is larger.

3.5.5(b): A Remark on Watermark Quantization

In the above analysis, I treated all of the vectors as consisting of ideal real numbers. In practice, the actual values inserted will be quantized to some extent. Our analysis of false positives does not depend on the distribution or even the domain of possible $\boldsymbol{X^*}$, and hence holds regardless of quantization effects.

There is an additional, extremely low-order quantization effect that occurs because $\boldsymbol{X}$ is generated with only finite precisions. However, this effect is caused only by the arithmetic precision, and not on the constraints imposed by the document. If each $x_i \in \boldsymbol{X}$ is stored as a double-precision real number, the difference between the calculated value of **sim**$(\boldsymbol{X^*},\boldsymbol{X})$ and its "ideal" value will be quite small for any reasonable ***n*** and any reasonable bound on the dynamic range of $\boldsymbol{X^*}$.

3.5.6: Robust Statistics

The above analysis required only the independence of X from X^* and did not rely on any specific properties of X^* itself. This fact gives us further flexibility when it comes to preprocessing X^*. We can process X^* in a number of ways to potentially enhance our ability to extract a watermark. For example, in some experiments on images we encountered instances where the average value of x_i^*, denoted $E_i(X^*)$, differed substantially from zero, due to the effects of a dithering procedure. While this artifact could be easily eliminated as part of the extraction process, it provides a motivation for post-processing extracted watermarks. I.J. Cox et al. have found that the simple transformation $x_i^* \leftarrow x_i^* - E_i(X^*)$ yielded superior values of **sim**(X^*,X). The improved performance resulted from the decreased value of X^*. X^*; the value of was only slightly affected.

In my experiments, it is frequently observed that x_i^* could be greatly distorted for some values of i. One post-processing option is to simply ignore such values, setting them to zero.
That is

$$x_i^* \leftarrow \begin{cases} x_i^*, & \text{if } |x_i^*| \leq \text{tolerance} \\ 0, & \text{otherwise.} \end{cases}$$

Again, the goal of such a transformation is to lower X^*. X^*. A less abrupt version of this approach is to normalize the X^* values to be either -1, 0 or 1, by

$$x_i^* \leftarrow \mathsf{sign}(x_i^* - E_i(X^*)).$$

This transformation can have a dramatic effect on the statistical significance of the result.

A natural question is whether such post-processing steps run the risk of generating false positives. Indeed, the same potential risk occurs whenever there is any latitude in the procedure for extracting X^* from D^*. However, as long as the method for generating a set of values for X^* depends solely on D

and $\boldsymbol{D*}$, our statistical significance calculation is unaffected. The only caveat to be considered is that the bound on the probability that one of $X_1^*, \cdots X_k^*$ generates a false positive is the sum of the individual bounds. Hence, to convince someone that a watermark is valid, it is necessary to have a published and rigid extraction and processing policy that is guaranteed to only generate a small number of candidate $\boldsymbol{X*}$.

3.6: Watermarking the Gray-scale Images

Figures 7 is a 256x256 gray-scale image of a horse and figure 8 shown above has a random watermark embedded into the original image. Practically; for any human being it is nearly impossible to detect the presence of a watermark embedded into the original cover image.

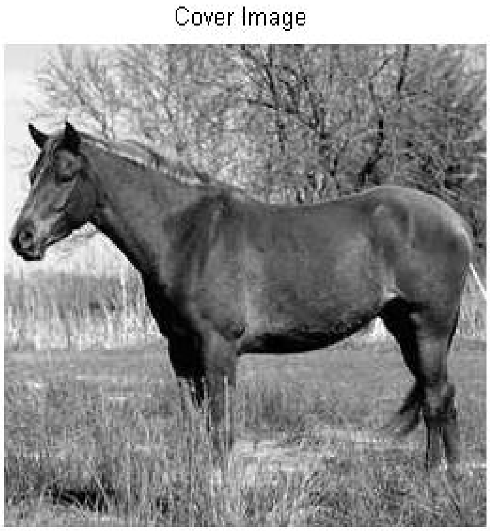

Cover Image

Figure 7: Gray-scale natural original image (256x256 size)

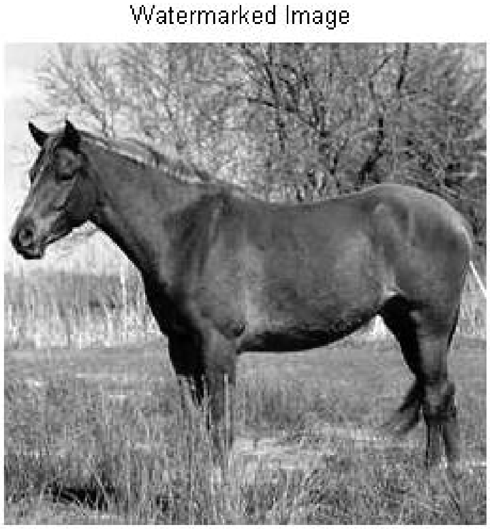

Watermarked Image

Figure 8: Watermarked gray-scale image (256x256 size) with random watermark and α = 0.1.

3.7: Watermarking the Color Images

Figures 9 is a 256x256 color image of a bird and figure 10 shown on next page has a random watermark embedded into the original image. Here also, practically; for any human being it is nearly impossible to detect the presence of a watermark embedded into the original cover image.

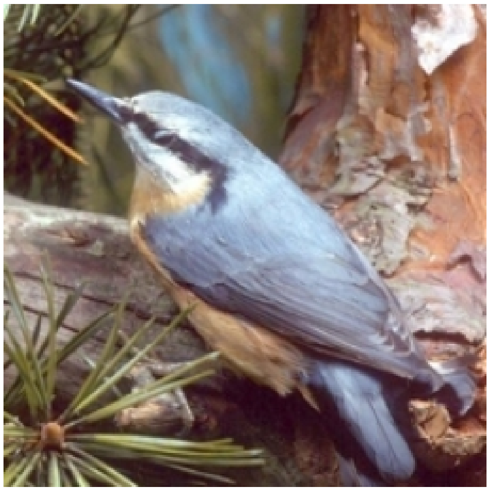


Figure 9: Original color image “Bird” (size 256x256).

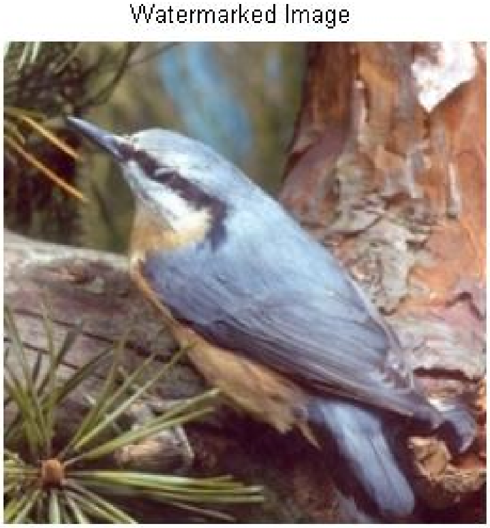


Figure 10: Watermarked version of image in figure 10 above; embedded with random watermark and α = 0.1.

The gray-scale or intensity images are directly processed for the watermark embedding and extracting algorithms. However, for color images a little modification in the procedure is required, though the primary algorithms of watermark embedding and extracting remain unaltered for both types of images.

Cox's method for watermarking the color images is also practically implemented. The most common transformation of a color image is to convert it to black and white (i.e. gray-scale). Color images are therefore converted into a **YIQ** representation and the intensity component **Y** is then watermarked. The color image can then be converted to other formats, but must be converted back to YIQ prior to extraction of the watermark.

Chapter 4:
EXPERIMENTS WITH SPREAD-SPECTRUM WATERMARKING ALGORITHM

In order to evaluate the proposed watermarking scheme, I have conducted experiments on “Horse” (256x256; gray-scale image) and “Bird” (256x256, natural color image) of figure 8 and figure 10 separately; and obtained the watermarked version of figure 9 and figure 11 accordingly. I then subjected the watermarked image to a series of image processing attacks on both of the images. These experiments showed resilience to many types of common image processing algorithms. Of note is this method’s resistance to JPEG conversion. The watermark detector’s response was well above the threshold, even with JPEG compressed image with 5% quality (i.e. 95% crucial data missing). Note that in the case of affine transforms, registration to the original image is crucial to successful extraction.

In experiments with 256x256 gray scale and color images a **random watermark** of length = 1000 was embedded; by modifying the same (1000) number of the most perceptually significant components of the images’ spectrum, using equation (3.2). A fixed scale factor of **α = 0.1** was used throughout.

Note: All the following experiments were performed using MATLAB (Release 14), on P-4 (2.4GHz) system with 256MB RAM, and WINOWS-XP environment. All attacks were conducted on both gray-scale and color images of size 256x256. However, images of different sizes can also be used, without any remarkable modification in the MATLAB program, which is included in the Appendix -A.

4.1: Experiment A: Uniqueness of Watermark

Figure 12 and fig.13 show the response of the watermark detector to 1000 randomly generated watermarks of which only one matches the watermark present in fig. 9 and fig.11 respectively. The positive response due to the correct watermark is very much stronger than the response to incorrect watermarks, suggesting that the algorithm has very low false positive response rates.

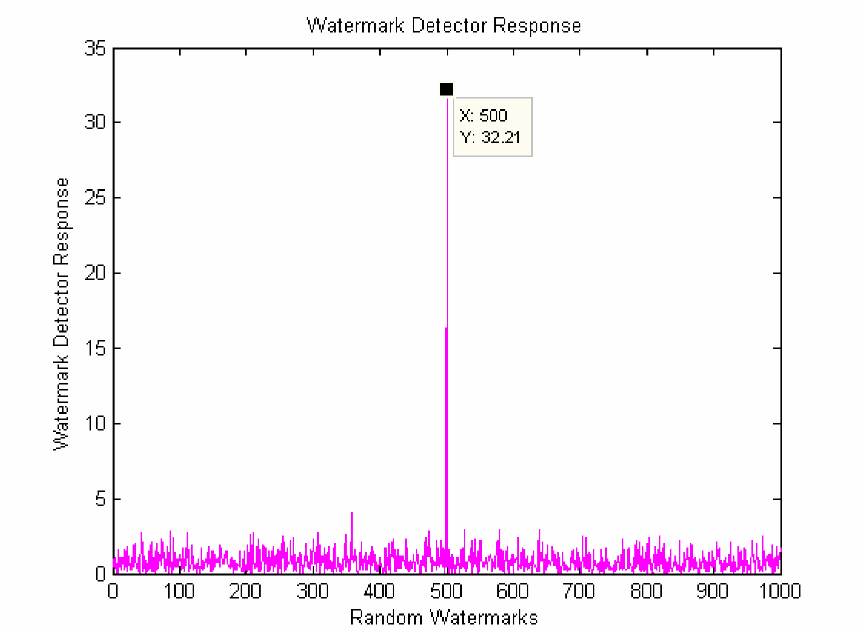


Figure12: Watermark detector response to 1000 randomly generated watermarks. Only one watermark (to which the detector was set to respond) matches that is present in watermarked gray-scale image of "Horse" in Figure 9.

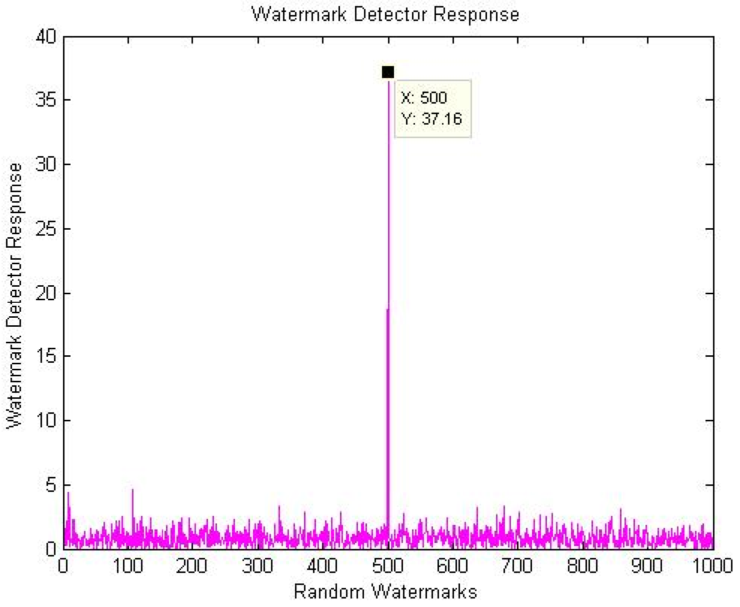


Figure 13: Watermark detector response to 1000 randomly generated watermarks. Only one watermark (to which the detector was set to respond) matches that is present in watermarked color image of "Bird" in Fig. 11.

4.2: Experiment B: Image Scaling and Re-scaling

In this experiment, I scaled the watermarked “Horse” and “Bird” both images to half of its original size, along both X and Y directions, as shown in fig.14(a) and 15(a) respectively. In order to recover the watermark, the quarter-sized image was rescaled to its original dimensions, as shown in fig.14(b) and 15(b) respectively, in both it is clear that considerable finer details have been lost in the scaling process. This is expected since sub-sampling of the image requires a low-pass spatial filtering operation.

The response of the watermark detector to the original gray-scale watermarked image of fig.9 was 32.2088. Which compare to a response of 13.266 as shown in fig. 14(c) for the rescaled version fig. 14(b). While the detector response is down by nearly 50%, the response is still well above random chance levels suggesting that the watermark is robust to geometric distortions. Moreover, it should be noted that 75% of the original data is missing from the scaled down image of fig. 14(a).

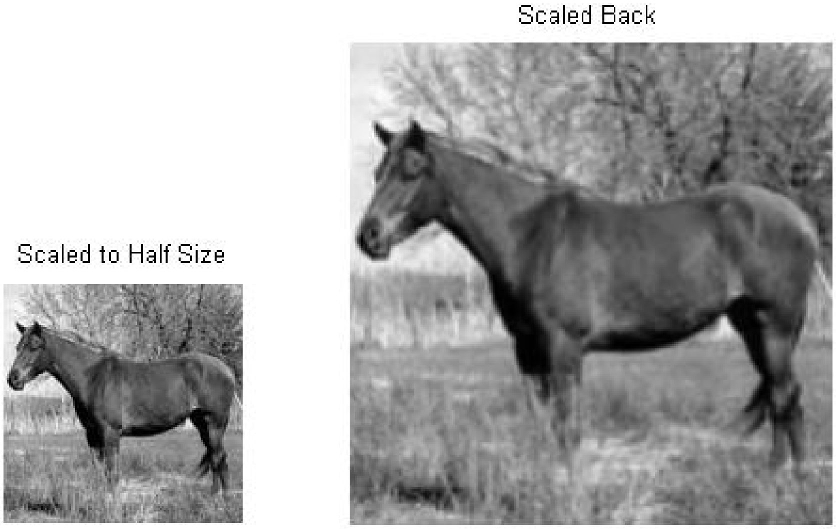


Figure 14(a): 0.5x Scaled image of “Horse” and 14(b): Scaled-back image to original size, showing noticeable loss of finer details.

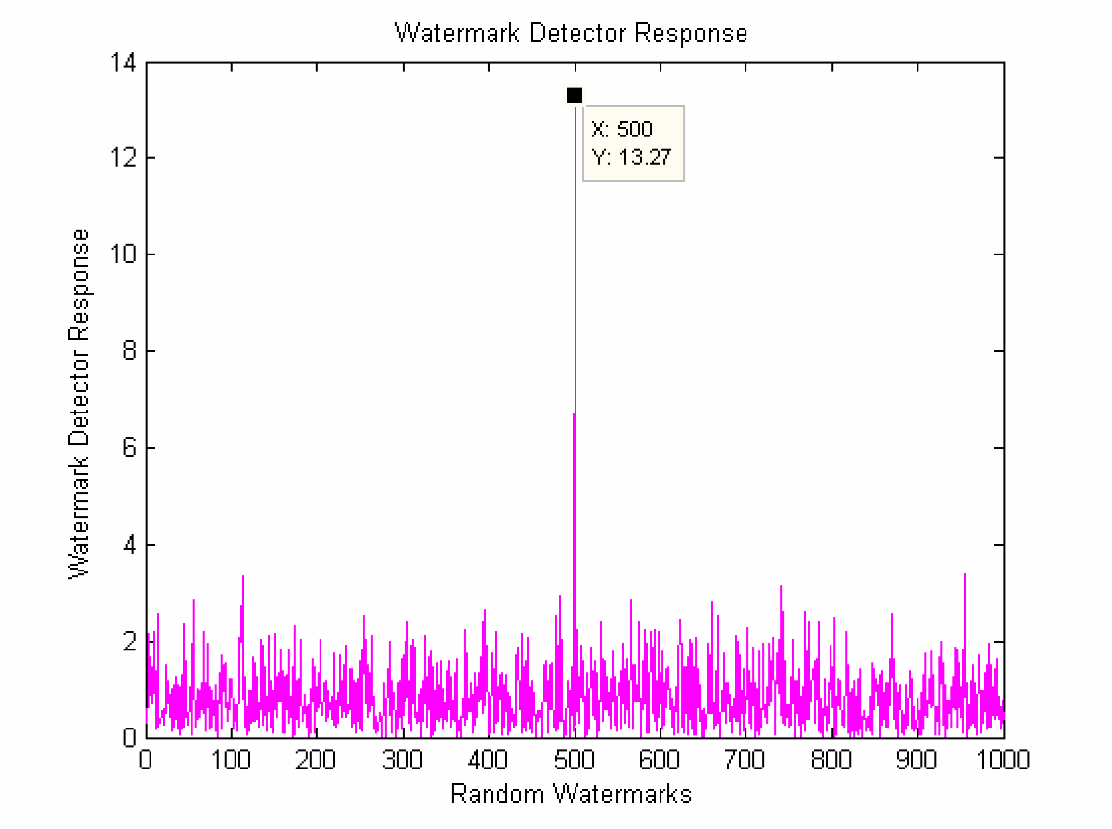


Figure 14(c): Watermark detector response to resized gray-scale image as shown in 14(b).

The color image was also subjected to similar attack and response of the watermark detector was observed.

I found that the response of the watermark detector to the original watermarked color image of fig.11 was 37.1567. Which compare to a response of 18.9143 as shown in fig. 15(c) for the rescaled version figure15(b). Again, the detector response is down by nearly 50%, and the response is still well above random chance levels suggesting that the watermark is robust to geometric distortions in color image too. Here also, it is to be noted that 75% of the original data is missing from the scaled down image of fig. 15(a). These images are shown on the next page.

.

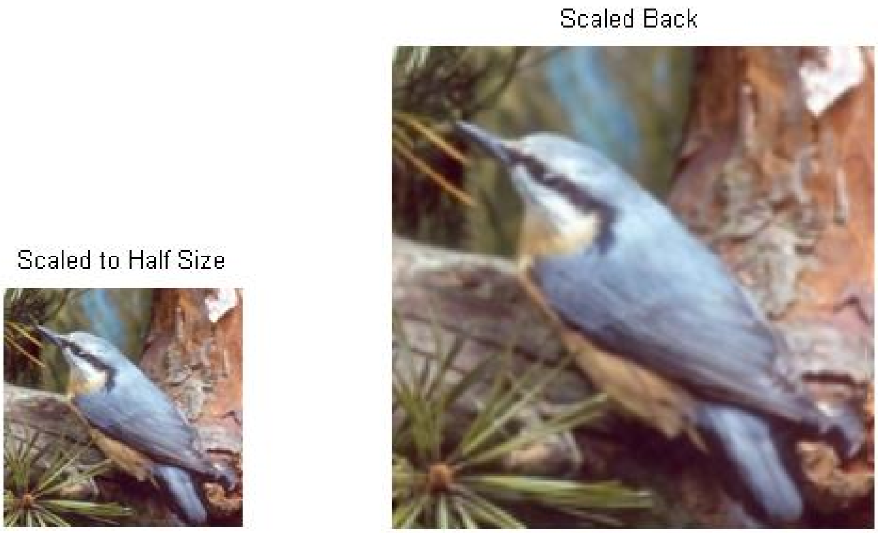


Figure 15(a): 0.5x Scaled image of "Bird" and 15(b): Scaled-back image to original size, showing noticeable loss of finer details.

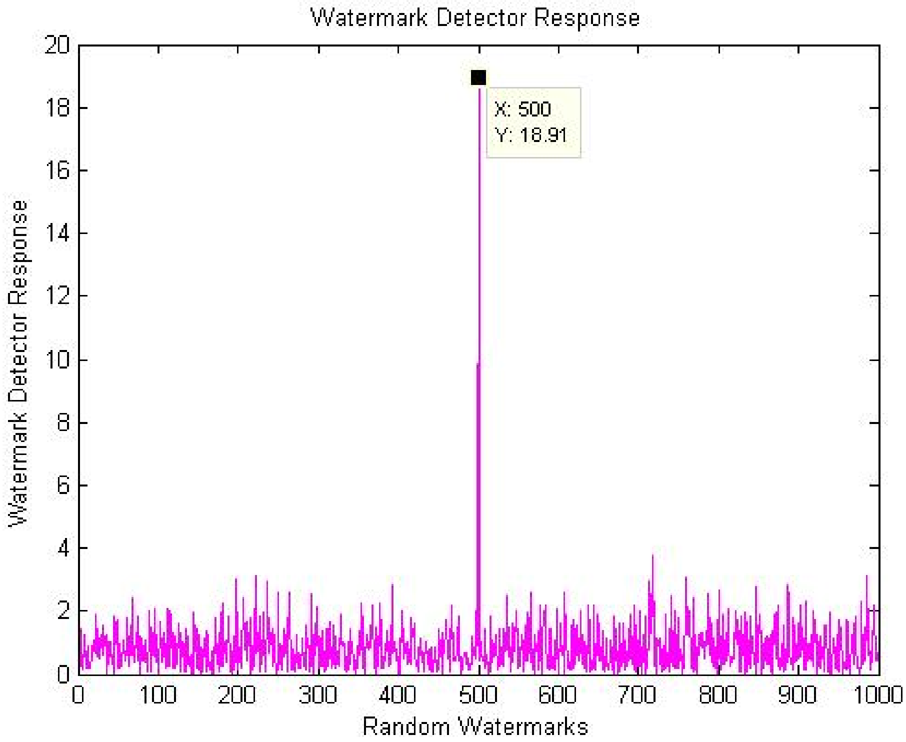


Figure 15(c): Watermark detector response to resized image as shown in 15(b).

4.3: Experiment C: JPEG Compression Distortion

The gray-scale images shown in the following figure 16(a), 16(b) and 16(c) are obtained after performing JPEG compression and these images are checked for the presence of the random watermark. The response of the detector is shown in figure 17(a), 17(b) and 17(c) respectively. These results are noted in table 1.

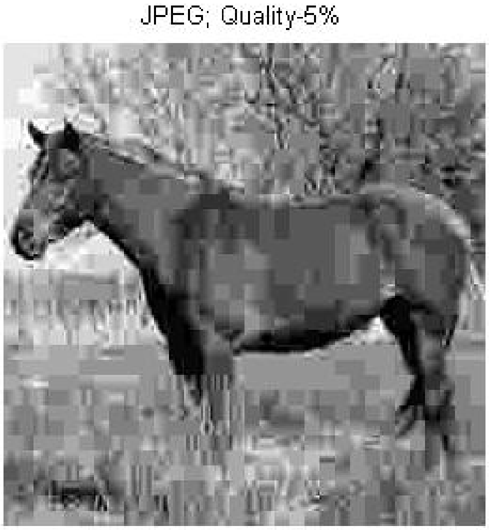


Figure 16(a): JPEG compressed version of "Horse" with 5% quality, 0% smoothing.

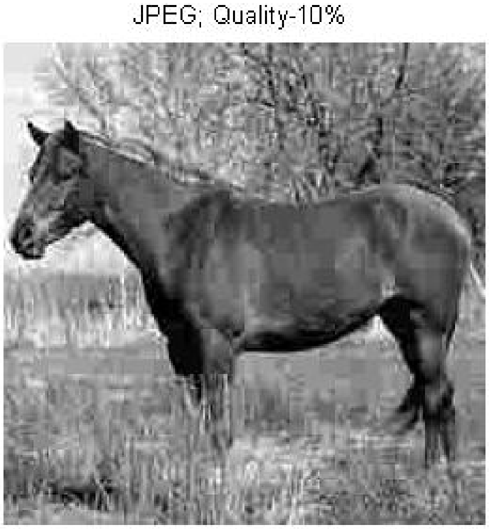


Figure 16(b): JPEG compressed version of "Horse" with 10% quality, 0% smoothing.

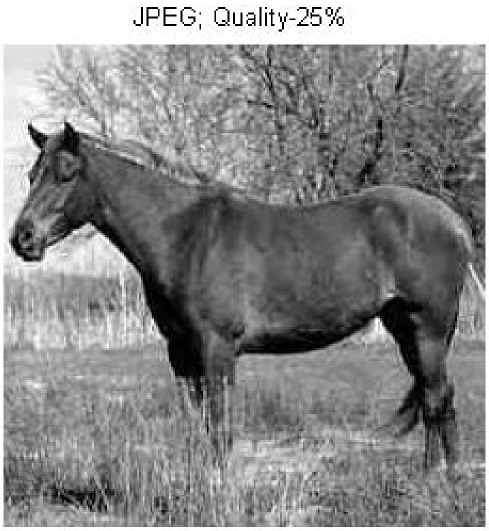


Figure 16(c): JPEG compressed version of “Horse” with 25% quality, 0% smoothing.

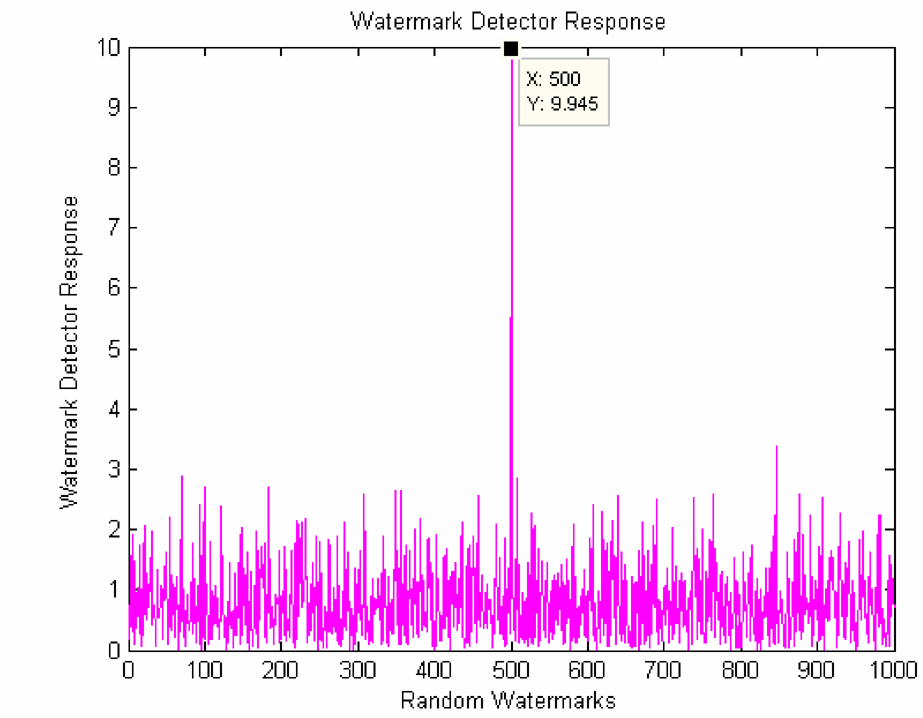


Figure 17(a): Watermark detector response to JPEG compressed version of “Horse” with 5% quality, 0%

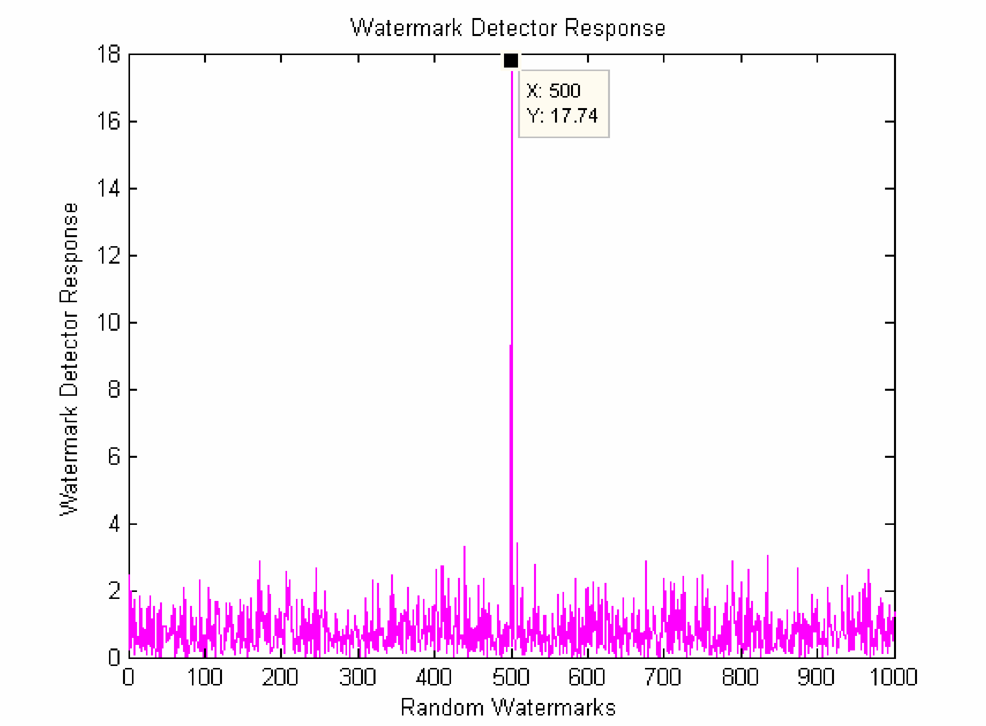


Figure 17(b): Watermark detector response to JPEG compressed version of "Horse" with 10% quality, 0%

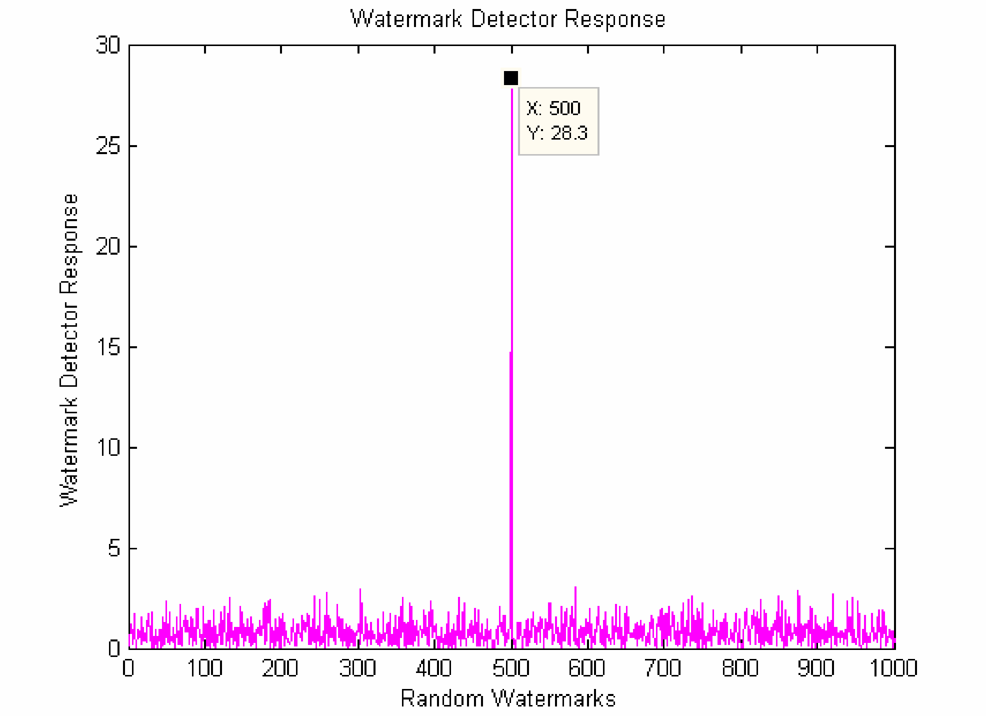


Figure 17(c): Watermark detector response to JPEG compressed version of "Horse" with 25% quality, 0%

Similarly, figures 18(a), 18(b) and 18(c) respectively show JPEG compressed versions of the “Bird” image with parameters of 5%, 10% and 25% quality, which results in clearly visible distortions of the image. The response of the watermark detector is 9.2325, 19.4182 and 31.8731 respectively, which is still well above random. The response again confirms that the algorithm is very robust to JPEG encoding distortions even while 95% to 75% of image data is compressed.

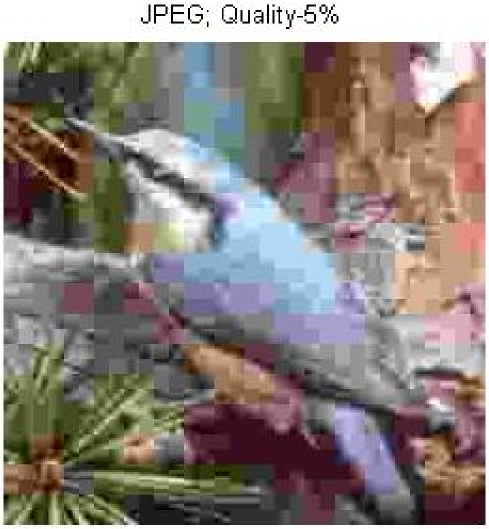


Figure 18(a): JPEG compressed version of “Bird” with 5% quality.

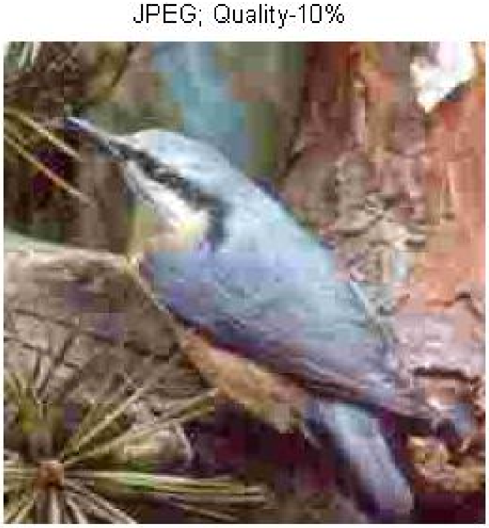


Figure 18(b): JPEG compressed version of “Bird” with 10% quality.

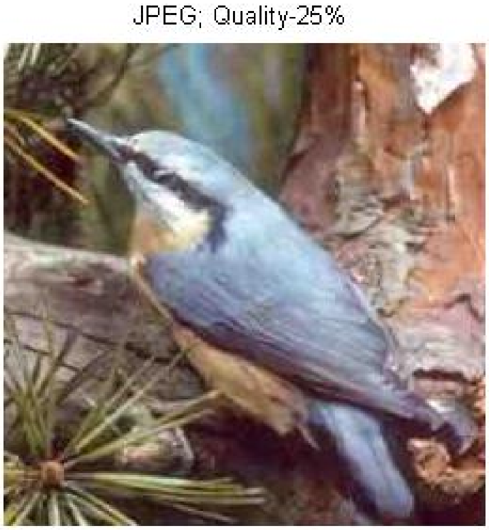


Figure 18(c): JPEG compressed version of “Bird” with 25% quality and 0% smoothing.

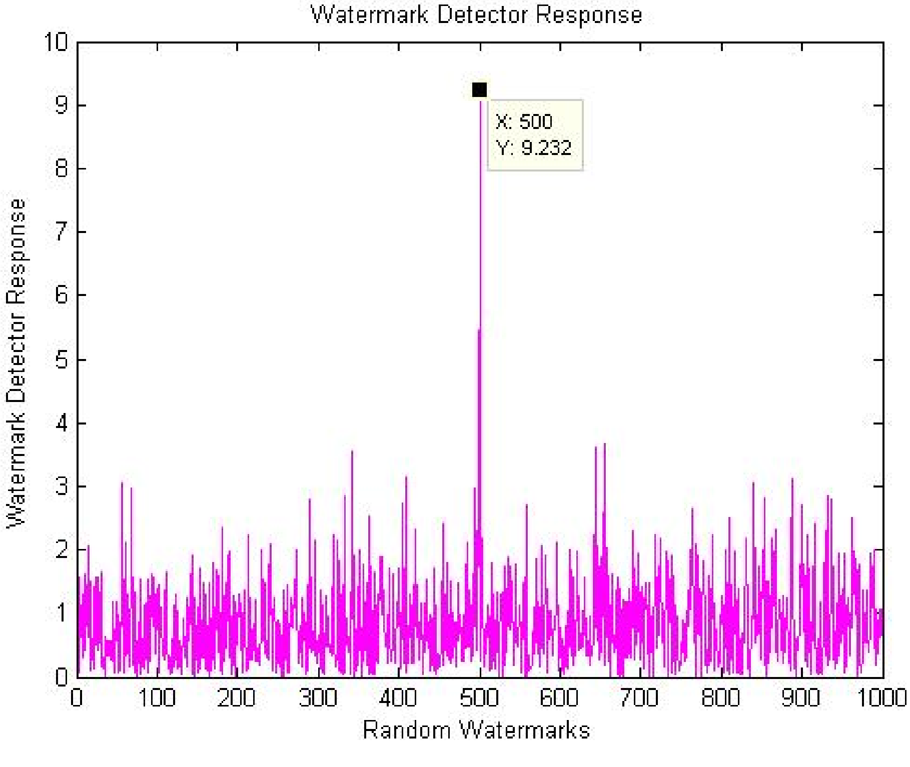


Figure 19(a): Watermark detector response to JPEG compressed image, with 5% quality and 0% smoothing in 18(a).

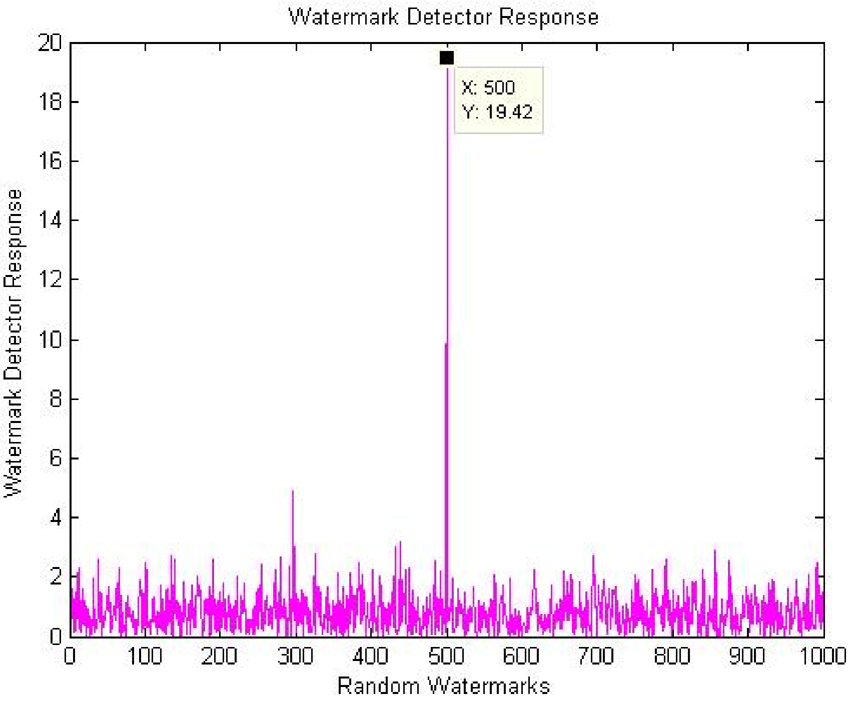


Figure 19(b): Watermark detector response to JPEG compressed image, with 10% quality and 0% smoothing in 18(b).

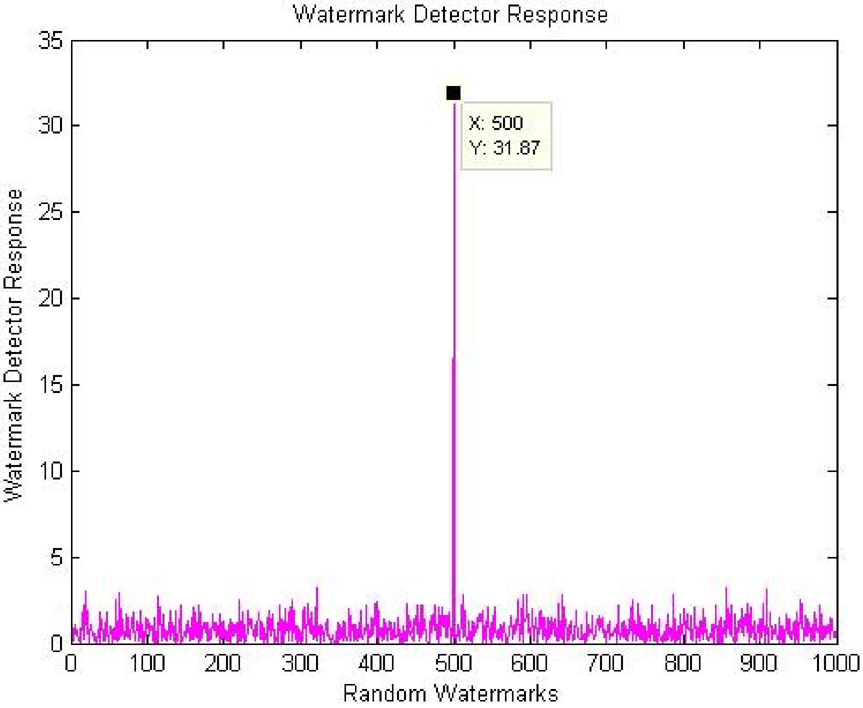


Figure 19(c): Watermark detector response to JPEG compressed image, with 25% quality and 0% smoothing in 18(c).

4.4: Experiment D: Dithering Distortion

Fig. 20(a) shows a dithered version of gray-scale “Horse” image. The response of the watermark detector shown in fig.20(b) is 19.3604, again proving that the algorithm is robust to common encoding distortions. In fact, more reliable detection can be achieved simply by removing any nonzero mean from the extracted watermark.

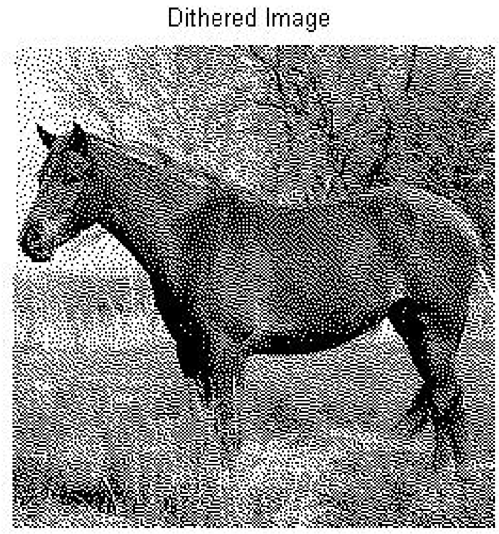


Fig. 20(a). Dithered version (8-scale) of the “Horse” image

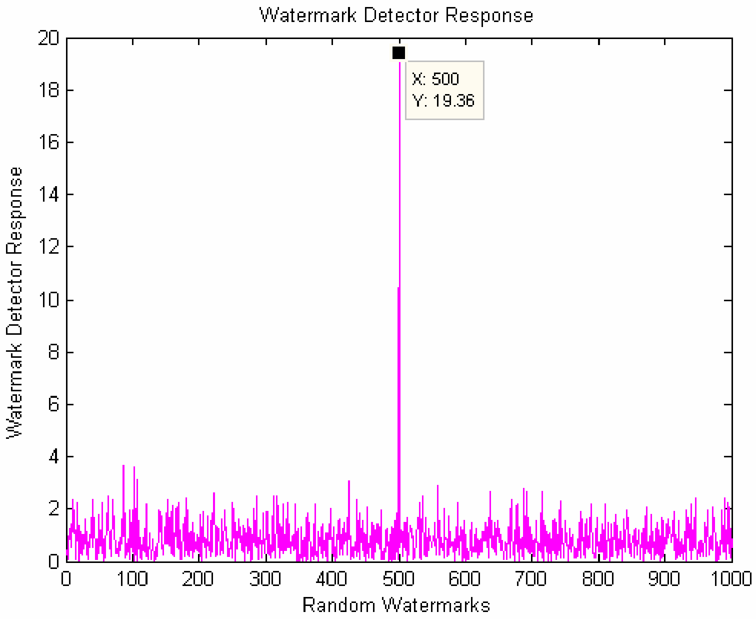


Fig. 20(b). Detector response to dithered image in 20(a).

Fig. 21(a) shows a dithered version of “Bird” (color) image. The response of the watermark detector shown in fig.21(b) is 11.493, again proving that the algorithm is robust to common encoding distortions. In fact, more reliable detection can be achieved simply by removing any nonzero mean from the extracted watermark.

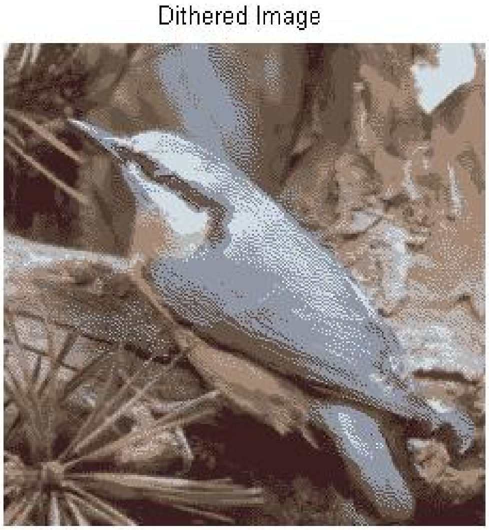


Fig. 21(a). Dithered version (8-scale) of the “Bird” image.

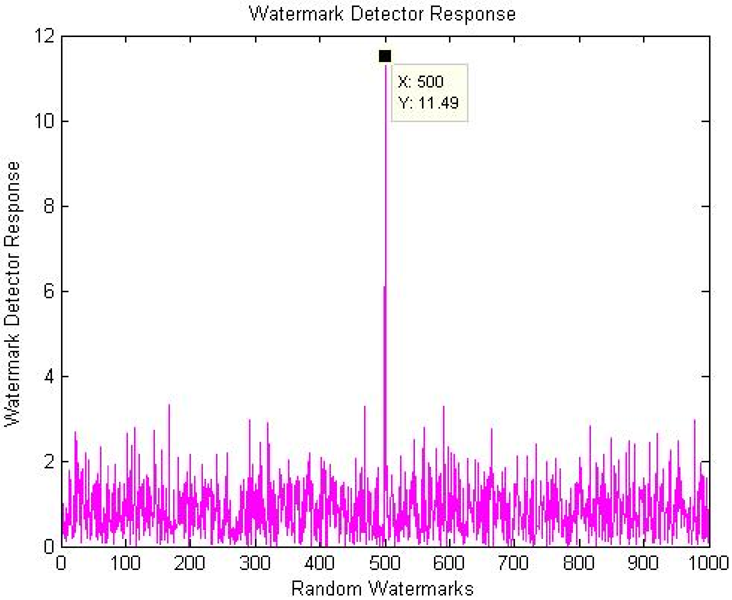


Fig. 21(b). Detector response to Dithered image in 21(a).

4.5: Experiment E: Rotation, Back-rotation, Cropping and Re-scaling

Figure 22(a) shows a rotated version of “Horse” by -5°. Fig. 22(b) is rotated-back, cropped and resized version of fig. 22(a); showing considerable distortions to the watermarked image. The response of the watermark detector is 9.2569, shown in fig. 24(a); again showing that the algorithm is robust to common encoding distortions.

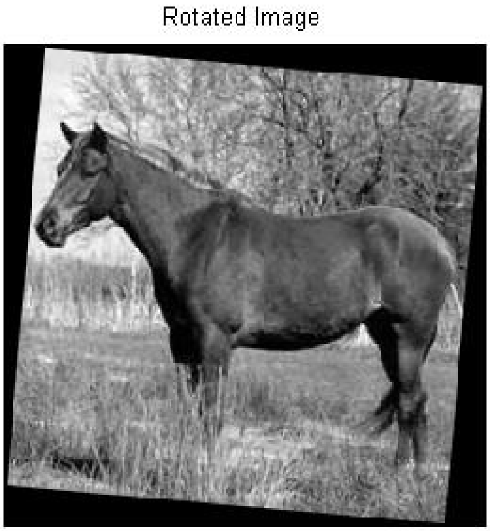


Figure 22(a): Watermarked gray-scale image rotated by - 5°.

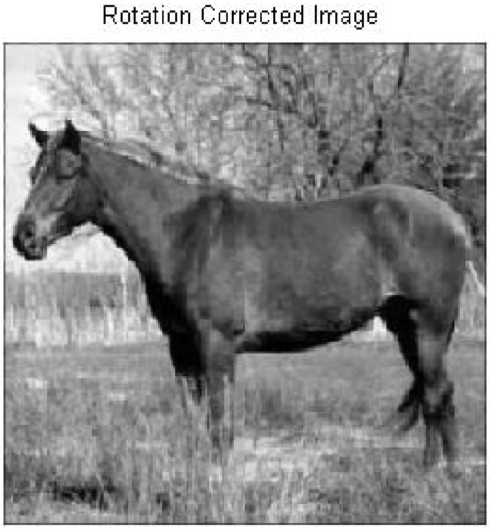


Figure 22(b): Image of figure 22(a) above; rotated back by +5°, then cropped and rescaled to original size, with visible distortions.

Similarly, fig.23(a) shows a rotated version of “Bird” by -5°. And fig.23(b) is rotated-back, cropped and resized version of fig.23(a); showing considerable distortions to the watermarked image. The response of the watermark detector is 11.582, shown in fig.24(b); again showing that the algorithm is robust to common encoding distortions.

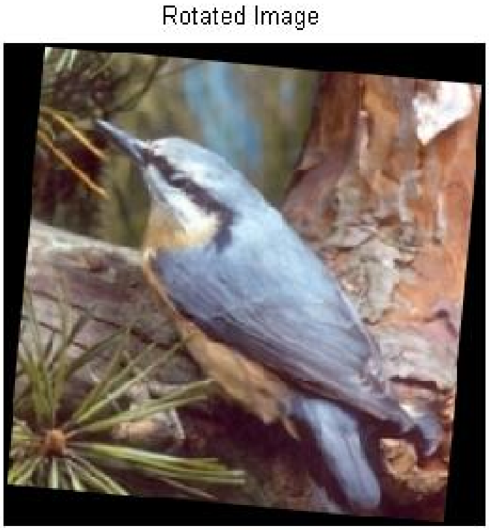


Figure 23(a): Watermarked color image rotated by - 5°.

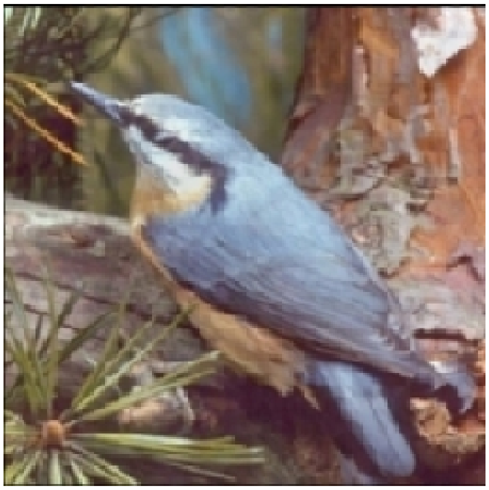


Figure 23(b): Image of figure 23(a) above; rotated back by +5°, then cropped and rescaled to original size, with visible distortions.

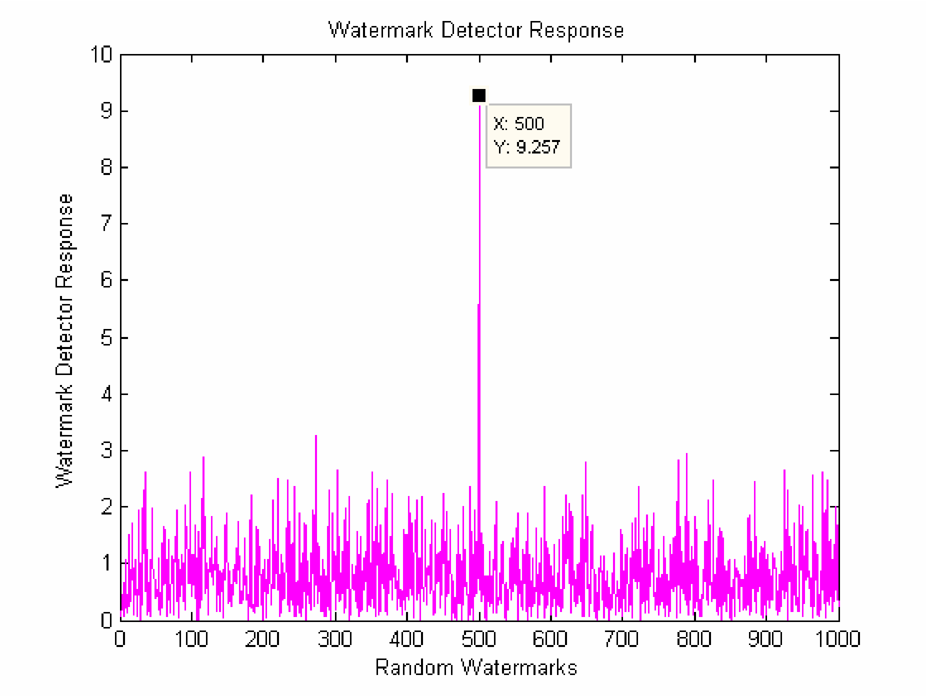


Figure 24(a): Watermark detector response to gray-scale image rotated back by +5°, then cropped and rescaled to original size shown in figure 22(b).

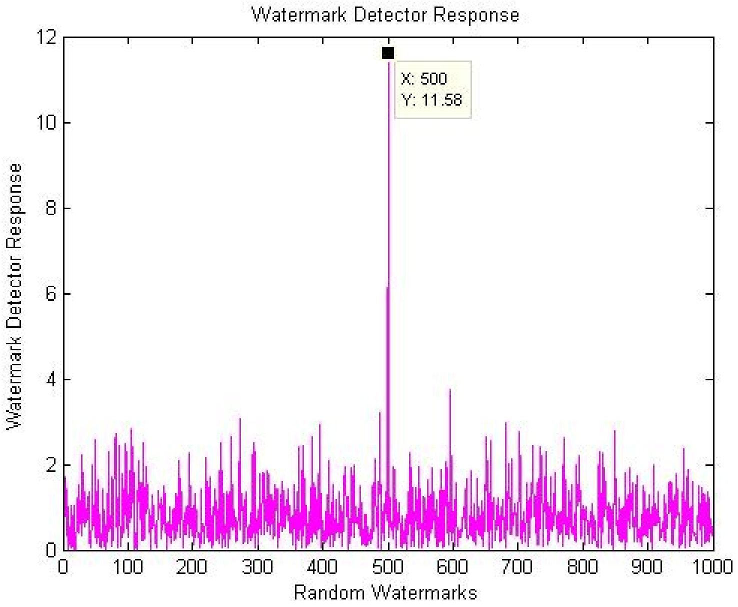


Figure 24(b): Watermark detector response to color image rotated back by +5°, then cropped and rescaled to original size as shown in figure 23(b).

4.6: Experiment F : Noise Attacks

Fig.25(a) and fig.25(b) show the gray-scale image of “Horse” attacked by noise, namely, “salt-n-pepper” and “A.W.G.N.” type noise, respectively. Fig. 26(a) and 26(b) show us that 11.8013 and 15.1190 are respective watermark detector responses to these attacks. This also confirms the watermark’s robustness to such noise attacks.

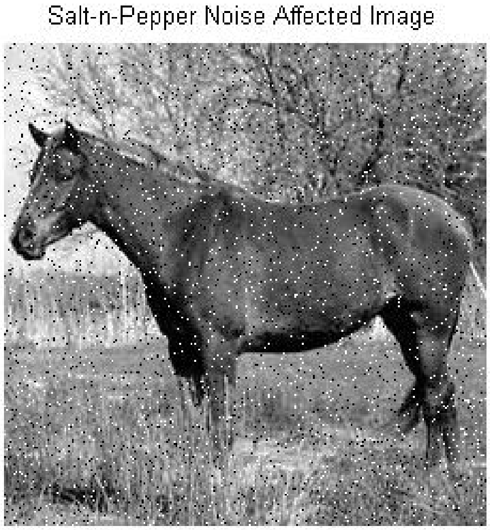


Figure 25(a): Gray-scale image attacked by “Salt-n-Pepper Noise”.

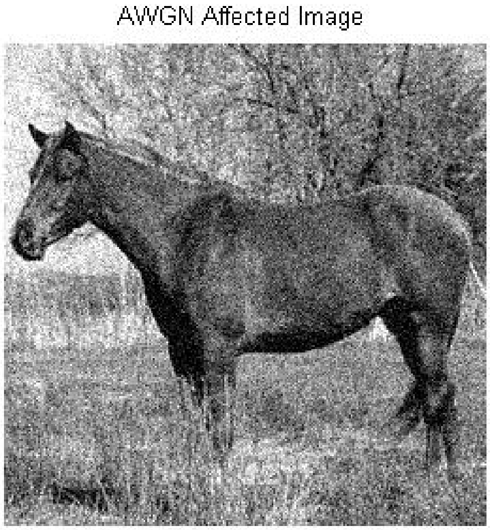


Figure 25(b): Gray-scale image attacked by “A.W.G. Noise”.

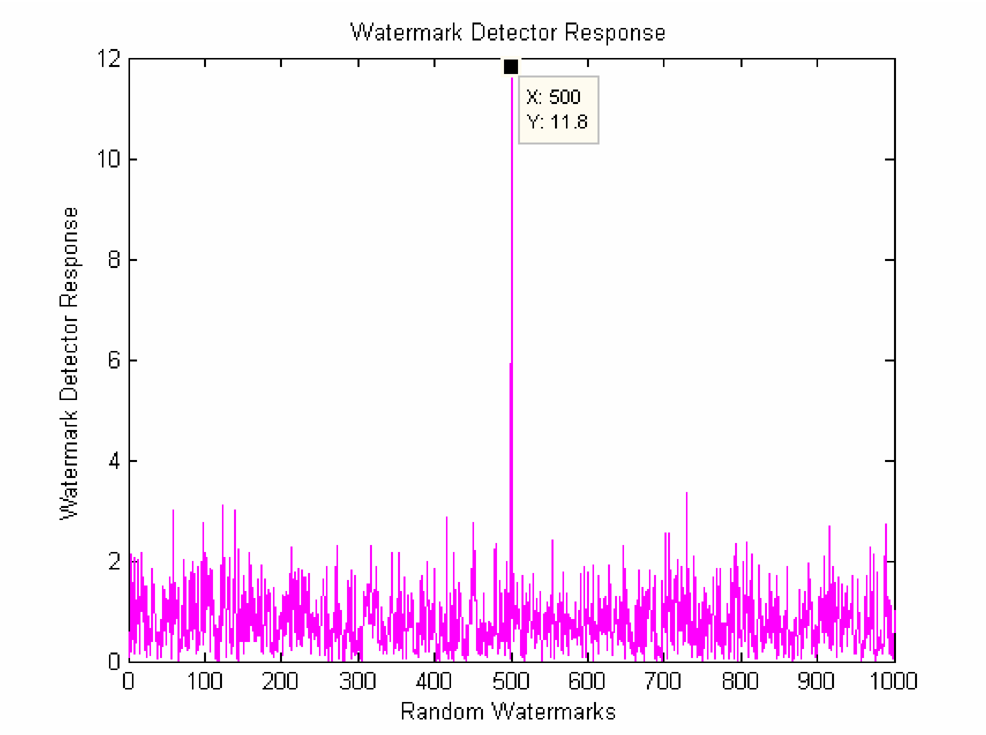


Figure 26(a): Detector Response to "Salt-n-Pepper Noise" attacked image.

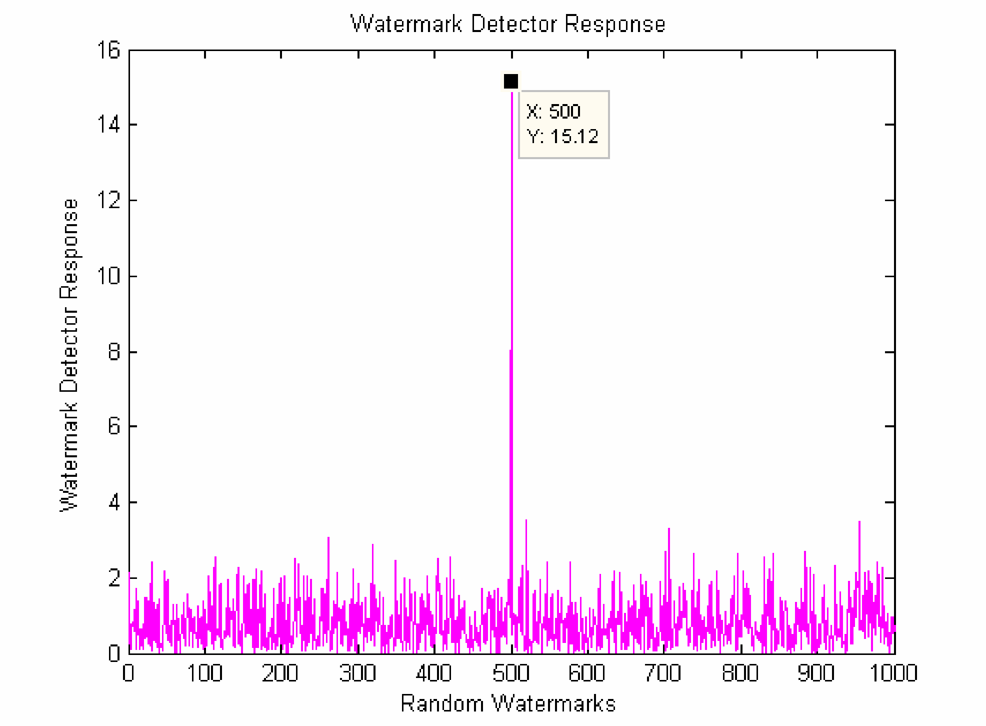


Figure 26(b): Detector Response to "A.W.G. Noise" attacked image.

Similarly, Fig.27(a) and fig.27(b) show the color image attacked by noise, namely, “salt-n-pepper” and “A.W.G.N.” type noise, respectively. Fig. 28(a) and 28(b) show us that 18.3711 and 22.4325 are respective watermark detector responses to these attacks. This again confirms the watermark’s robustness to such noise attacks.

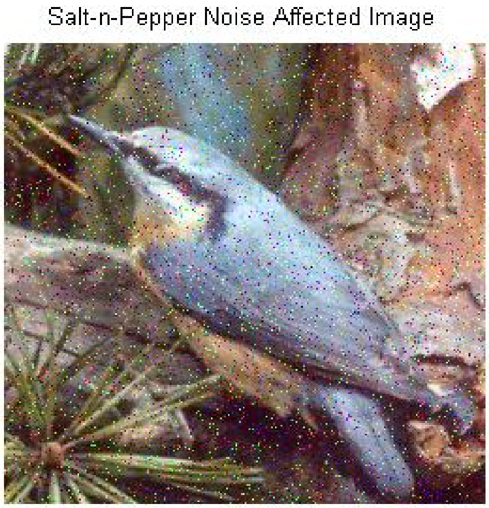


Figure 27(a): Color image attacked by “Salt-n-Pepper Noise”.

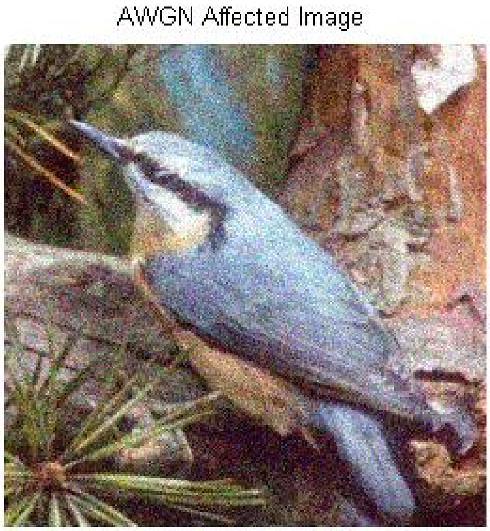


Figure 27(b): Color image attacked by “A.W.G. Noise”.

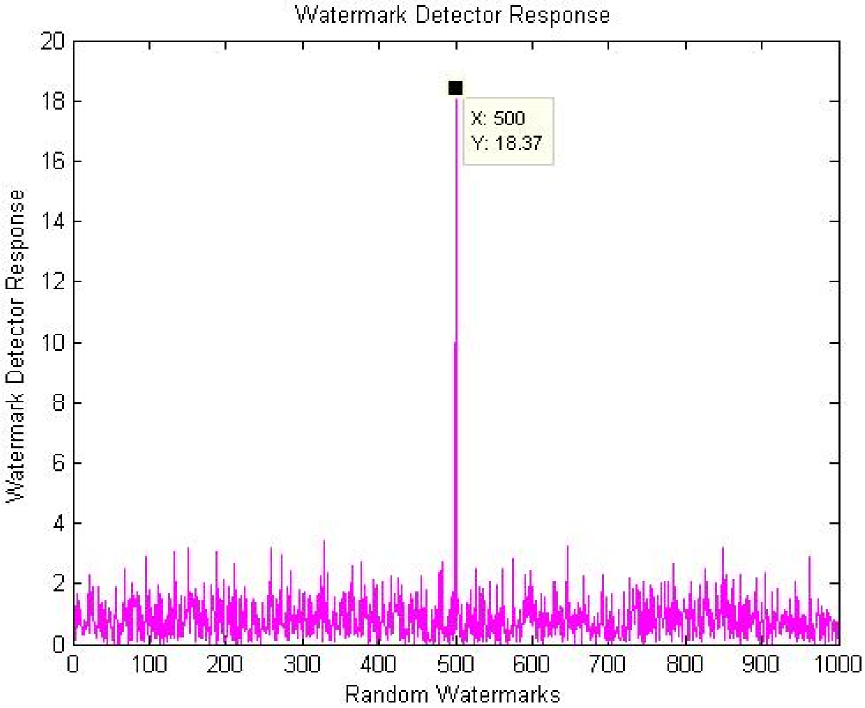


Figure 28(a): Detector response to "Salt-n-Pepper Noise" attacked color image.

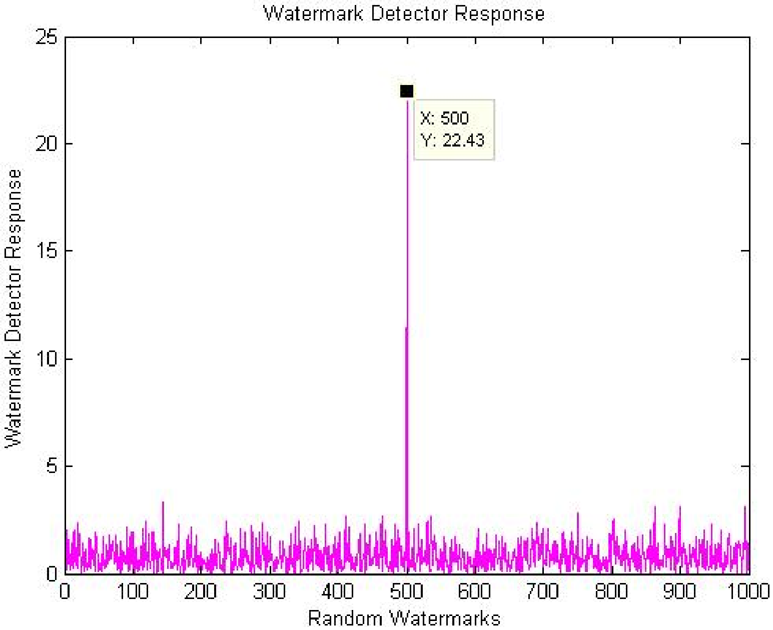


Figure 28(b): Detector response to "A.W.G. Noise" attacked color image.

4.7: Experiment G: Linear Filtering Attacks

Fig.29(a) shows the gray-scale “Horse” image attacked by “Average Filtering” and fig.29(b) shows the image attacked by “Median Filtering”. Fig. 30(a) and fig.30(b) show their detector response, i.e. 12.5543 and 18.9581 respectively. The results are well above the mean, confirming our watermark’s resilience to such attacks.

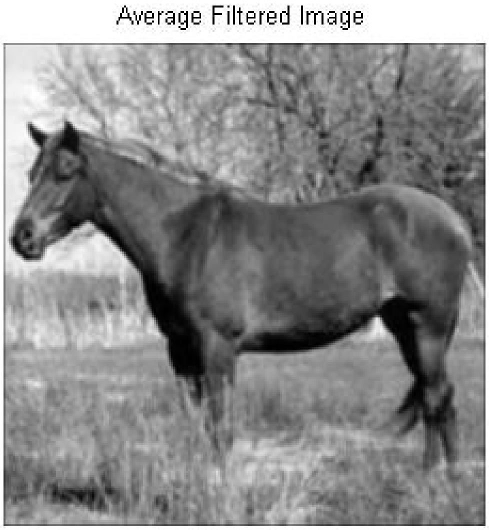


Figure 29(a): Gray-scale image after Average Filtering.

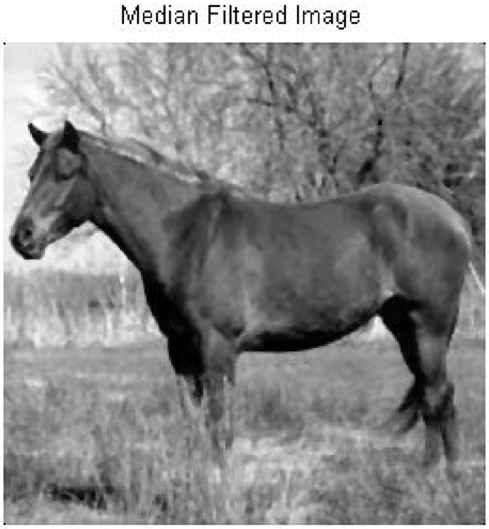


Figure 29(b): Gray-scale image after Median Filtering.

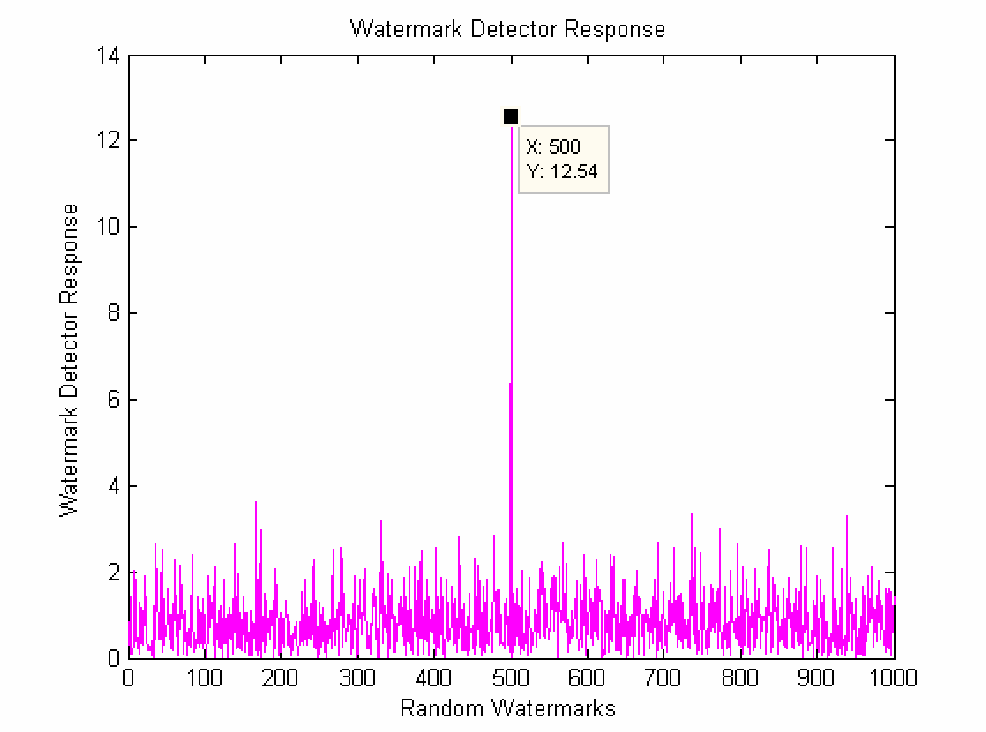


Figure 30(a): Detector response to Average Filtered grayscale image.

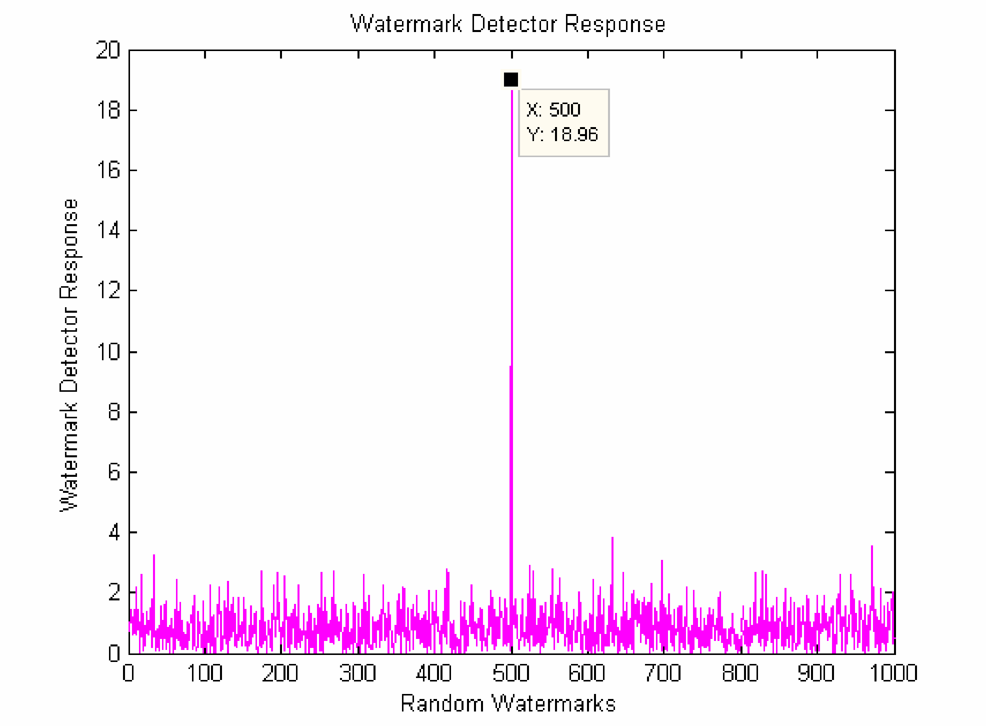


Figure 30(b): Detector response to Median Filtered gray-scale image.

Similarly, fig.31(a) shows the color image of “Bird” attacked by “Average Filtering” and fig.31(b) shows the image attacked by “Median Filtering”. Fig. 32(a) and fig.32(b) show their detector response, i.e. 18.169 and 27.7176 respectively. The results are well above the mean, confirming our watermark’s resilience to such attacks.

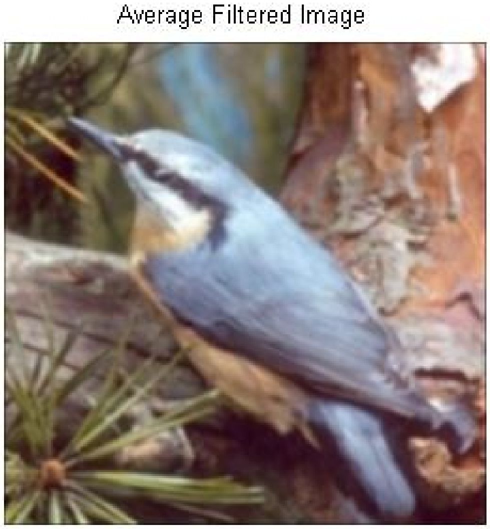


Figure 31(a): Color image after Average Filtering.

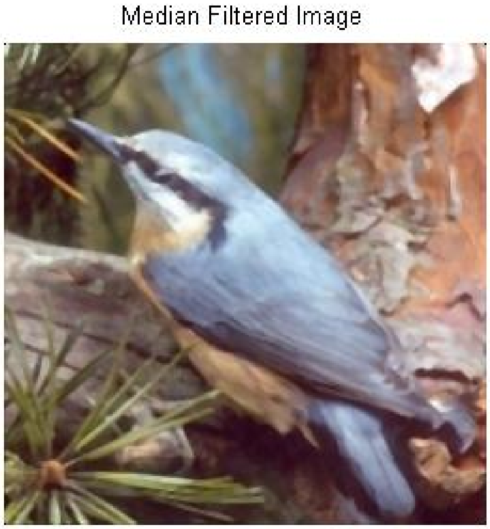


Figure 31(b): Color image after Median Filtering.

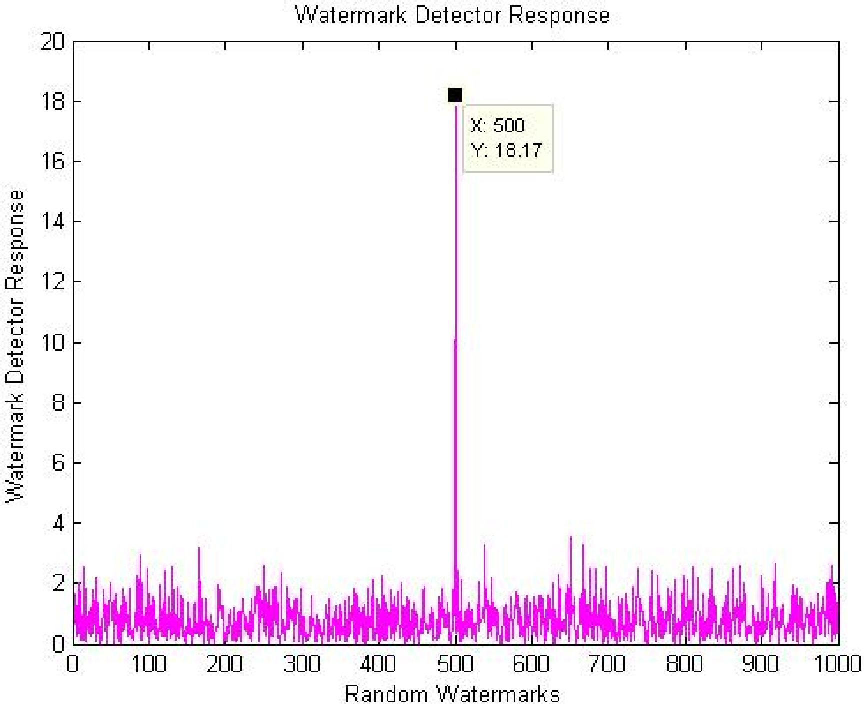


Figure 32(a): Detector response to Average Filtered color image

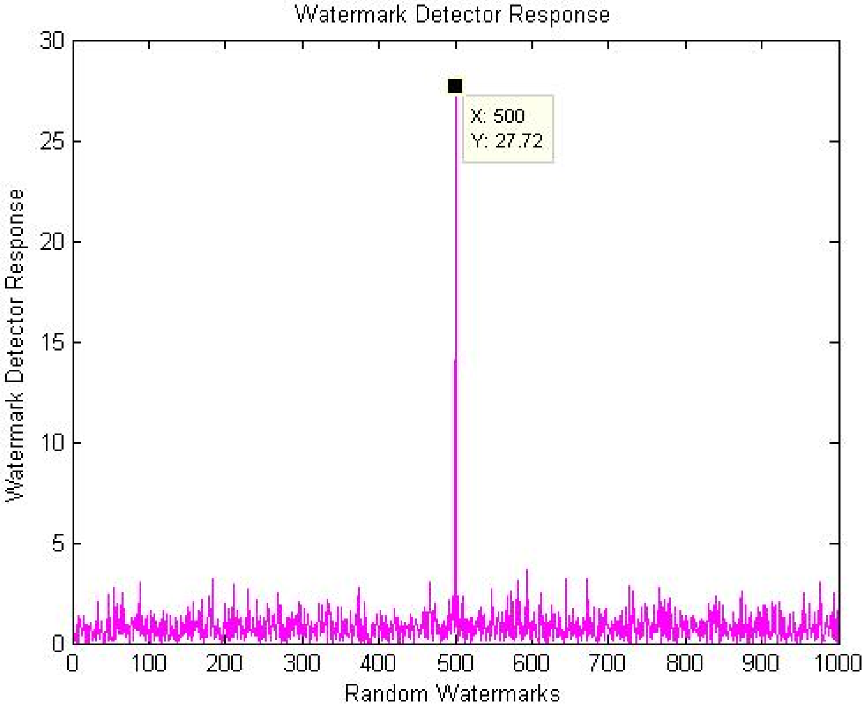


Figure 32(b): Detector Response to Median Filtered color image

The **Table 1** shown below summarizes the results obtained regarding the watermark detector response to attacked images using Cox's spread-spectrum watermarking algorithm with random watermark, at α = 0.1. "Similarity" indicates the watermark detector's response.

.

S.No.	**Attack**	**SNR (dB)** (Gray-scale image)	**Similarity** (Gray-scale image)	**SNR (dB)** (Color image)	**Similarity** (Color image)
1	No Attack	28.4292	32.2088	26.6813	37.1567
2	Re-sized	19.0155	13.2660	22.4245	18.9143
3	JPEG (5%)	16.7325	09.9455	18.5299	09.2325
4	JPEG (10%)	18.6677	17.7398	21.0434	19.4182
5	JPEG (25%)	21.1555	28.2960	23.4578	31.8731
6	Rotation & Cropped	17.0177	09.2569	19.6212	11.5820
7	De-blurred	21.4644	19.1376	23.1440	18.8408
8	Salt-pepper Noise	12.6447	11.8013	15.4174	18.3711
9	AWG Noise	14.3630	15.1190	17.1041	22.4325
10	Average filtered	18.8015	12.5443	22.0689	18.1690
11	Median filtered	20.1196	18.9581	24.0866	27.7176
12	Dithered(8 colors)	01.2828	19.3604	16.6114	11.4930

Table1: Experimental results obtained with 256x256 gray-scale and color images, using a random watermark, by applying Cox's Spread-Spectrum Watermarking Algorithm.

4.8: Experiment H: Wiener Attacks by MMSE estimation

Lastly, the random watermark embedded image is subjected to **"Wiener-Attacks"**; as will be explained in following sections. We observe that by applying Wiener attack, the attacker minimizes the mean-squared-error estimated over the whole image and hence the visual quality of the attacked image is also degraded.

Figures 33(a) and 33(j) show the gray-scale images subjected to Wiener attacks; in each γ is incremented by 0.2. Figures 34(a) to 34(j) show the detector response to these images respectively. At γ =1 (Removal attack) the response falls to a similarity value of 7.5379 for gray-scale image and for the color image it becomes 11.6698.

The value of the scaling factor γ is increased from 0 to 2, in equal steps of 0.2, and the detector's response is recorded. It is also observed that the image quality starts degrading drastically on increasing γ value beyond 1, which can be easily detected by normal human vision. **Surprisingly, our random watermark fails to withstand such tricky attacks**. Such deliberately performed attacks pull our attention again towards the robustness issue of watermarking scheme, and hence we are motivated to adopt the idea of power-spectrum compliant (i.e. energy-efficient) watermark; as will be discussed in details in next chapter.

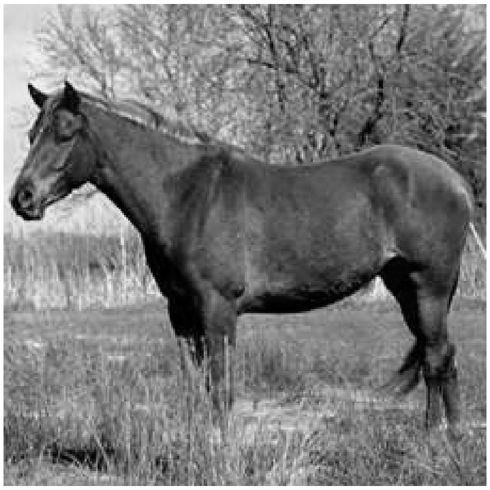


Figure 33(a): Wiener attacked (γ = 0.2) gray-scale image.

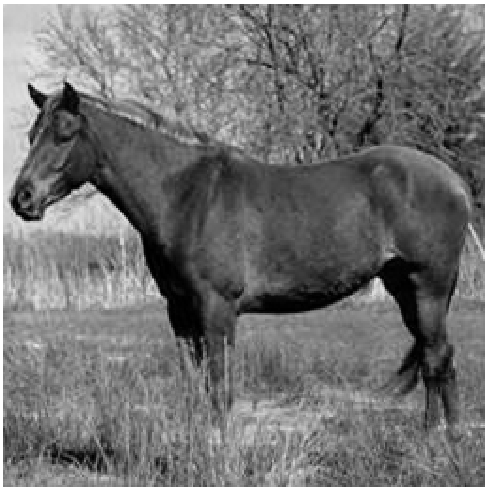


Figure 33(b): Wiener attacked (γ = 0.4) gray-scale image.

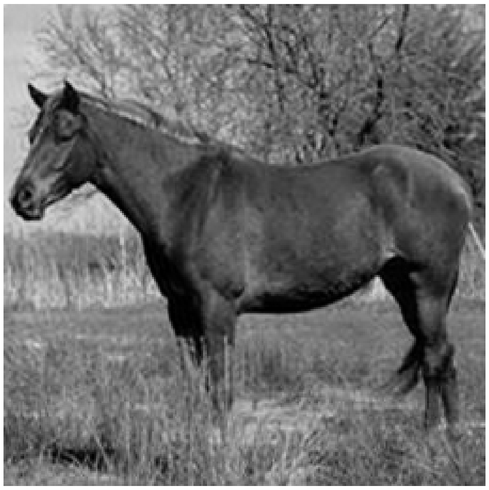


Figure 33(c): Wiener attacked (γ = 0.6) gray-scale image.

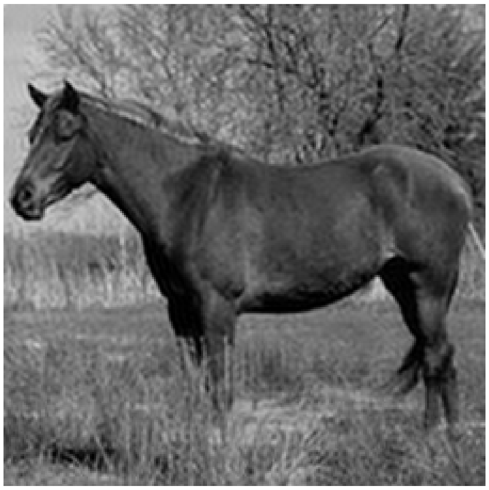


Figure 33(d): Wiener attacked (γ = 0.8) gray-scale image.

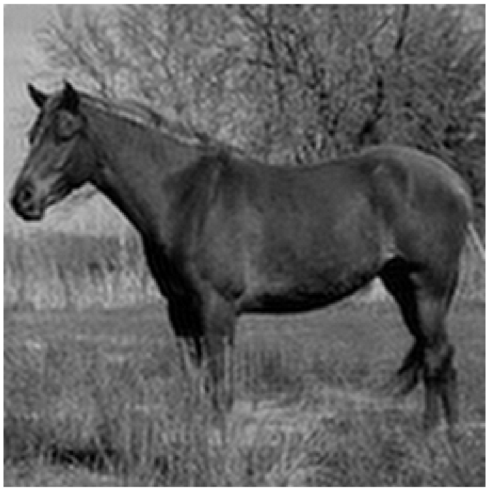


Figure 33(e): Wiener attacked (γ = 1.0) gray-scale image.

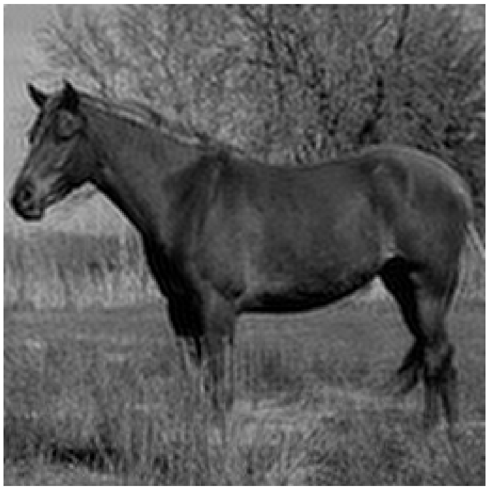


Figure 33(f): Wiener attacked (γ = 1.2) gray-scale image.

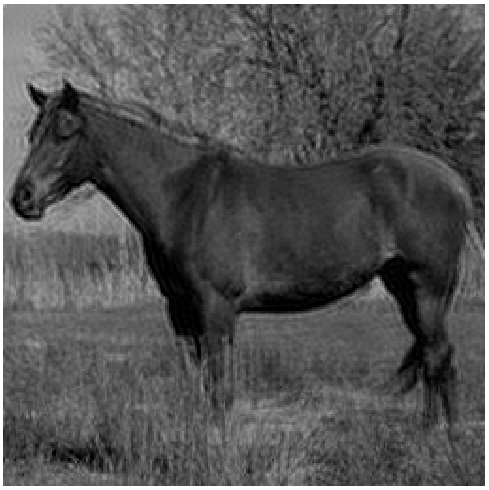


Figure 33(g): Wiener attacked (γ = 1.4) gray-scale image.

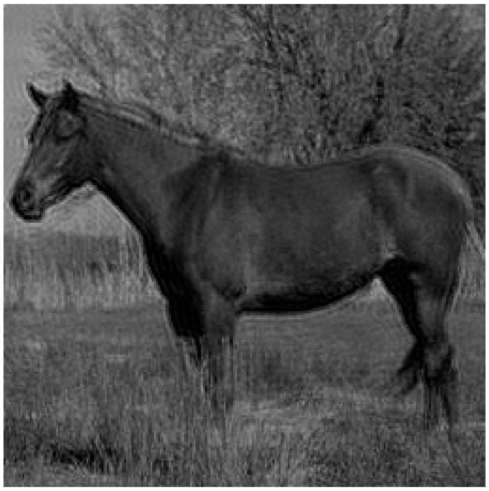


Figure 33(h): Wiener attacked (γ = 1.6) gray-scale image.

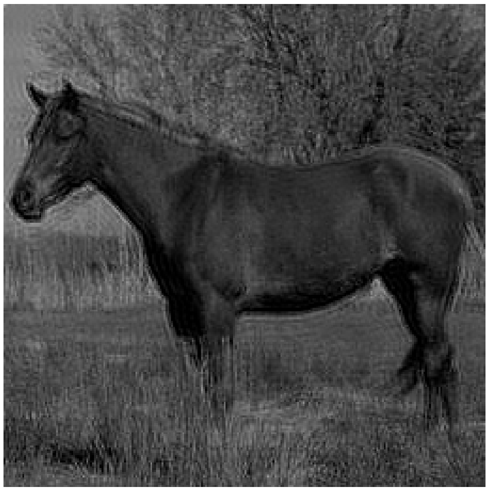


Figure 33(i): Wiener attacked (γ = 1.8) gray-scale image.

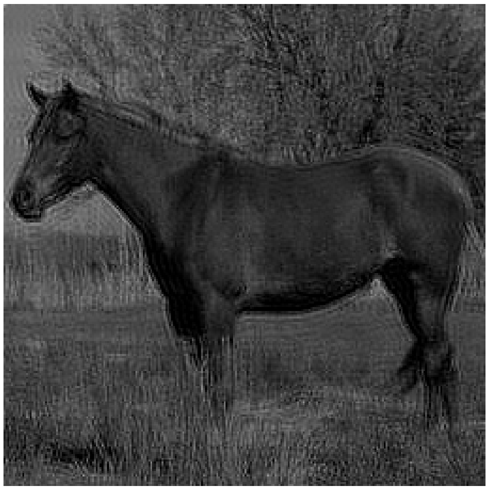


Figure 33(j): Wiener attacked (γ = 2.0) gray-scale image.

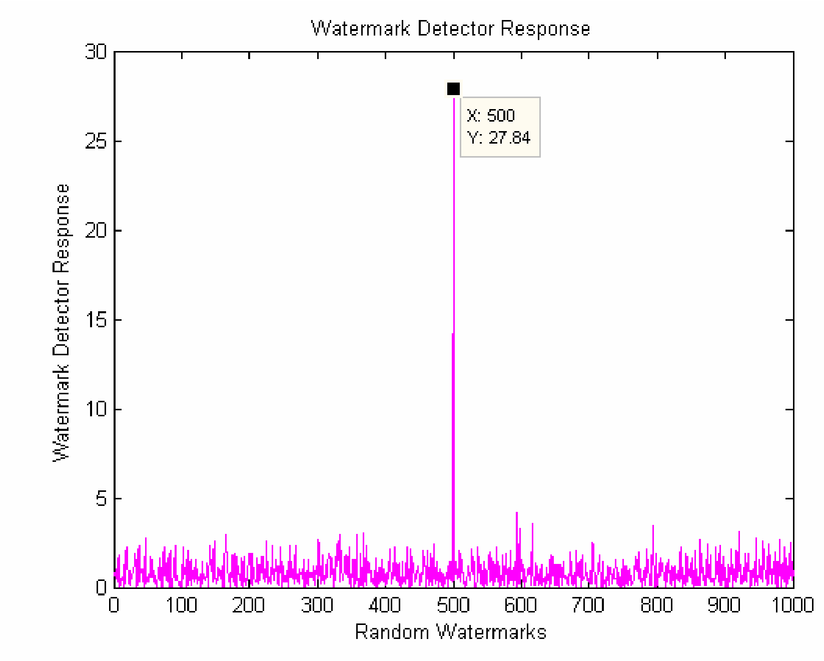


Figure 34(a): Response to Wiener attacked (γ = 0.2) gray-scale image.

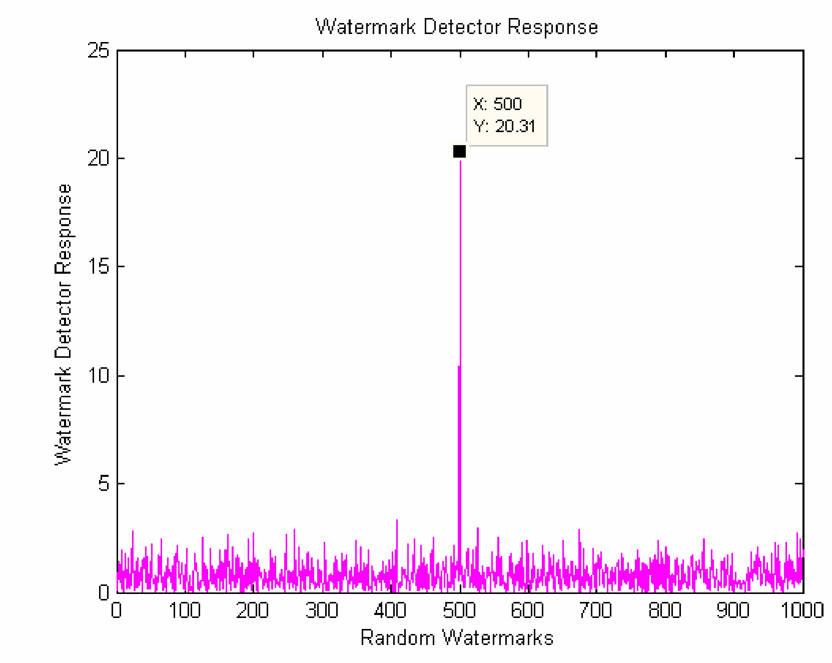


Figure 34(b): Response to Wiener attacked (γ = 0.4) gray-scale image.

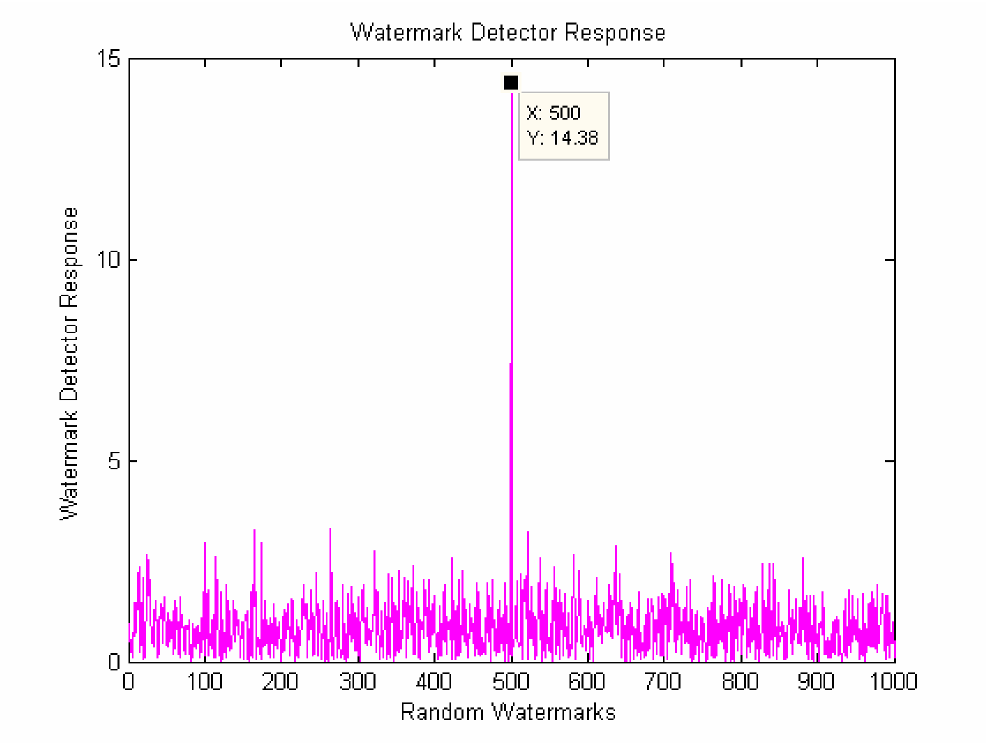


Figure 34(c): Response to Wiener attacked (γ = 0.6) gray-scale image.

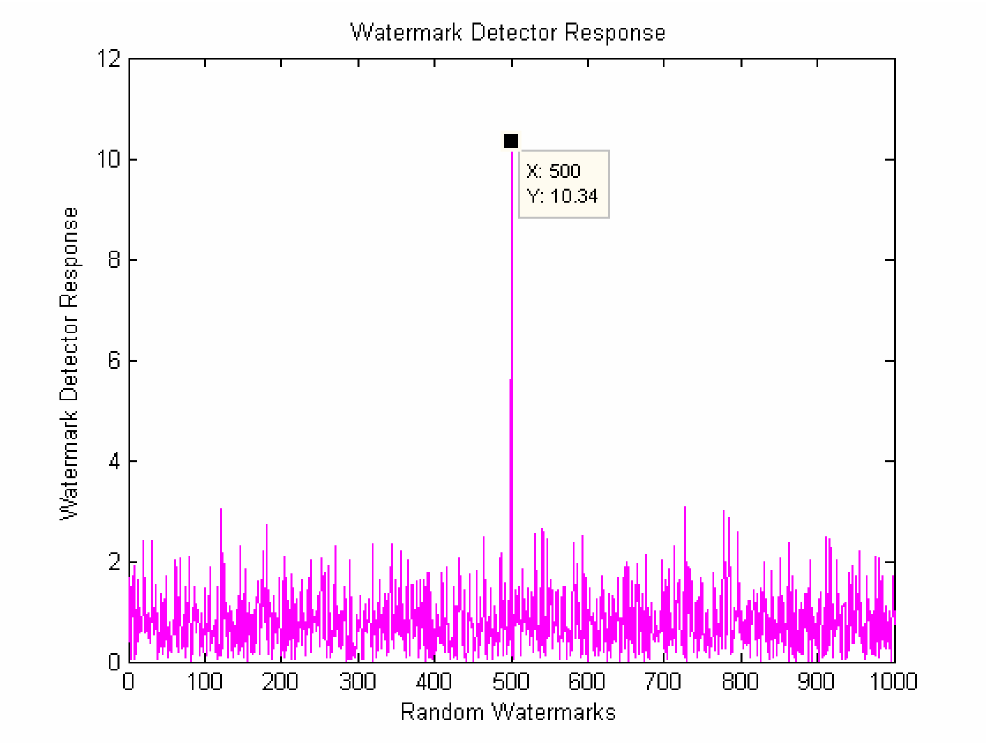


Figure 34(d): Response to Wiener attacked (γ = 0.8) gray-scale image.

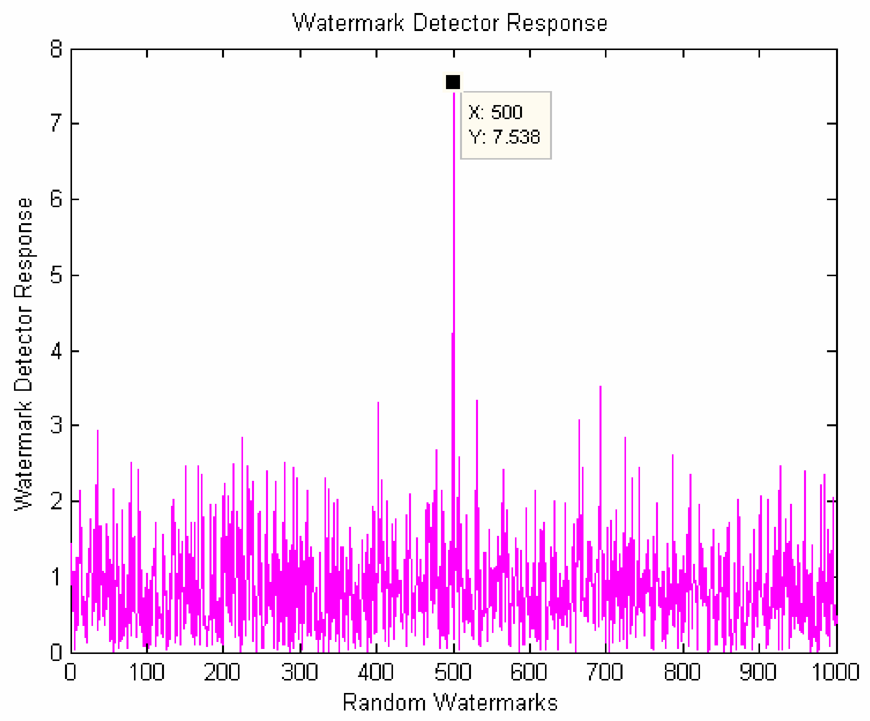


Figure 34(e): Response to Wiener attacked (γ = 1.0) gray-scale image.

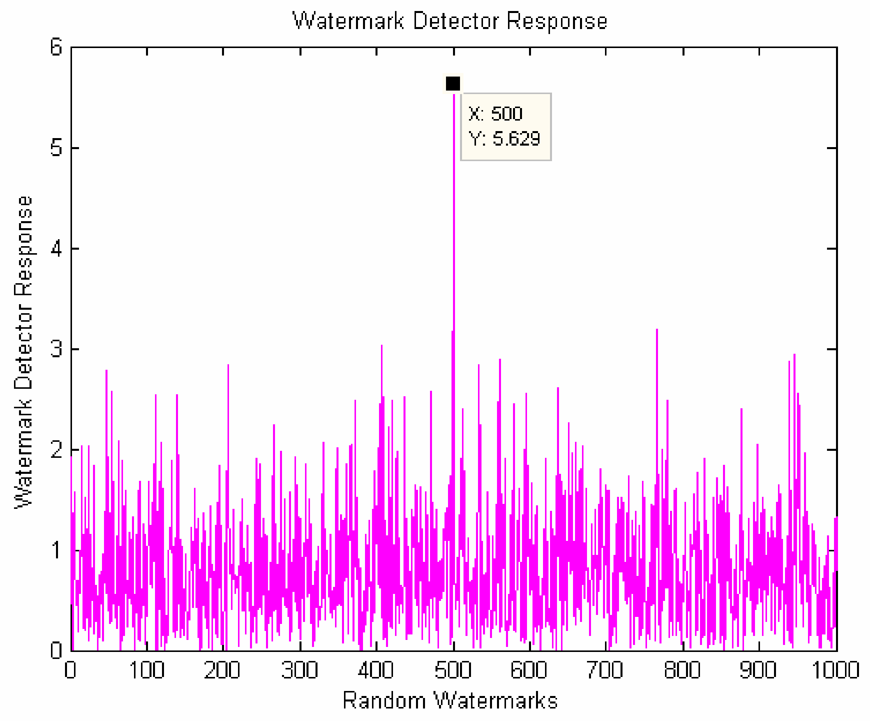


Figure 34(f): Response to Wiener attacked (γ = 1.2) gray-scale image.

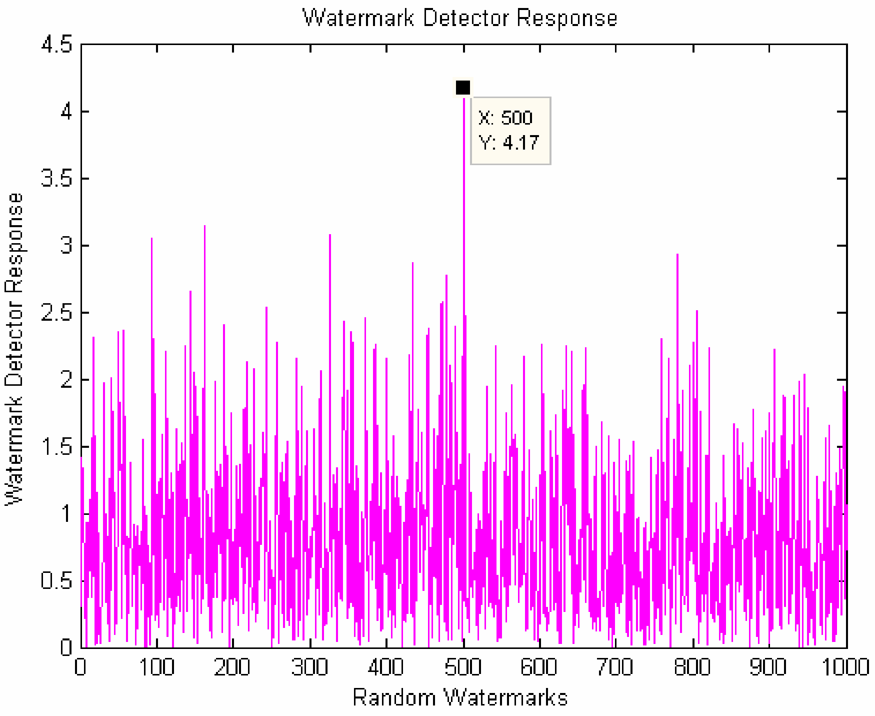


Figure 34(g): Response to Wiener attacked (γ = 1.4) gray-scale image.

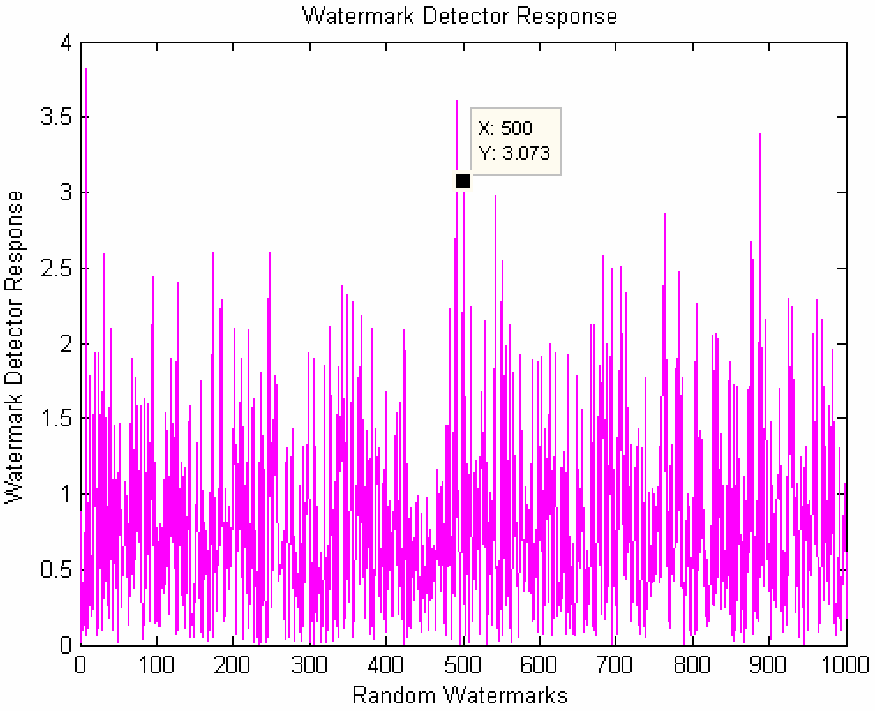


Figure 34(h): Response to Wiener attacked (γ = 1.6) gray-scale image.

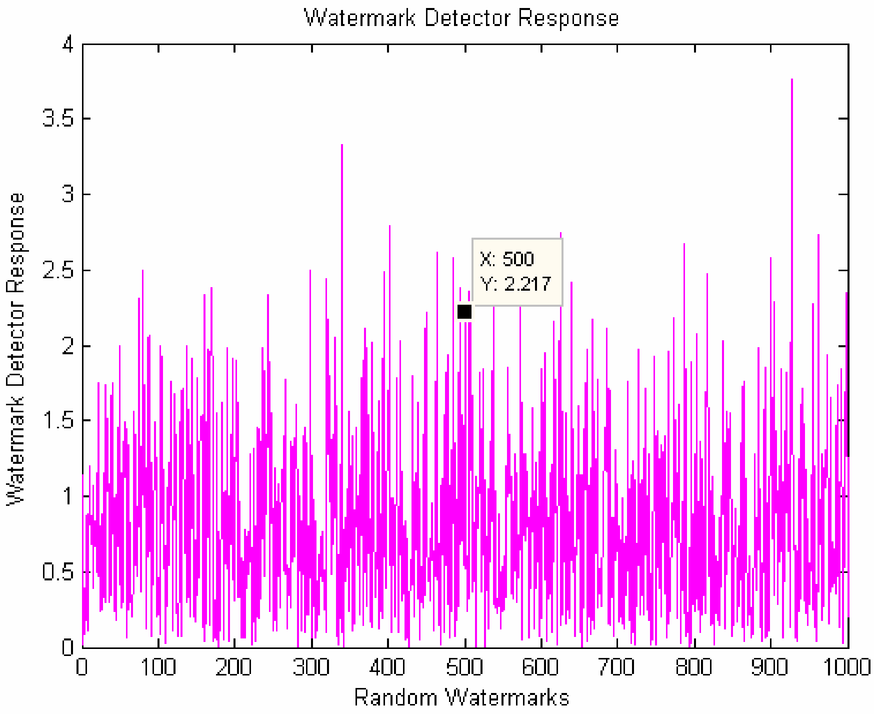


Figure 34(i): Response to Wiener attacked (γ = 1.8) gray-scale image.

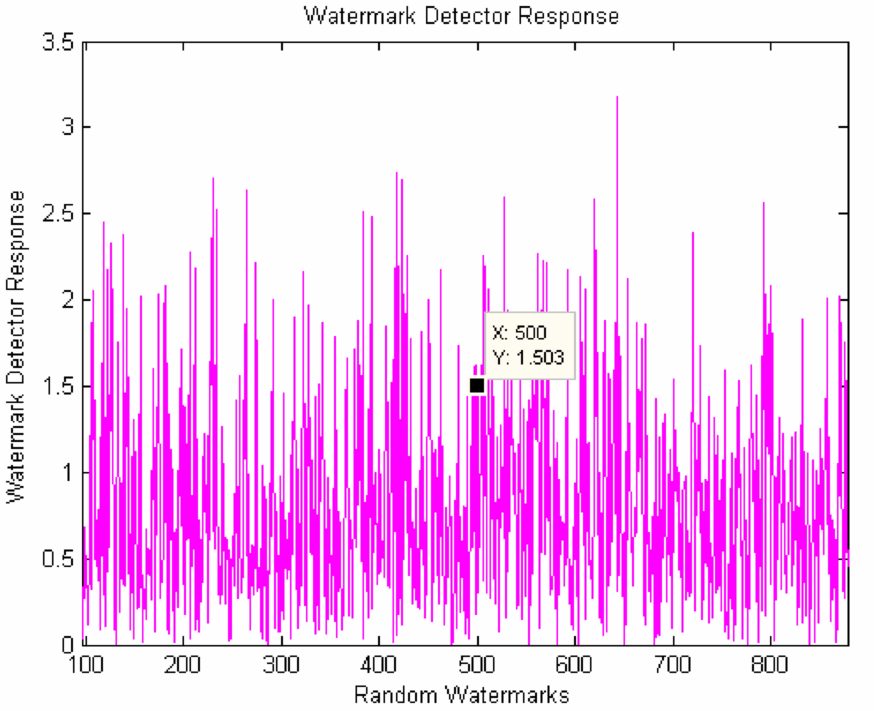


Figure 34(j): Response to Wiener attacked (γ = 2.0) gray-scale image.

The similar Weiner attacks were also conducted on the color images and results were recorded. Attacked images are shown in figures 35(a) to 35(j). Similar degradation in the visual quality along with poorer detector responses (as shown in figures 36(a) to 36(j) respectively) were observed, if we increase γ value. These results are recorded in Table 2.

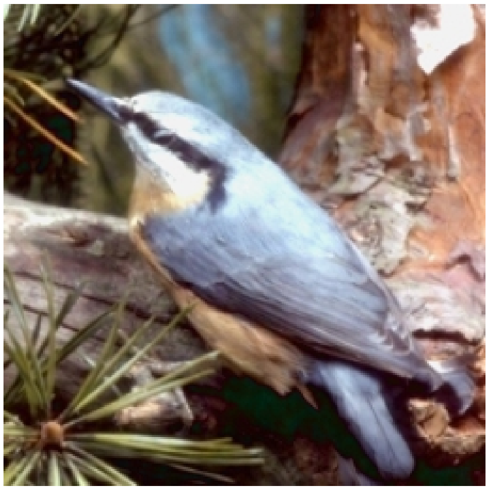


Figure 35(a): Wiener attacked (γ = 0.2) color image.

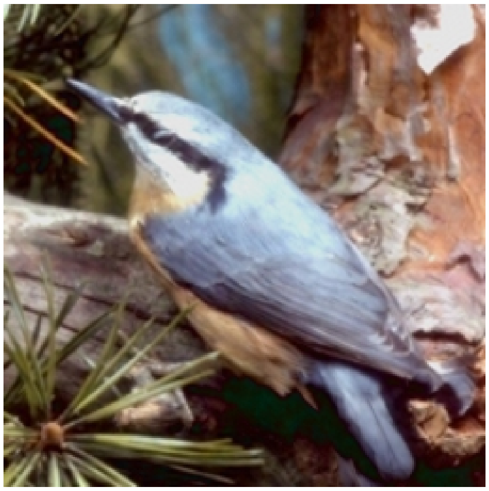


Figure 35(b): Wiener attacked (γ = 0.4) color image.

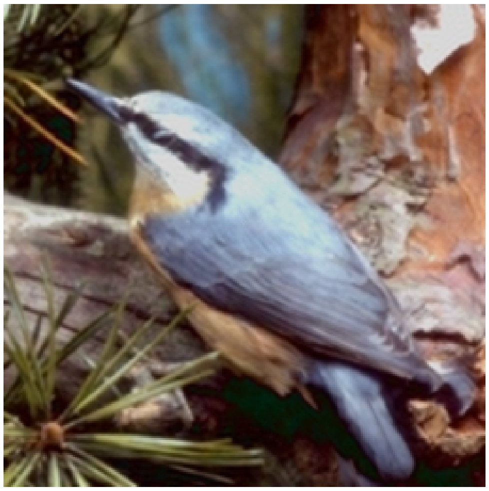


Figure 35(c): Wiener attacked (γ = 0.6) color image.

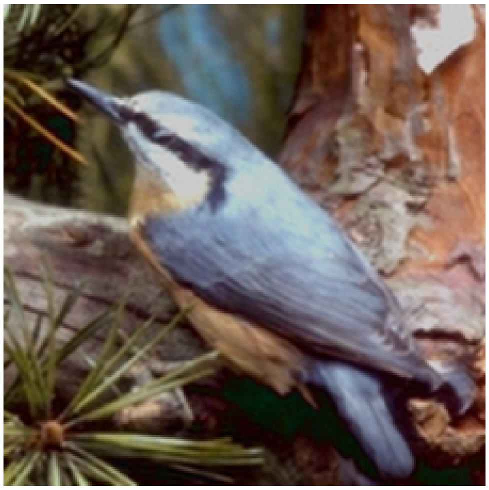


Figure 35(d): Wiener attacked (γ = 0.8) color image.

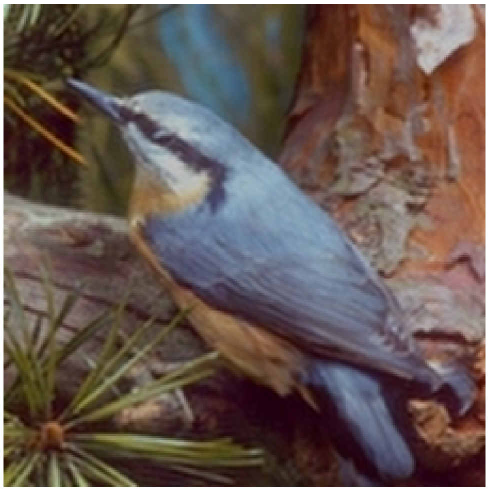


Figure 35(e): Wiener attacked (γ = 1) color image.

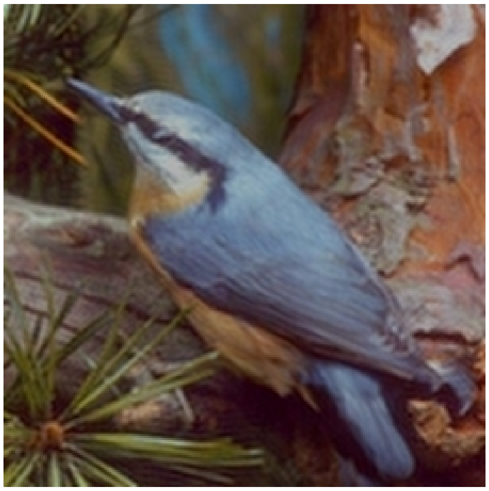


Figure 35(f): Wiener attacked (γ = 1.2) color image.

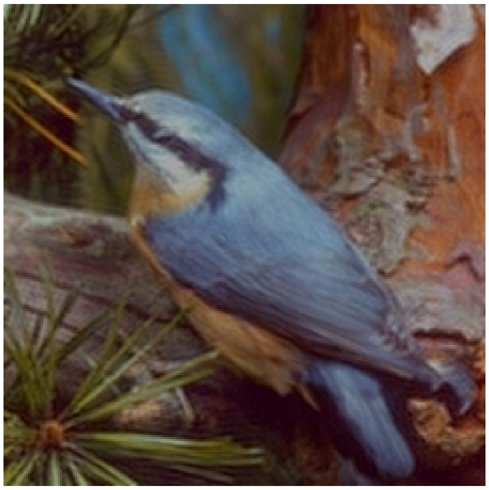


Figure 35(g): Wiener attacked (γ = 1.4) color image.

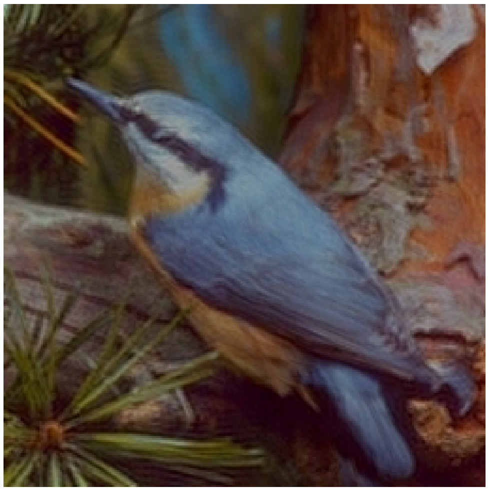


Figure 35(h): Wiener attacked (γ = 1.6) color image.

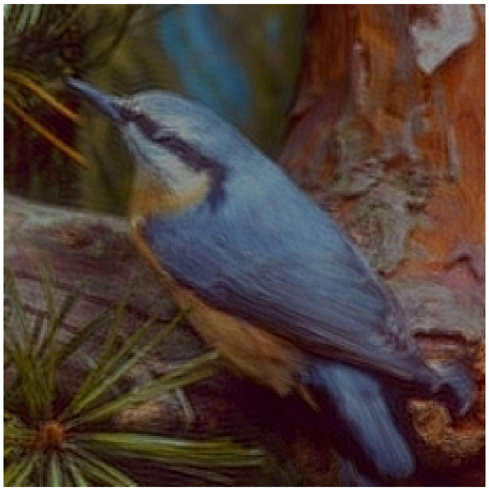


Figure 35(i): Wiener attacked (γ = 1.8) color image.

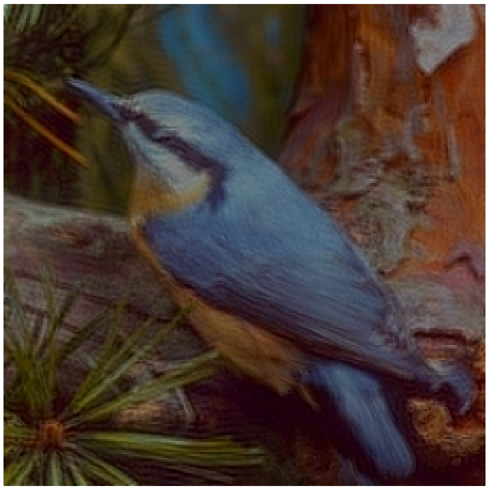


Figure 35(j): Wiener attacked (γ = 2.0) gray-scale image.

.

The watermark detector responses for images shown in figures 35(a) to 35(j) are shown in figures 36(a) to 36(j) respectively. From these figures one can clearly analyze the effect of Wiener filter to a watermarked image. It is also apparent that not only the detector response, but the visual quality of the images also degrade, particularly when the value of γ is increased beyond unity.

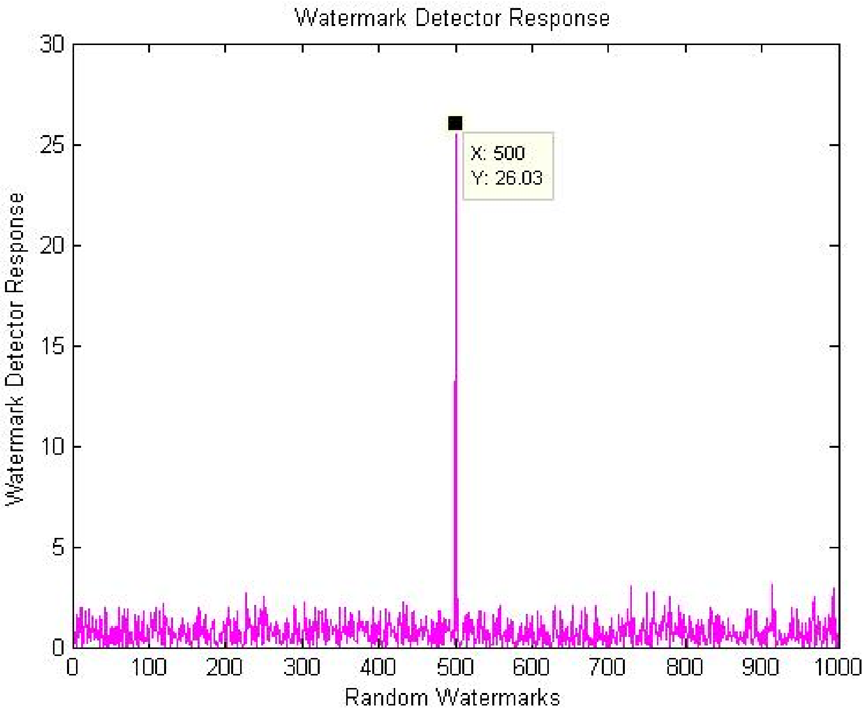


Figure 36(a): Response to Wiener attacked (γ = 0.2) color image.

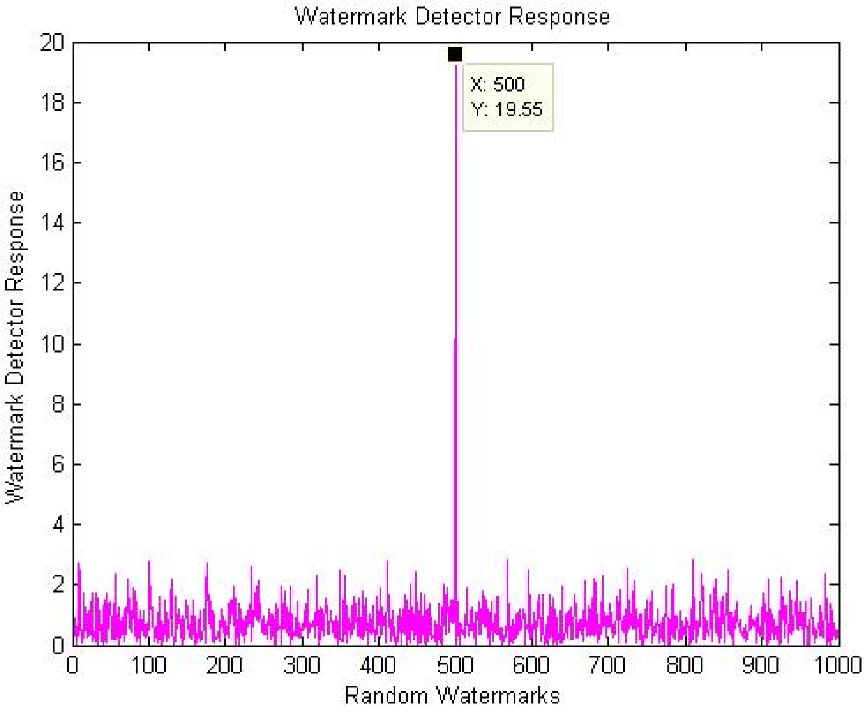


Figure 36(b): Response to Wiener attacked (γ = 0.4) color image.

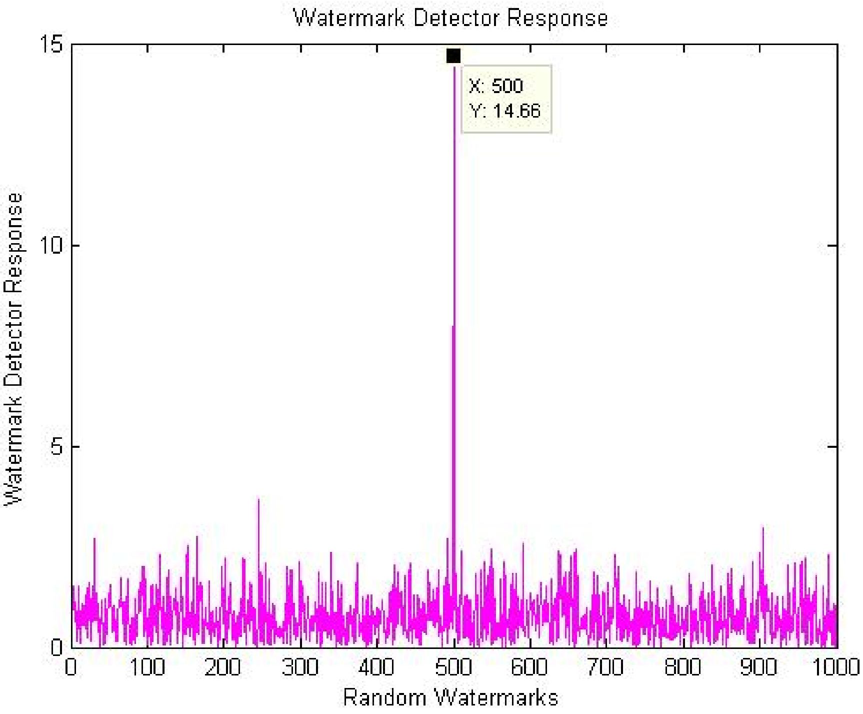


Figure 36(c): Response to Wiener attacked (γ = 0.6) color image.

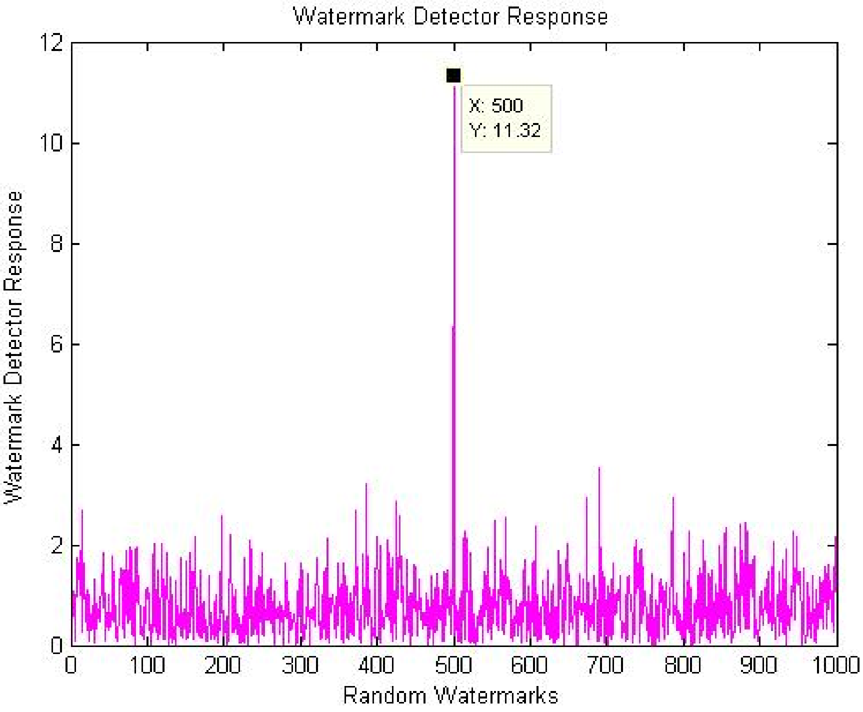


Figure 36(d): Response to Wiener attacked (γ = 0.8) color image.

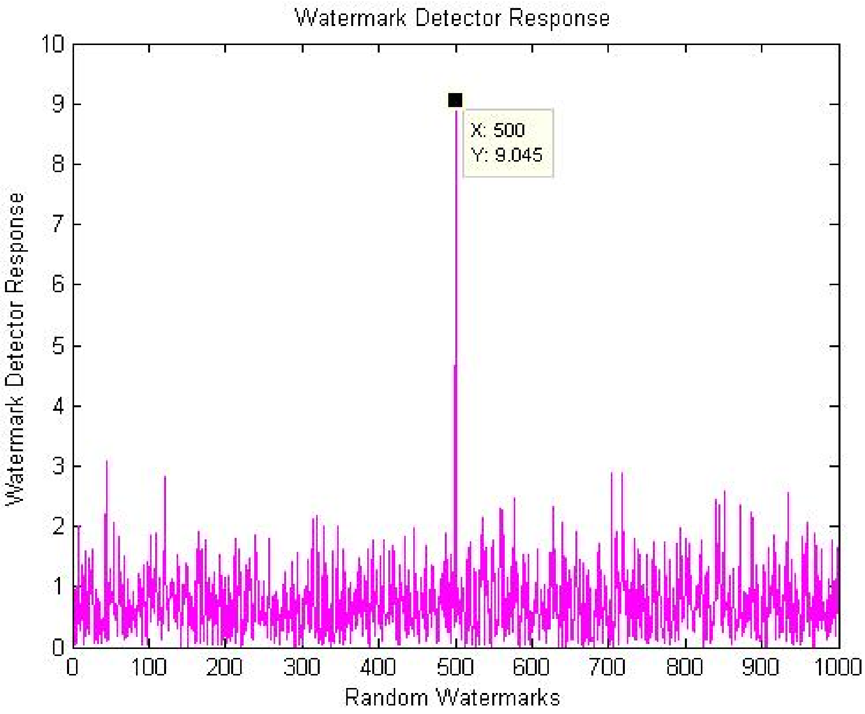


Figure 36(e): Response to Wiener attacked (γ = 1.0) color image.

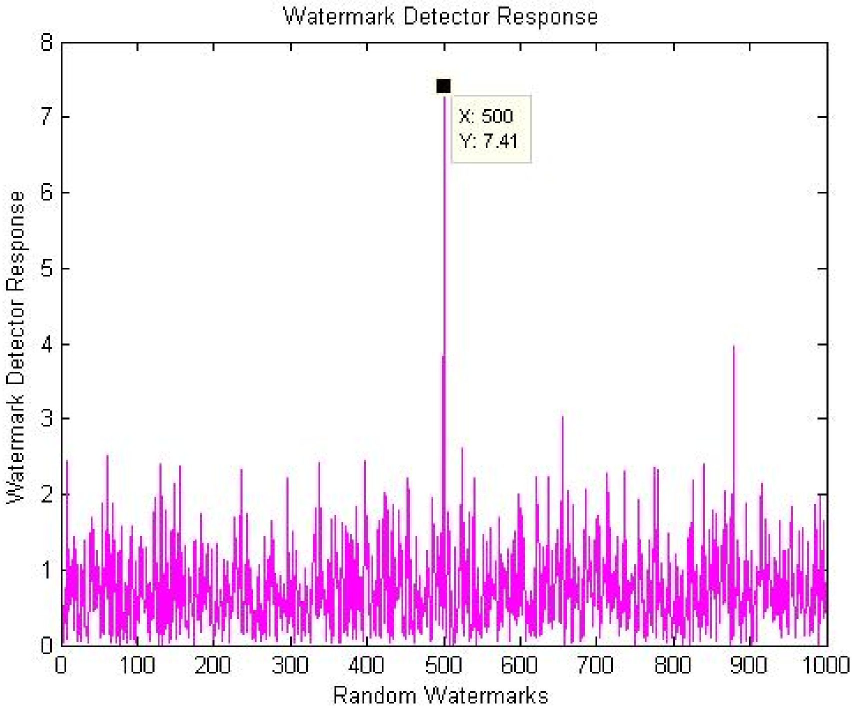


Figure 36(f): Response to Wiener attacked (γ = 1.2) color image.

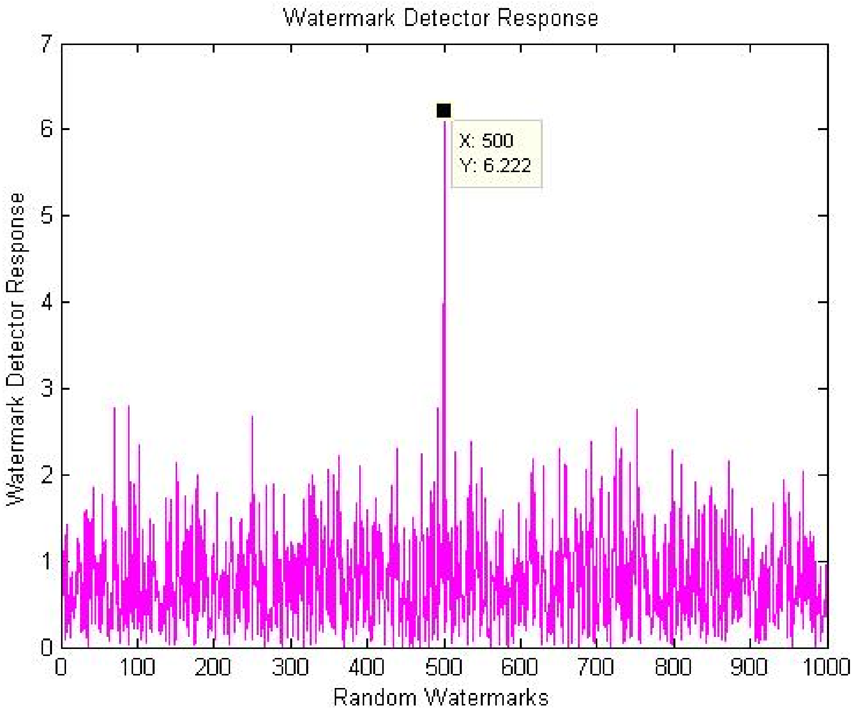


Figure 36(g): Response to Wiener attacked (γ = 1.4) color image.

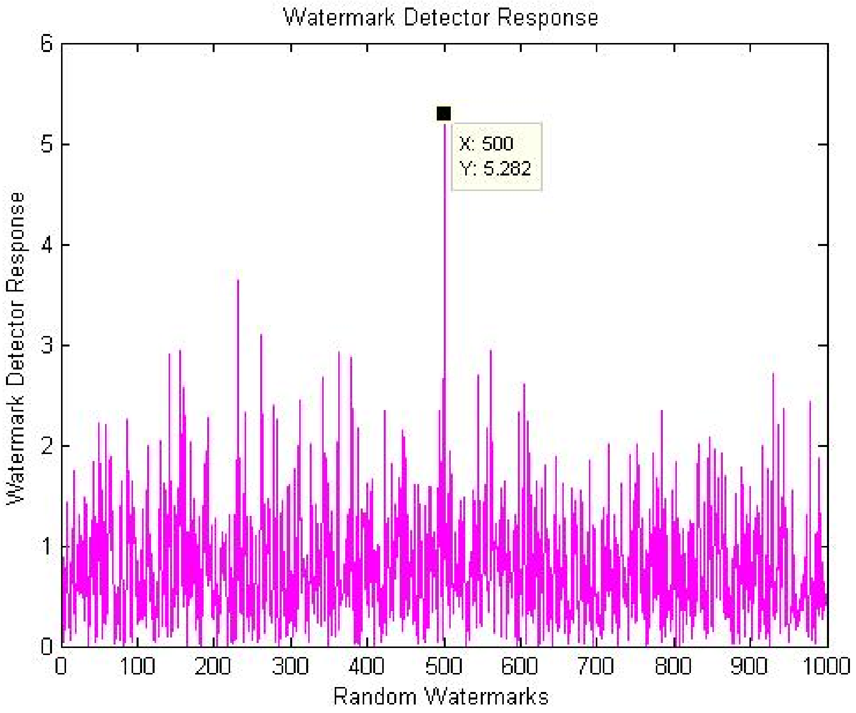


Figure 36(h): Response to Wiener attacked (γ = 1.6) color image.

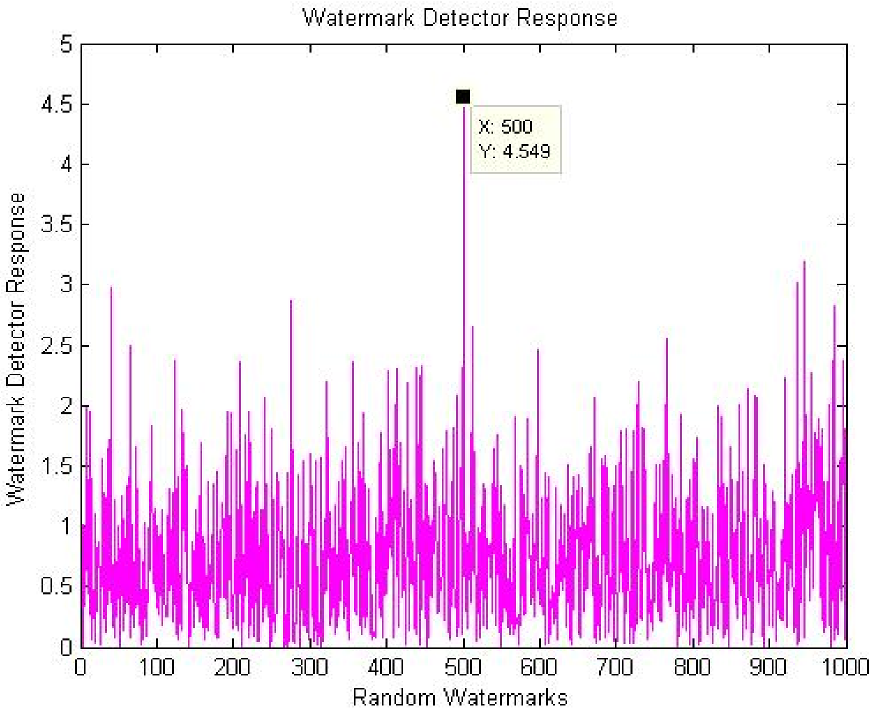


Figure 36(i): Response to Wiener attacked (γ = 1.8) color image.

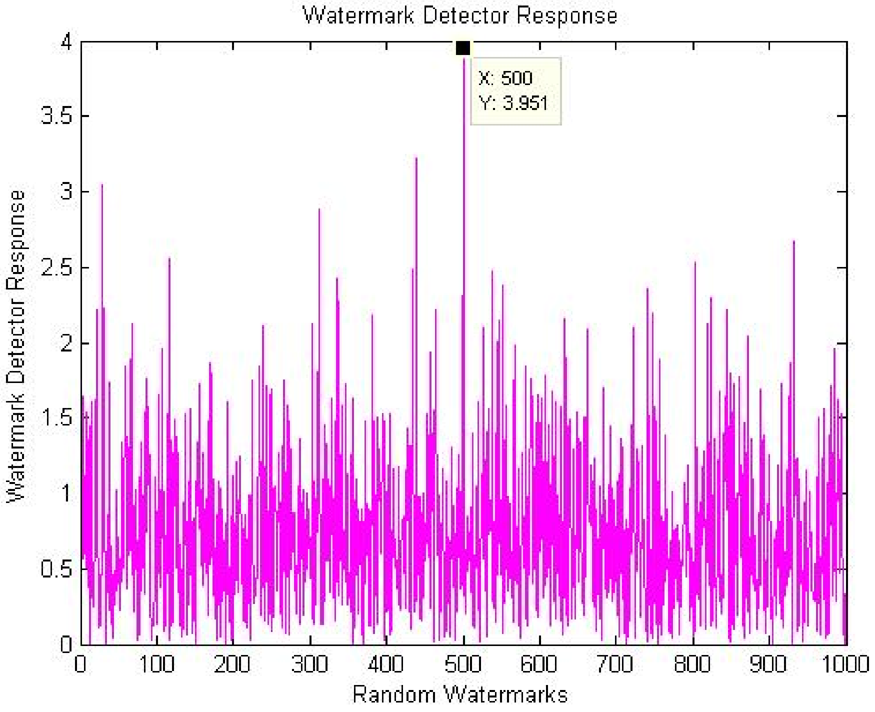


Figure 36(j): Response to Wiener attacked (γ = 2.0) color image.

S.No.	**Wiener Attack**	**SNR (dB)** (Gray-scale image)	**Similarity** (Gray-scale image)	**SNR (dB)** (Color image)	**Similarity** (Color image)
1	Wiener $\gamma = 0.2$	23.7593	27.8437	24.0446	26.0344
2	Wiener $\gamma = 0.4$	18.8807	20.3104	19.1838	19.5519
3	Wiener $\gamma = 0.6$	15.5821	14.3766	15.9265	14.6619
4	Wiener $\gamma = 0.8$	13.1562	10.3430	13.5263	11.3237
5	Wiener $\gamma = 1.0$	11.2461	7.5379	11.6363	9.0452
6	Wiener $\gamma = 1.2$	9.6791	5.6285	10.0788	7.4104
7	Wiener $\gamma = 1.4$	8.3481	4.1702	8.7573	6.2216
8	Wiener $\gamma = 1.6$	7.1937	3.0731	7.6091	5.2821
9	Wiener $\gamma = 1.8$	6.1744	2.2172	6.5951	4.5494
10	Wiener $\gamma = 2.0$	5.2631	1.5026	5.6869	3.9514

Table 2: Experimental results obtained with 256x256 gray-scale and color images, embedding a random watermark, applying Cox's Spread-Spectrum Watermarking Algorithm and subjected to Wiener attacks.

The observations in above (Table 2) indicate that small energy of the random watermark signal, which is embedded into a cover image signal by applying Cox's Spread-Spectrum Watermarking Algorithm can be removed by Wiener attacks. Therefore, we should explore some methods to generate energy-efficient watermarks, to withstand such attacks.

4.9: Necessity for 'Energy-Efficient' Watermarking

After evaluating common attacks (as shown in Table 1) on the watermarked images, I tried to attempt so called '**WIENER ATTACKS** explored by Jonathan Su and B. Girod [2]. The results in above Table -1 indicate that the Wiener attacks remove the MMSE estimates, including the random watermark to much extent, which was inserted by implementing Cox's spread-spectrum watermarking algorithm. This framework leads to develop an idea of **energy-efficient watermarking**, and it enables us to link watermark detectability to signal quality. The latter property produces a meaningful robustness criterion.

Chapter 5: ENERGY– EFFICIENT WATERMARKING

5.1: WATERMARKING MODEL

5.1.1: Watermark Embedding (A Theoretical Approach)

In the reference paper [2] Su and Girod have proposed a generalized approach, considering the cover image and the watermark both as random processes as below:

The watermarked signal $y[n]$ is simply $y[n] = x[n] + w[n]$, where $x[n]$ and $w[n]$ are realizations of the respective random processes $\mathbf{x}[n]$ and $\mathbf{w}[n]$. In the context of random processes:

$$\mathbf{y}[n] = \mathbf{x}[n] + \mathbf{w}[n] \tag{5.1}$$

Since and are independent

$$\Phi_{yy}(\omega) = \Phi_{xx}(\omega) + \Phi_{ww}(\omega), \quad \text{and} \quad \Phi_{wy}(\omega) = \Phi_{ww}(\omega) \tag{5.2}$$

where $\Phi_{wy}(\omega)$ is the cross-power spectrum of $\mathbf{w}[n]$ and $\mathbf{y}[n]$.

Su and Girod remark that many current watermarking methods are based on spread-spectrum communications [13]. The seminal work on digital image fingerprinting by Cox et al. in [1] popularized the use of direct-sequence spread-spectrum for watermarking. The model in equation (5.1) encompasses spread-spectrum watermarking.

Therefore, in order to apply energy-efficient watermarking in spread-spectrum domain; I have used Cox's watermarking algorithm [1] in my experiments.

5.1.2: Distortion Measure

To quantify signal quality, we measure the distortion between a signal $\hat{x}[n]$ and the original signal $x[n]$ via the sample mean-squared error (sample MSE):

$$D(\hat{x}, x) = \frac{1}{N} \sum_{n \in \mathcal{N}} (\hat{x}[n] - x[n])^2$$

In the context of random processes $\hat{\mathbf{x}}[n]$ and $\mathbf{x}[n]$, the sample MSE is replaced by an expectation, and the distortion is the (ensemble) MSE

$$D(\hat{\mathbf{x}}, \mathbf{x}) = \mathrm{E}\left[(\hat{\mathbf{x}}[n] - \mathbf{x}[n])^2\right] \tag{5.3}$$

Note that $D(\hat{x}, x)$ is a sample average, while the distortion $D(\hat{\mathbf{x}}, \mathbf{x})$ is an ensemble average. We also express signal quality as fidelity via the original-to-noise ratio (**ONR**), given by $\mathrm{ONR}(\hat{\mathbf{x}}, \mathbf{x}) = 10\log_{10}\sigma_x^2/D(\hat{\mathbf{x}}, \mathbf{x})$ dB. For the watermarked signal $\mathbf{y}[n]$, the embedding distortion is $D(\mathbf{y}, \mathbf{x}) = \sigma_w^2$. The watermark signal should be imperceptible, so we define the watermark-to-original ratio (**WOR**) by

$\mathrm{WOR} = 10\log_{10}(\sigma_w^2/\sigma_x^2)$ = **– ONR(y,x)** dB.

As a rule of thumb for image watermarking, **WORs** below 20 dB are required to keep the watermark imperceptible. For an attacked signal $\hat{\mathbf{y}}[n]$, the attack distortion is$D(\hat{\mathbf{y}}, \mathbf{x})$.

5.1.3: Watermark Detection

Given a received signal $\hat{y}[n]$, the watermark detector makes a (possibly incorrect) decision about the presence or absence of $w[n]$. We assume that the detector is synchronized with the embedded watermark. A popular detection method is correlation detection, in which the detector computes the sample correlation statistic

$$s = \frac{1}{N}\sum_{n\in\mathcal{N}} \hat{y}[n]w[n] \tag{5.4}$$

and then compares to a threshold T to decide whether$w[n]$ is present in $\hat{y}[n]$ ($s > T$) or not ($s \leq T$). A larger value of corresponds to increasing confidence that is indeed present in $\hat{y}[n]$, and typically lies between 0 and σ_w^2. An important assumption in this paper is that the detector is fixed.

In the random-signal context, during detection, the watermark signal is a particular realization$w[n]$ of $\mathbf{w}[n]$ and is completely known to the detector. Hence, when treating the correlation statistic as a random variable, we must condition on $w[n]$. Then, r the expected value of the correlation statistic, is

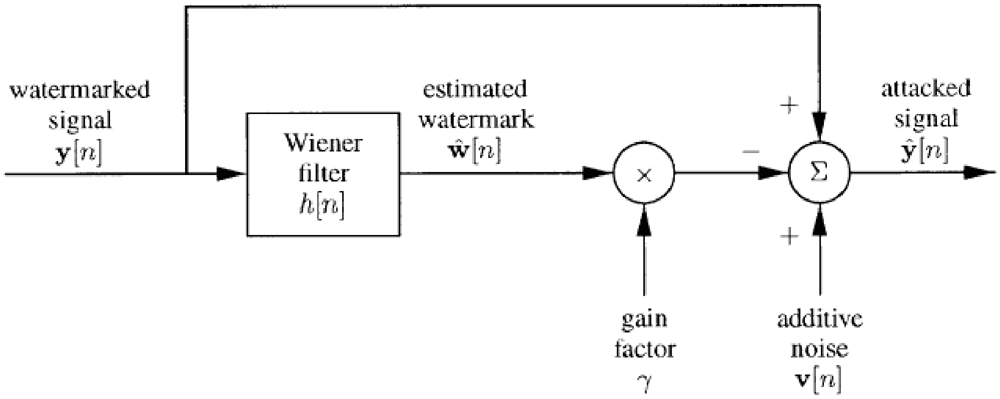
$$r = \mathrm{E}\left[\mathrm{E}\left[\mathbf{s}|\mathbf{w}[n]\right]\right] = \mathrm{E}\left[\mathbf{s}\right]. \tag{5.5}$$

Since usually $\boldsymbol{0} < \boldsymbol{T} < \sigma_w^2$, we often normalize r by σ_w^2 to describe the relative amount of watermark power that reaches the receiver.

Please note that in my experiments, I have implemented watermark embedding in accordance to the Cox's algorithm [1]; therefore, I have used the same procedure for detection and similarity measure; for consistency. So, in this case r is analogous to ***sim(X*, X)*** as explained in equation (3.4).

5.2: WIENER ATTACK

The attacker's goal is to minimize the attack distortion $D(\hat{\mathbf{y}}, \mathbf{x})$, such that $r = r_0$, (i.e. in our experiment analogous to value of **sim(X*, X) ≥ 6**). To impose some structure on the problem, we assume that the attack consists of LSI filtering and additive noise. As indicated in the block diagram of Wiener attack shown in figure 37 below:



watermarked signal $\mathbf{y}[n]$
Wiener filter $h[n]$
estimated watermark $\hat{\mathbf{w}}[n]$
×
−
+
Σ
+
attacked signal $\hat{\mathbf{y}}[n]$
gain factor γ
additive noise $\mathbf{v}[n]$

Figure 37: Block diagram of Wiener Attack.

Let $g[n]$ and $G(\omega)$ denote the filter's impulse response and transfer function, respectively, and $\mathbf{v}[n]$ denote the noise, which has power spectrum $\Phi_{vv}(\omega)$ and is independent of $\mathbf{x}[n]$ and $\mathbf{w}[n]$. Then the attacked signal is

$$\hat{\mathbf{y}}[n] = g[n] * \mathbf{y}[n] + \mathbf{v}[n] = g[n] * (\mathbf{x}[n] + \mathbf{w}[n]) + \mathbf{v}[n]. \qquad (5.6)$$

We formally state the attacker's problem as: Given $\Phi_{xx}(\omega)$, $\Phi_{ww}(\omega)$, and r_0, select $G(\omega)$, $\Phi_{vv}(\omega)$ to minimize $D(\hat{\mathbf{y}}, \mathbf{x})$ such that $r = r_0$. The solution is given by the following theorem, which is proved in the Appendix B

Theorem 5.1 ***(Wiener Attack)*** Let $\Phi_{xx}(\omega)$, $\Phi_{ww}(\omega)$, and r_0, be given. Under the constraint $r = r_0$, $D(\hat{\mathbf{y}}, \mathbf{x})$ is minimized if and only if

$$G(\omega) = 1 - \gamma H(\omega), \qquad \Phi_{vv}(\omega) = 0, \tag{5.7}$$

where γ is a real, scalar gain factor, and

$$H(\omega) = \frac{\Phi_{ww}(\omega)}{\Phi_{xx}(\omega) + \Phi_{ww}(\omega)}, \tag{5.8}$$

With $G(\omega)$, $H(\omega)$, and $\Phi_{vv}(\omega)$ so defined, for any γ,

$$r = \sigma_w^2 - \gamma\sigma_{\hat{w}}^2 \tag{5.9}$$

$$D(\hat{\mathbf{y}}, \mathbf{x}) = \sigma_w^2 - \gamma(2-\gamma)\sigma_{\hat{w}}^2, \tag{5.10}$$

$$E = \mathrm{E}\left[(\mathbf{w}[n] - \hat{\mathbf{w}}[n])^2\right] = \sigma_w^2 - \sigma_{\hat{w}}^2 \tag{5.11}$$

where,

$$\sigma_{\hat{w}}^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\Phi_{ww}^2(\omega)}{\Phi_{ww}(\omega) + \Phi_{xx}(\omega)}\, d\omega. \tag{5.12}$$

Hence, to achieve $r = r_0$:

$$\gamma = \left(\sigma_w^2 - r_0\right)/\sigma_{\hat{w}}^2. \tag{5.13}$$

The corresponding attack distortion is

$$D(\hat{\mathbf{y}}, \mathbf{x}) = \sigma_w^2 - 2\left(\sigma_w^2 - r_0\right) + \frac{\left(\sigma_w^2 - r_0\right)^2}{\sigma_{\hat{w}}^2}. \tag{5.14}$$

Let $h[n]$ denote the impulse response corresponding to $H(\omega)$, so $g[n] = \delta[n] - \gamma h[n]$. Also let $\hat{\mathbf{w}}[n] = h[n] * \mathbf{y}[n]$. Observe that is the transfer function $H(\omega)$ of the Wiener filter for estimating $\mathbf{w}[n]$ from $\hat{\mathbf{y}}[n]$, so $\hat{\mathbf{w}}[n]$ is the Wiener or linear minimum mean-squared error (LMMSE) estimate of given $\mathbf{w}[n]$. ***E*** in (5.11) is the MSE of the estimate. If $\mathbf{x}[n]$ and $\mathbf{w}[n]$ are further assumed to be jointly Gaussian, then the Wiener filter produces the MMSE estimate among all estimators, including nonlinear estimators. Equation (5.6) becomes

$$\hat{\mathbf{y}}[n] = (\delta[n] - \gamma h[n]) * \mathbf{y}[n] + \mathbf{v}[n] = \mathbf{y}[n] - \gamma \hat{\mathbf{w}}[n] \qquad (5.15)$$

since (5.7) indicates that $\mathbf{v}[n] = 0, \forall\, n$. From (5.15), the attack can be viewed as first computing the Wiener estimate $\hat{\mathbf{w}}[n]$ of the watermark signal $\mathbf{w}[n]$ from $\mathbf{y}[n]$ and then modifying $\mathbf{y}[n]$ by subtracting a weighted version of $\hat{\mathbf{w}}[n]$ and adding noise $\mathbf{v}[n]$. J.K. Su and B. Girod call this as the Wiener attack [2].

5.2.1: Discussion of the Wiener Attack

The theorem-1 indicates that the attack should not introduce any additive noise. Intuitively, the attacker can only affect r through $\hat{\mathbf{w}}[n]$ since $\hat{\mathbf{w}}[n]$ is the MMSE estimate of $\mathbf{w}[n]$. Any other changes to $\mathbf{y}[n]$ are uncorrelated with $\mathbf{w}[n]$ and can thus only increase $D(\hat{\mathbf{y}}, \mathbf{x})$ without reducing r. Technically, an examination of the expressions for r and $D(\hat{\mathbf{y}}, \mathbf{x})$ in Appendix B reveals that setting $\sigma_v^2 > 0$ increases $D(\hat{\mathbf{y}}, \mathbf{x})$ but does not affect r. The noise does not improve the attack, so the attacker should set $\mathbf{v}[n] = 0, \forall\, n$. This somewhat surprising result occurs because the fixed correlation detector does not compensate for the attack; if the receiver is compensated for the attack, noise would be necessary.

For given power spectra $\Phi_{xx}(\omega)$ and $\Phi_{ww}(\omega)$, we can easily compute the relationship between r and $D(\hat{\mathbf{y}}, \mathbf{x})$. We only need to compute $\sigma_{\hat{w}}^2$ in (5.12) (e.g., by numerical integration), and then we can use (5.14) to find $D(\hat{\mathbf{y}}, \mathbf{x})$ for any r_0. We can also compute ***E*** via (5.11).

From (5.9) and (5.10), both and can be parameterized by the gain factor γ. It is now possible to relate watermark detectability, in terms of **sim(X*,X)**, to the attack distortion $D(\hat{\mathbf{y}}, \mathbf{x})$. The attacker varies to trade off r and $D(\hat{\mathbf{y}}, \mathbf{x})$. Two values of γ result in interesting special cases of the Wiener attack.

5.2.1(a): Removal Attack With γ = 1, the Wiener attack is a removal attack. For the attacker, this form has the appealing property that it removes as much of the watermark energy as possible while minimizing the attack distortion. This case is equivalent to Wiener denoising. The result is intuitively clear from (5.15), or it may be derived by taking (5.10) and setting $dD(\hat{\mathbf{y}}, \mathbf{x})/d\gamma = 0$. Also, $r = E$ when γ=1.

5.2.1(b): Anti-correlation Attack The attacker can instead select γ so that r_0 = 0, at the expense of increasing $D(\hat{\mathbf{y}}, \mathbf{x})$. We denote this special value ofγ by γ_0,

$$\gamma_0 = \sigma_w^2/\sigma_{\hat{w}}^2. \tag{5.16}$$

This choice of γ drives r to zero with the minimum corresponding distortion $D(\hat{\mathbf{y}}, \mathbf{x})$. Since usually **0 < *T* <** σ_w^2, the probability that the detector mistakenly decides that $w[n]$ is not present in $\hat{y}[n]$ is at least 0.5. We call this attack an anticorrelation attack; the name emphasizes that the attack forces r to zero, as opposed to disabling detection by some other mechanism (e.g., desynchronization). We do not use the term “decorrelation attack,” which could imply transforming $\mathbf{w}[n]$ or $\mathbf{y}[n]$ into uncorrelated components. The Wiener attack is easier to analyze because of its linearity.

5.2.2: Energy-Efficient Watermarking

We can interpret the normalized MSEE/σ_w^2 as the fraction of watermark energy that resists MMSE estimation. Since energy that can be estimated can also be removed, it is wasted. A watermark that maximizes

E/σ_w^2 wastes the minimum fraction of its energy and is said to be energy-efficient. Since $0 \leq E/\sigma_w^2 \leq 1$, we can also compare E/σ_w^2 for different watermarks. A larger ratio means greater resistance to MMSE estimation. In addition, we now have a well-defined way of evaluating the robustness of a watermark. Given different watermarks $\mathbf{w}_1[n]$, $\mathbf{w}_2[n]$ etc., which are characterized by their respective power spectra $\Phi_{w_1w_1}(\omega)$, $\Phi_{w_2w_2}(\omega)$ etc., the watermark that produces the largest value of $D(\hat{\mathbf{y}}_j, \mathbf{x})$ for a given value of $r = r_0$ is most robust. Similarly, if all watermarks yield the same attack distortion D_0, then the watermark with the greatest value of r_j is most robust. We thus have a meaningful way to compare the robustness of watermarks.

5.3: RESISTING THE WIENER ATTACK

Now let us consider the watermarker's perspective. The watermarker wishes to maximize $D(\hat{\mathbf{y}}, \mathbf{x})$ under the constraints $r = r_0$ and $D(\mathbf{y}, \mathbf{x}) = \sigma_w^2 \leq D_{\text{embed}}$. So that the greatest amount of watermark energy might reach the receiver, the watermarker should choose $\sigma_w^2 = D_{\text{embed}}$. The watermarker cannot alter the original signal's power spectrum $\Phi_{xx}(\omega)$, but the watermarker has the freedom to specify the watermark's power spectrum $\Phi_{ww}(\omega)$. From (14), $D(\hat{\mathbf{y}}, \mathbf{x})$ is maximized when $\sigma_{\hat{w}}^2$ is minimized, and from (5.11), $\sigma_{\hat{w}}^2$ is minimized when ***E*** is maximized. Hence, regardless of r_0, the watermarker should choose $\Phi_{ww}(\omega)$ to maximize ***E*** ; and hence create an energy-efficient watermark under the variance constraint. The solution of this problem leads to the theorem below; the proof appears in [11].

Theorem 5.2 (Power-Spectrum Condition): For the watermarking model (5.1), ***E*** is maximized if and only if

$$\Phi_{ww}(\omega) = \frac{\sigma_w^2}{\sigma_x^2}\,\Phi_{xx}(\omega), \qquad (5.17)$$

and, for any dimensionality, the maximum MSE is

$$E_{\mathrm{PSC}} = \frac{\sigma_x^2 \sigma_w^2}{\sigma_w^2 + \sigma_x^2} = \alpha_{\mathrm{PSC}} \sigma_w^2 \tag{5.18}$$

where, $\alpha_{\mathrm{PSC}} = \sigma_x^2/(\sigma_w^2 + \sigma_x^2)$.

5.3.1: Consequences of the Power-Spectrum Condition

We refer to (5.17) as the power-spectrum condition (PSC). It states that the watermark's power spectrum should be directly proportional the original signal's power spectrum. In this sense, the watermark should look like the original. We say that a watermark that satisfies (5.17) is spectrally matched to the original or PSC-compliant. In this section, we study what happens when (5.17) is satisfied.

The main result is that a spectrally-matched watermark signal is most robust, in the sense that the attacker must introduce the greatest amount of distortion to make $r = r_0$. Important conditions are the assumptions of a fixed correlation detector and the form of the attack (LSI filtering and additive noise).

The Wiener filter transfer function (5.8) reduces to

$$H_{\mathrm{PSC}}(\omega) = H_{\mathrm{PSC}}(\vec{\omega}) = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_x^2} = 1 - \alpha_{\mathrm{PSC}} \tag{5.19}$$

and the corresponding maximum MSE is given in (5.18). Note that the normalized MSE E/σ_w^2 for a PSC-compliant watermark is simply α_{PSC} .

From (5.12), $\sigma_{\hat{w}}^2 = \sigma_w^4/(\sigma_x^2 + \sigma_w^2) = (1 - \alpha_{\mathrm{PSC}})\sigma_w^2$. Then (5.9) and (5.10) give

$$r = (1 - \gamma(1 - \alpha_{\mathrm{PSC}}))\,\sigma_w^2, \tag{5.20}$$

$$D(\hat{\mathbf{y}}, \mathbf{x}) = (1 - \gamma(2 - \gamma)(1 - \alpha_{\mathrm{PSC}}))\,\sigma_w^2 \tag{5.21}$$

These expressions hold regardless of the dimensionality ***M***. Since $\mathbf{w}[n]$ should be imperceptible, we assume $\sigma_x^2 \gg \sigma_w^2$, so $\alpha_{\mathrm{PSC}} \approx 1$.

5.3.2: Special Cases of the Wiener Attack

If the attacker sets γ = 1 for a removal attack, then the expected correlation statistic and attack distortion become $r \ = \quad D(\hat{\mathbf{y}}, \mathbf{x}) = \alpha_{\mathrm{PSC}}\sigma_w^2 \approx \sigma_w^2$. As a result, the variance of the watermark is hardly reduced by the attack, and the distortion of the attacked signal $\hat{\mathbf{y}}[n]$ is a negligible improvement over the watermarked signal $\mathbf{y}[n]$.Indeed, $\mathrm{ONR}(\hat{\mathbf{y}}, \mathbf{x}) = (\sigma_x^2 + \sigma_w^2)/\sigma_w^2 \approx \sigma_x^2/\sigma_w^2 = \mathrm{ONR}(\mathbf{y}, \mathbf{x})$.

Suppose instead that the attacker performs the anti-correlation attack. From (5.16), γ_0 becomes $\gamma_{0,\,\mathrm{PSC}} = (\sigma_w^2 + \sigma_x^2)\sigma_w^2 = 1/(1 - \alpha_{\mathrm{PSC}})$, and (5.10) gives $D(\hat{\mathbf{y}}, \mathbf{x}) \ = \ \sigma_x^2 + \sigma_v^2$. As a result, the attack distortion will be at least as large as the variance of the original signal, and $\mathrm{ONR}(\hat{\mathbf{y}}, \mathbf{x}) \ \leq \ 0$ dB. Such an attacked signal will certainly be useless.

5.4: HOW TO GENERATE ENERGY-EFFICIENT WATERMARKS?

In order to generate energy-efficient (PSC-compliant) watermarks, first, we evaluate the power spectrum $\Phi_{xx}(\omega_1, \omega_2)$ of the $N_1 \times N_2$ original gray-scale image $x[n_1, n_2]$ by using the periodogram, $\mathrm{Per}_{xx}[k_1, k_2] \ = \ |X[k_1, k_2]|^2/(N_1 N_2)$; (for a color image, it should be converted into an intensity image by using **YIQ** conversion, before taking the periodogram); where $X[k_1, k_2]$ is the 2-D FFT of $x[n_1, n_2]$.

Then we produce the energy-efficient watermark by:

$$w[n_1, n_2] = \sqrt{\sigma_w^2/\sigma_x^2}\,\mathrm{IFFT}\left\{\sqrt{\mathrm{Per}_{xx}[k_1, k_2]}\,U[k_1, k_2]\right\} \qquad (5.22)$$

Where, $U[k_1, k_2]$ is the 2-D FFT of the output $u[n_1, n_2]$ of a unit-variance white Gaussian random number generator.

5.5: PRACTICAL LIMITATIONS AND CONSTRAINTS

In this project I have implemented Cox's spread-spectrum algorithm for watermark embedding and detection, since this methodology offers the benefits of imperceptibility and robustness. This algorithm utilizes a one-dimensional random watermark of about 1000 odd samples. However, robustness of such watermarking scheme is challenged by applying Wiener attack; to counter this attack, a PSC-compliant (energy-efficient) watermark has to be applied.

In my experiments conducted on images (both gray-scale and color) of size 256x256 or above; I faced following major constraints:

(a) The PSC-compliant watermark generated using equation (5.22) is a 2-dimensional watermark. However, the Cox's algorithm embeds a one-dimensional random watermark.

(b) The size of the PSC-compatible watermark is the same size as that of the original cover image (65,536 or more samples). With such a large size of PSC-compatible watermark, the detector fails to give reliable and consistent results.

Therefore, the 2-dimensional energy-efficient watermark has to be converted into a continuous watermark string (one-dimensional); in order to maintain compatibility with the algorithm. Further, the size of PSC-compliant watermark was reduced in two ways:

(1) I took a scaled image, suitably reduced (e.g. 1/10) in both directions, maintaining aspect ratio. Then, I generated a 2-dimensional PSC-compliant watermark with respect to this scaled image, and later converted it into a one-dimensional string before embedding into the original (un-scaled) image. Since scaling of an image essentially performs a low-pass filtering and loss of vital image information, the watermark generated by this method may not perfectly follow the power spectrum of the original cover image.

(2) I generated full-size 2-dimensional PSC-compliant watermark from the cover image and then embedded only its first 2000 samples into the full size original cover image.

> Although this may not be a perfect watermark, but this yielded consistent results than the earlier one, so, I recommend this method.

Moreover, Su and Girod [2] have clearly remarked that *taking the full-size transform of an image may not be the best implementation for actual watermarking schemes*. Also, the periodogram produces an unbiased, but not a consistent, estimate of a signal's power spectrum. Nonetheless, these methods are sufficient for illustrating the relationship between theory and practice.

5.6: RESULTS AND ANALYSIS OF ENERGY-EFFICIENT WATERMARKS

The same experiments, as shown in Chapter 4, are performed again using PSC-compliant (energy-efficient) watermark. The original cover images, PSC-compliant watermarked images and attacked images visually appear to be the same. However, these images are different statistically. So, for conservation of space, these images are omitted here and only the generated PSC-compliant watermark and resulting responses of the watermark detector are shown below:

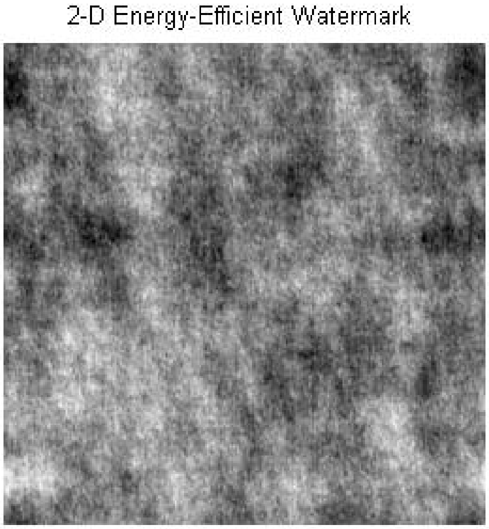


Figure 38: 2-D PSC-compliant watermark for gray-scale image.

Figure 38 and 39 respectively show the actual 2D PSC-compliant watermark and practically embedded watermark for the original gray-scale "Horse" image.

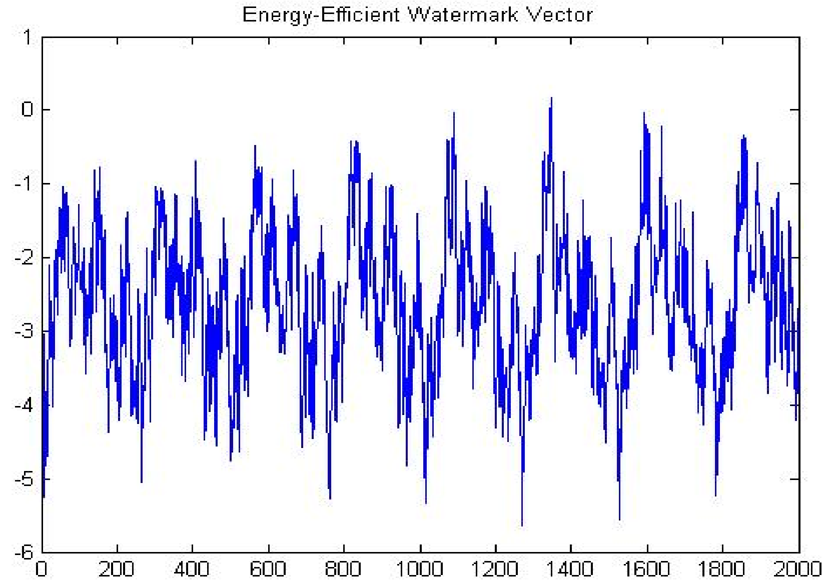


Figure 39: PSC-compliant (1-D) watermark {derived from 2-D watermark shown in fig. 38} practically embedded into the gray-scale cover image.

Also, fig. 40 and fig. 41 show the actual 2D PSC-compliant watermark and practically embedded watermark for the original color ”Bird” image.

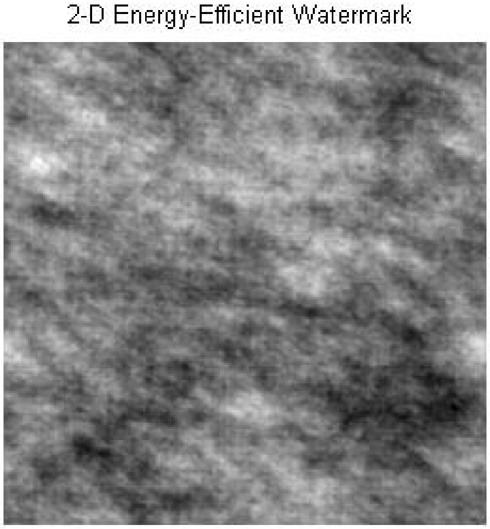


Figure 40: 2-D PSC-compliant watermark for color image.

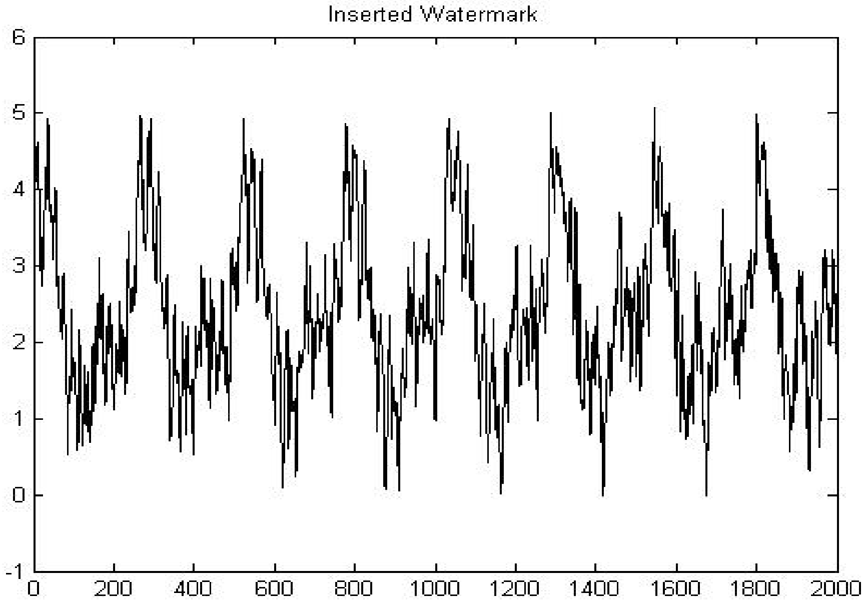


Figure 41: PSC-compliant (1-D) watermark {derived from 2-D watermark shown in fig. 40} practically embedded into the original color image.

Figures 42(a) to 42(l) show the detector's response to PSC-compliant watermark embedded in 256x256 gray-scale "Horse" image. And figures 43(a) to 43(j) show the response to Wiener attacks on it.

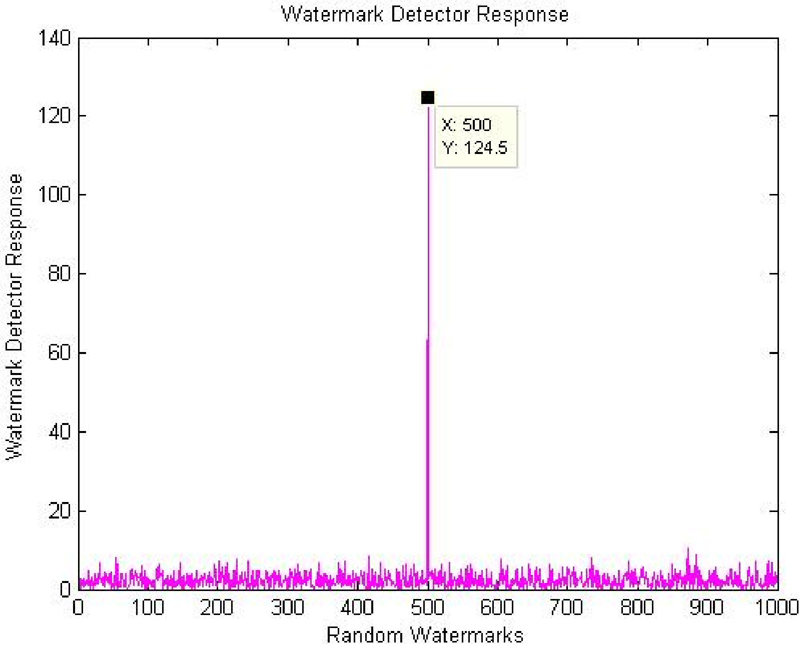


Figure 42(a): Detector response to PSC-compliant watermark embedded in gray-scale image.

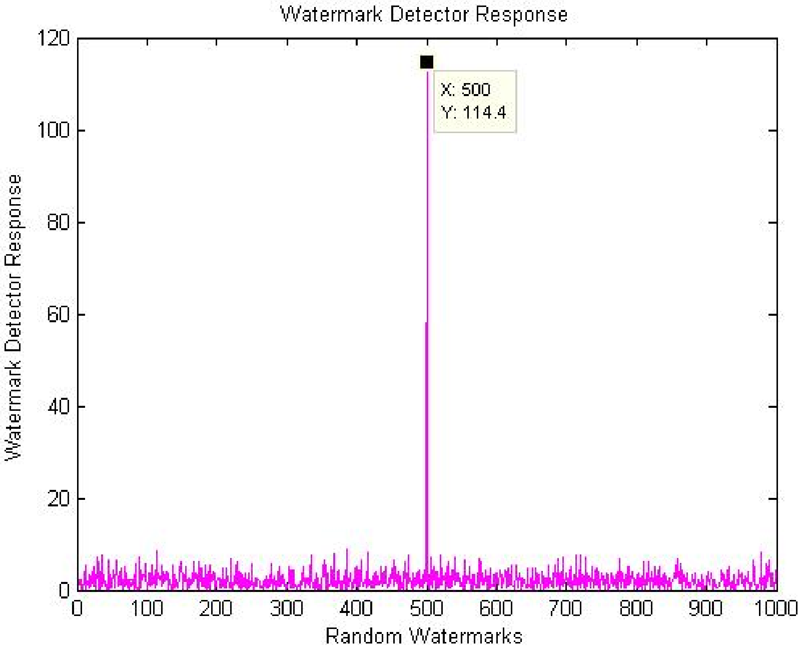


Figure 42(b): Detector response to PSC-compliant watermark in scaled to half, and rescaled gray-scale image.

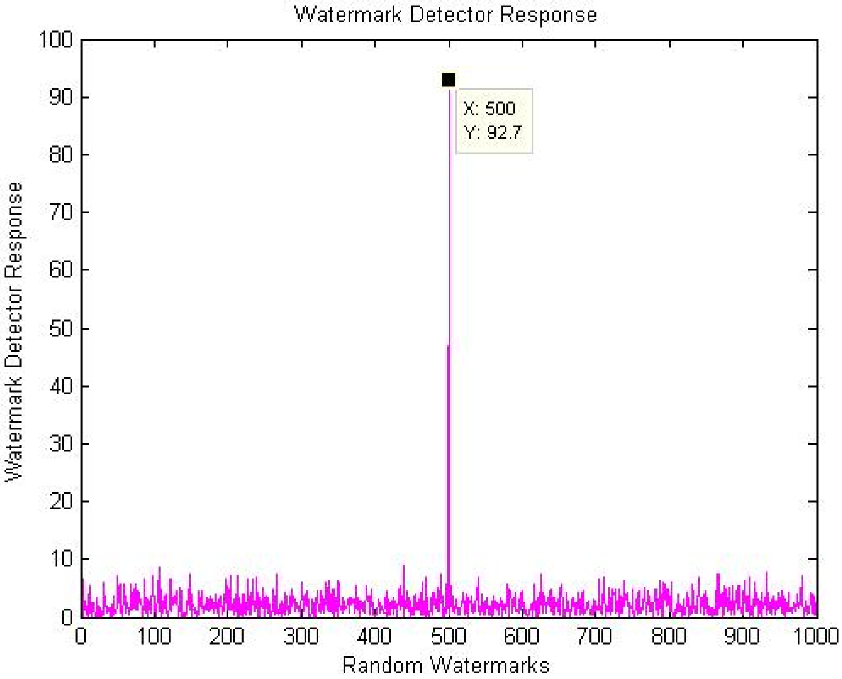


Figure 42(c): Detector response to PSC-compliant watermark in JPEG compressed (5% quality) gray-scale image.

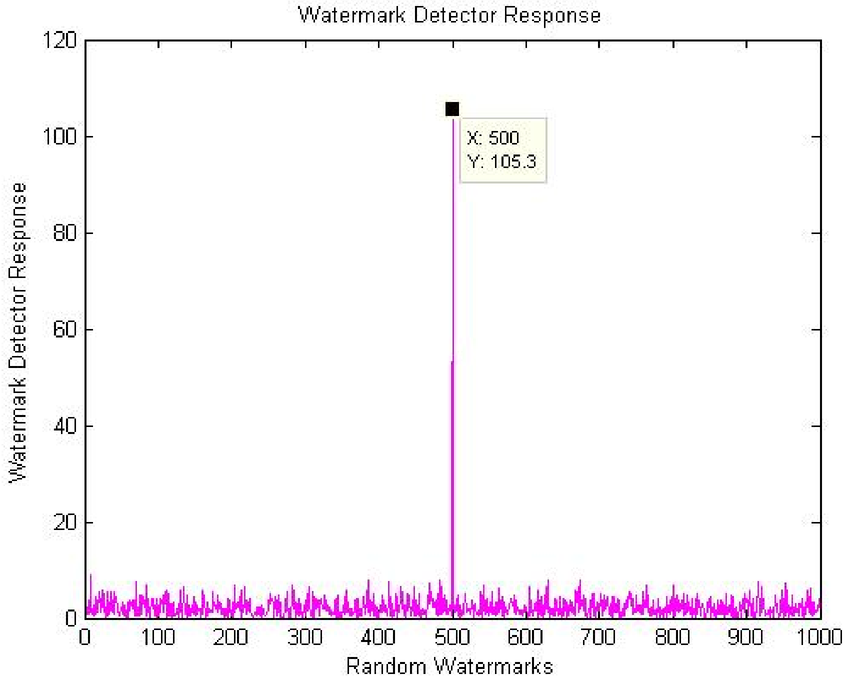


Figure 42(d): Detector response to PSC-compliant watermark in JPEG compressed (10% quality) gray-scale image.

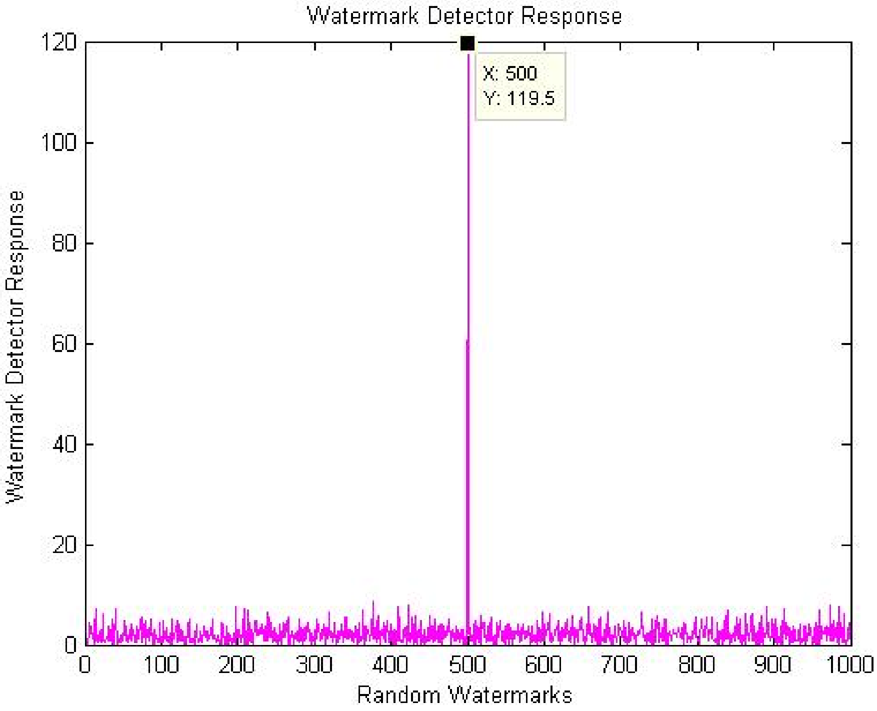


Figure 42(e): Detector response to PSC-compliant watermark in JPEG compressed (25% quality) gray-scale image.

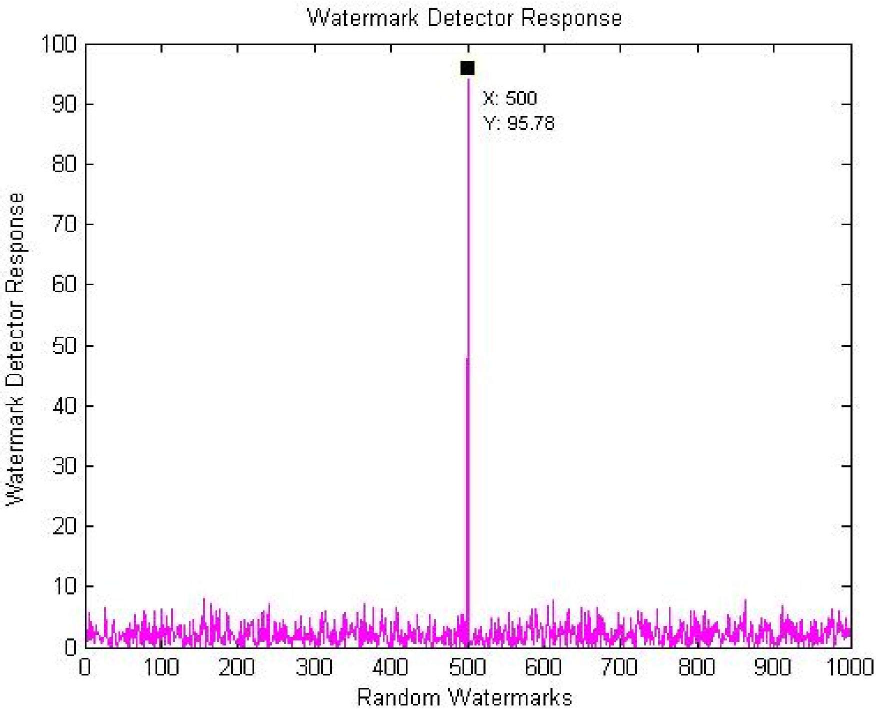


Figure 42(f): Detector response to PSC-compliant watermark in rotation corrected, cropped and scaled gray-scale image.

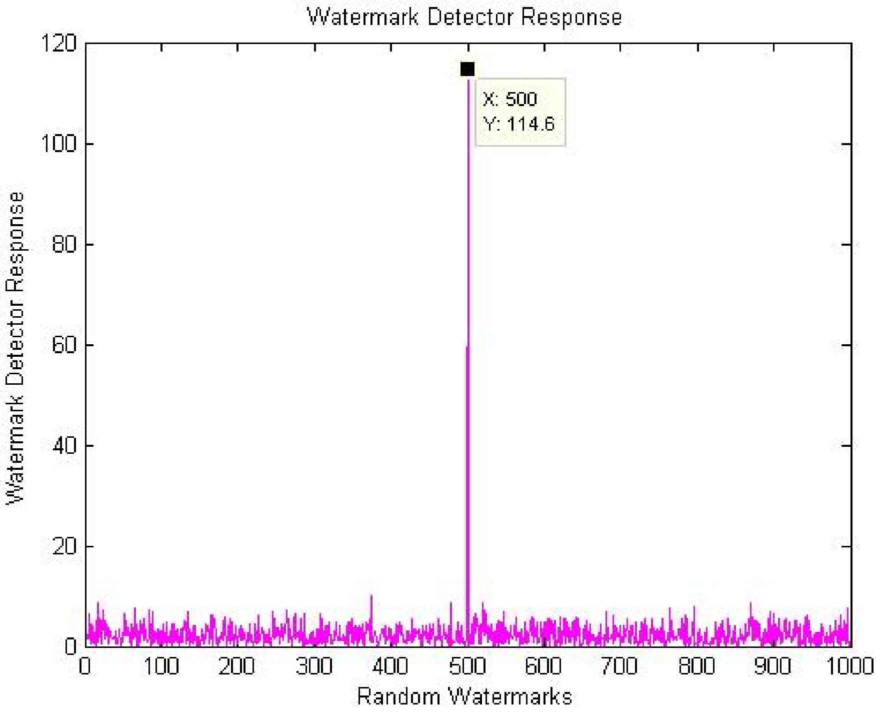


Figure 42(g): Detector response to PSC-compliant watermark in deblurred gray-scale image.

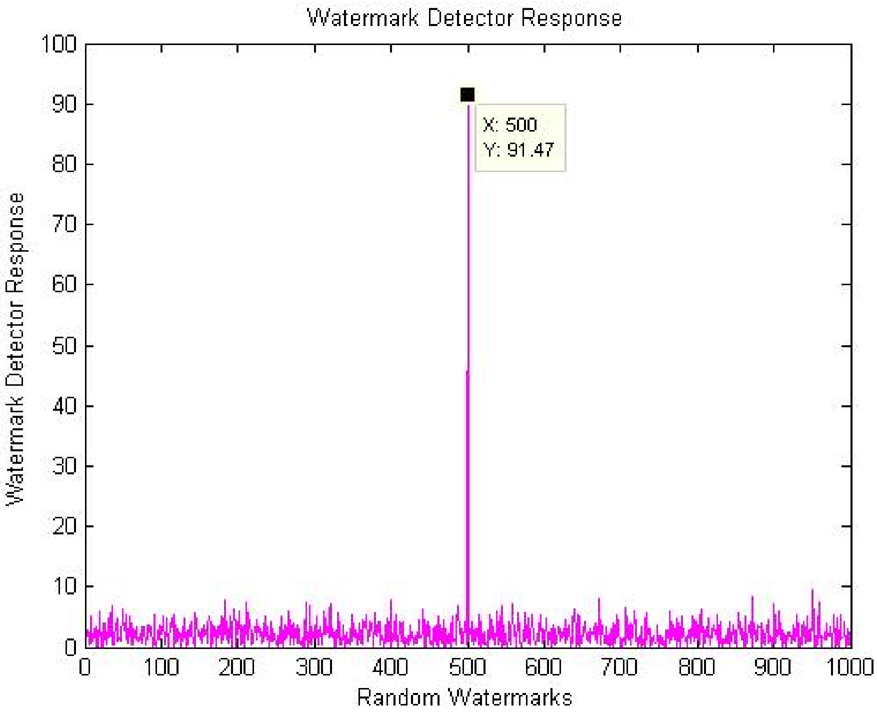


Figure 42(h): Detector response to PSC-compliant watermark in "Salt-n-Pepper Noise" affected gray-scale image.

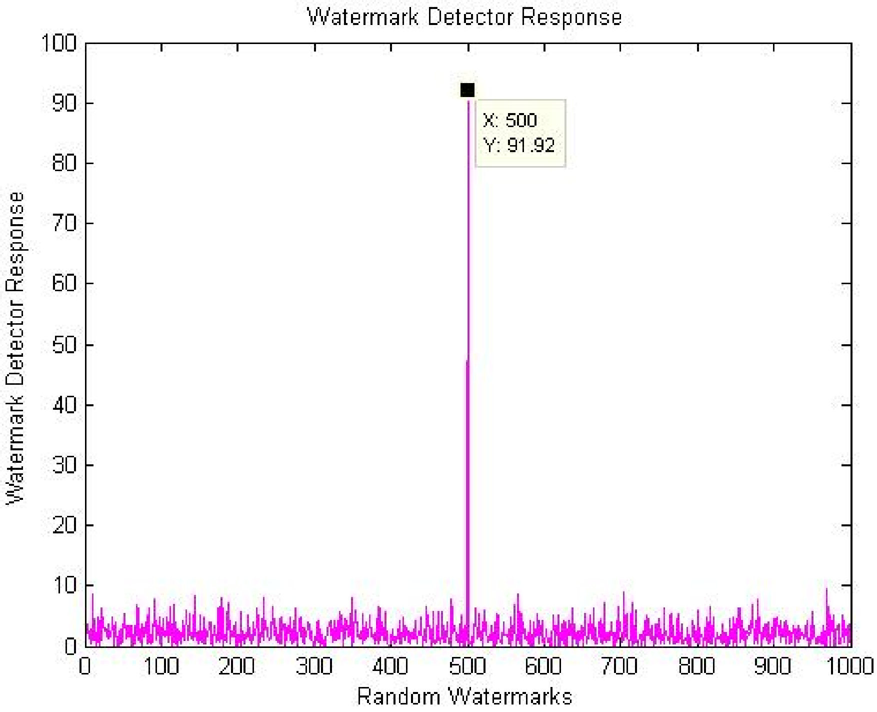


Figure 42(i): Detector response to PSC-compliant watermark in A.W.G. Noise affected gray-scale image.

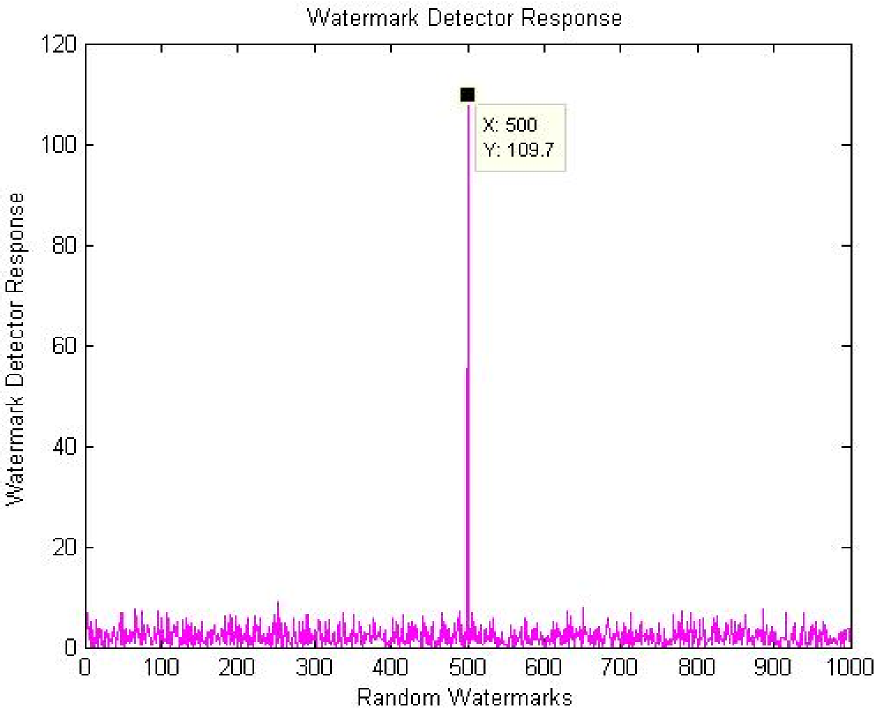


Figure 42(j): Detector response to PSC-compliant watermark in Average Filtered gray-scale image.

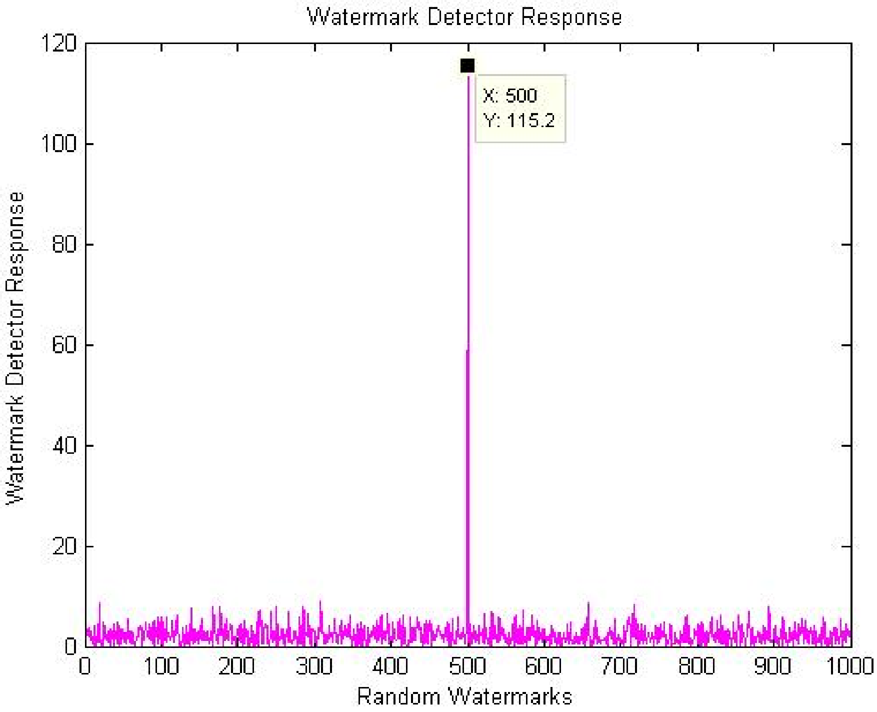


Figure 42(k): Detector response to PSC-compliant watermark in Median Filtered gray-scale image.

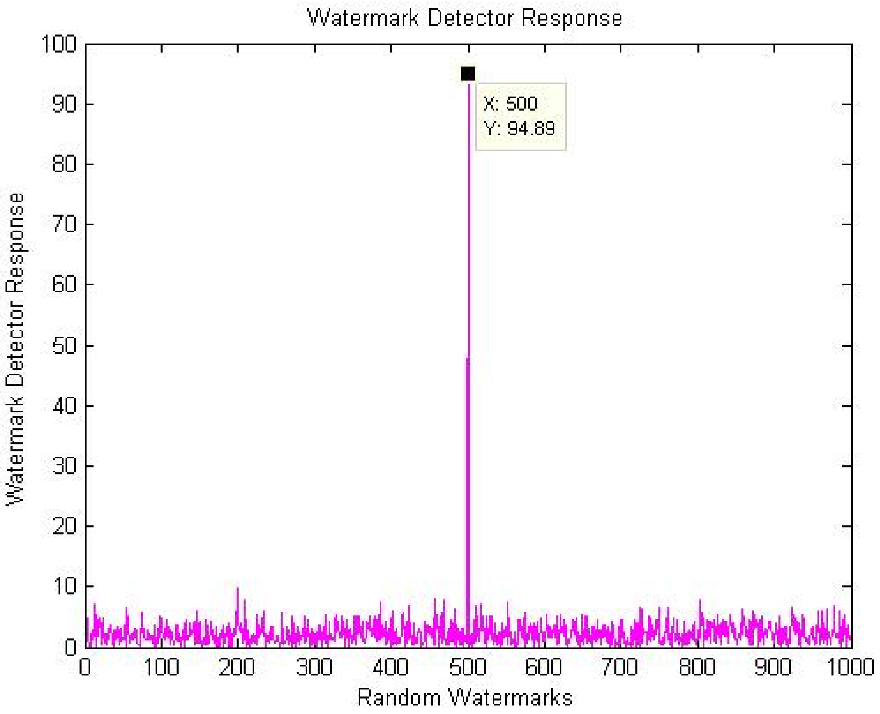


Figure 42(l): Detector response to PSC-compliant watermark in Dithered gray-scale image.

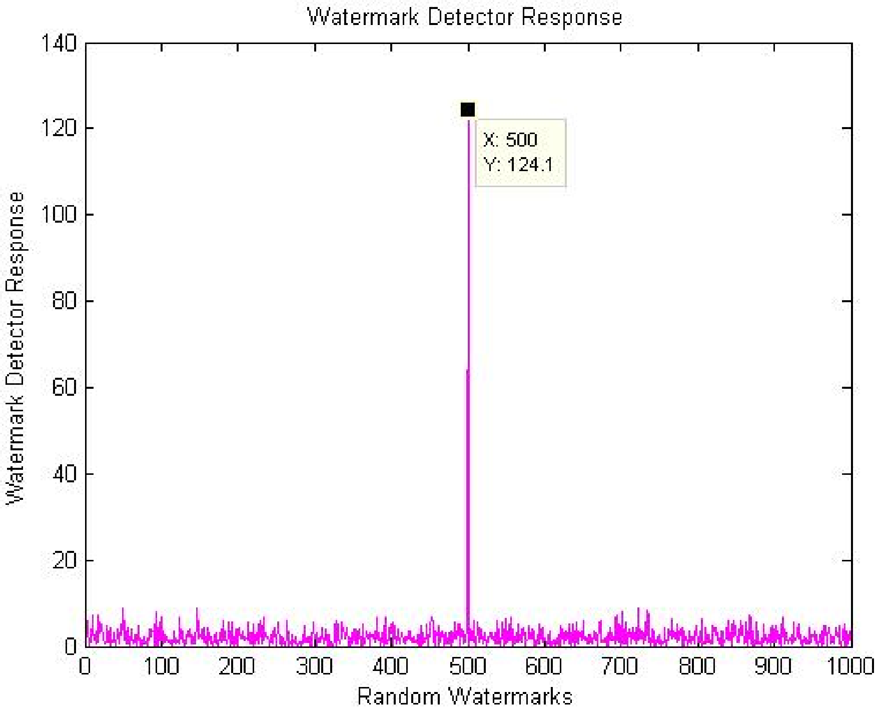


Figure 43(a): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 0.2) gray-scale image.

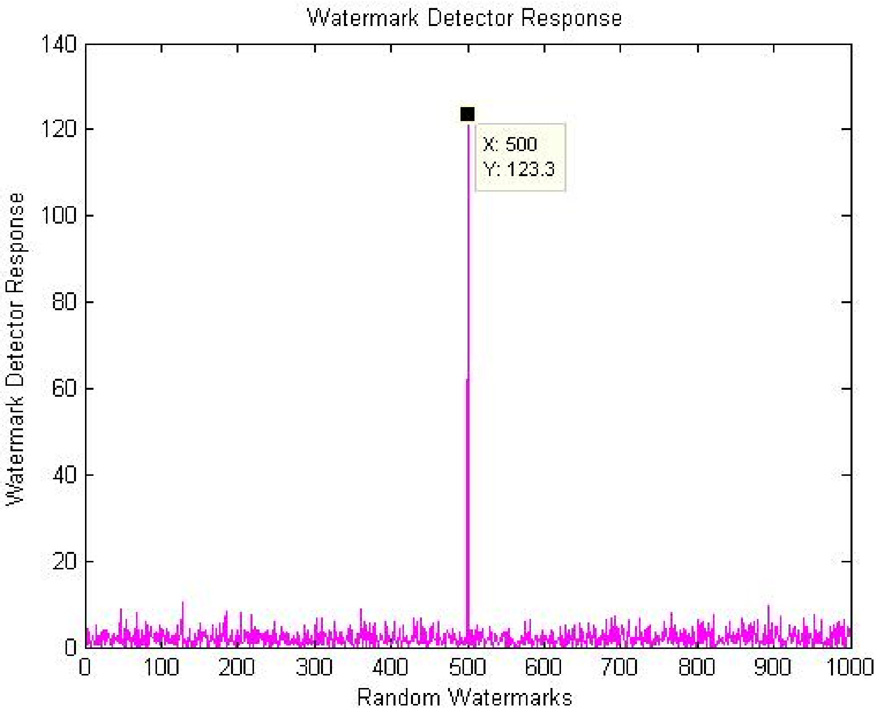


Figure 43(b): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 0.4) gray-scale image.

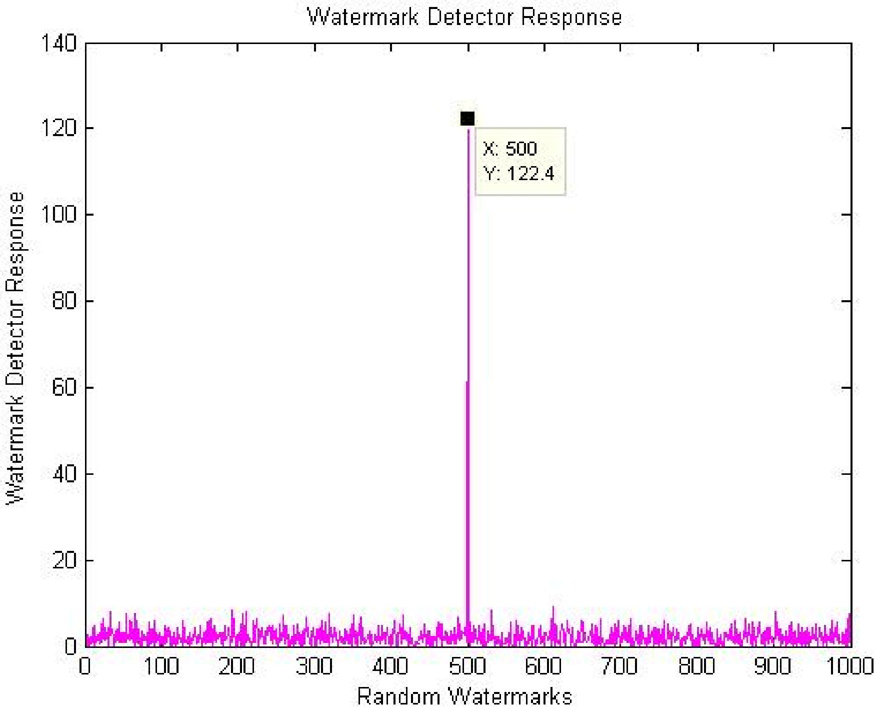


Figure 43(c): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 0.6) gray-scale image.

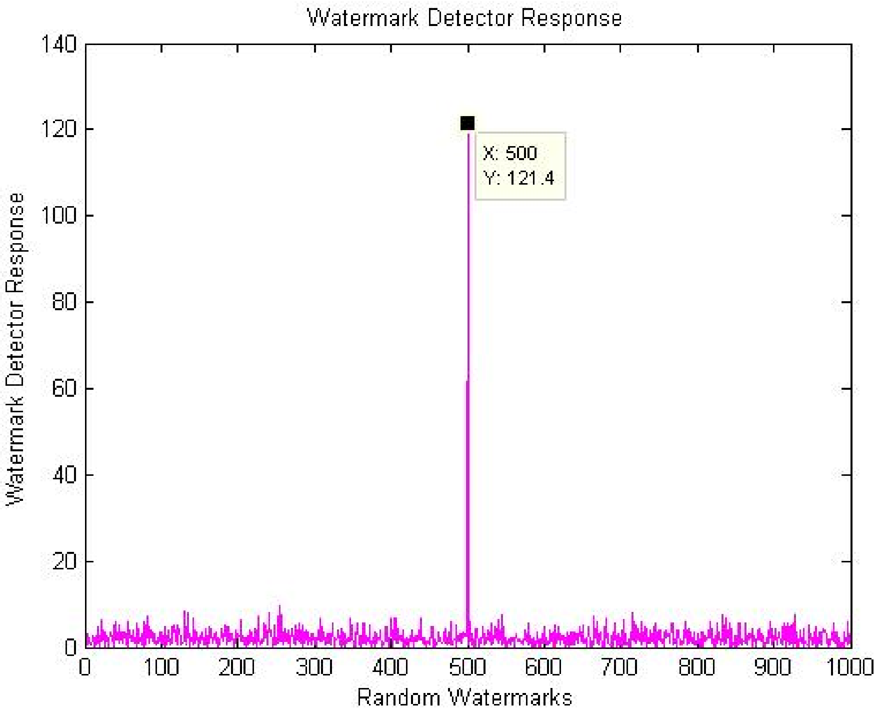


Figure 43(d): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 0.8) gray-scale image.

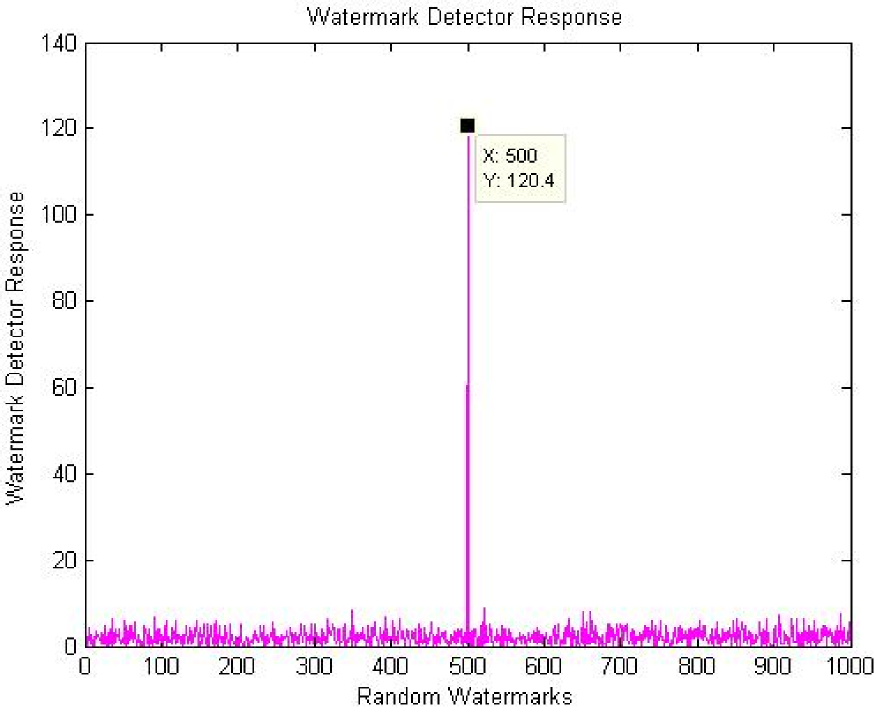


Figure 43(e): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.0) gray-scale image.

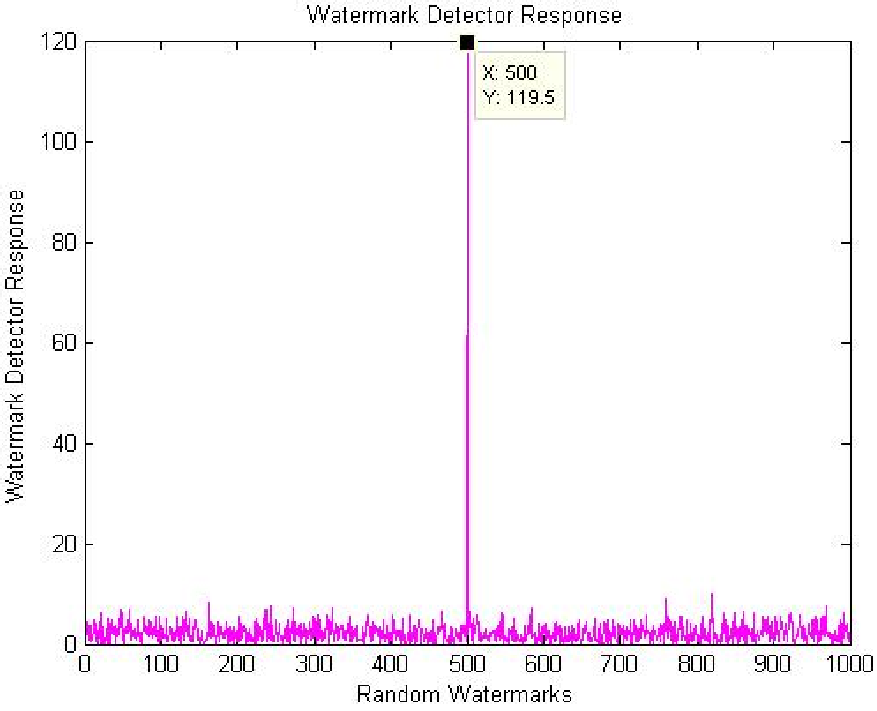


Figure 43(f): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.2) gray-scale image.

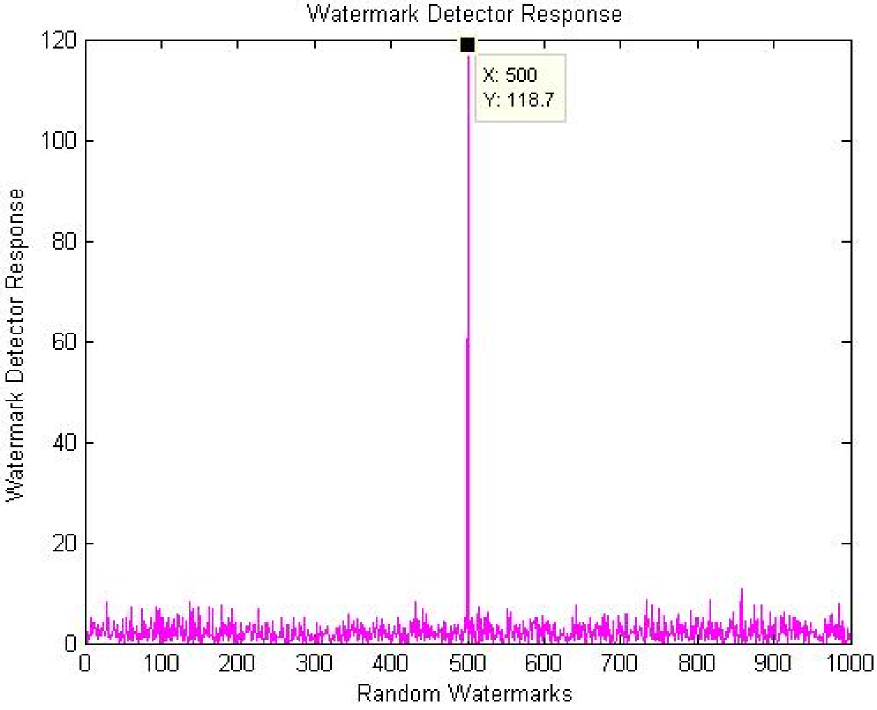


Figure 43(g): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.4) gray-scale image.

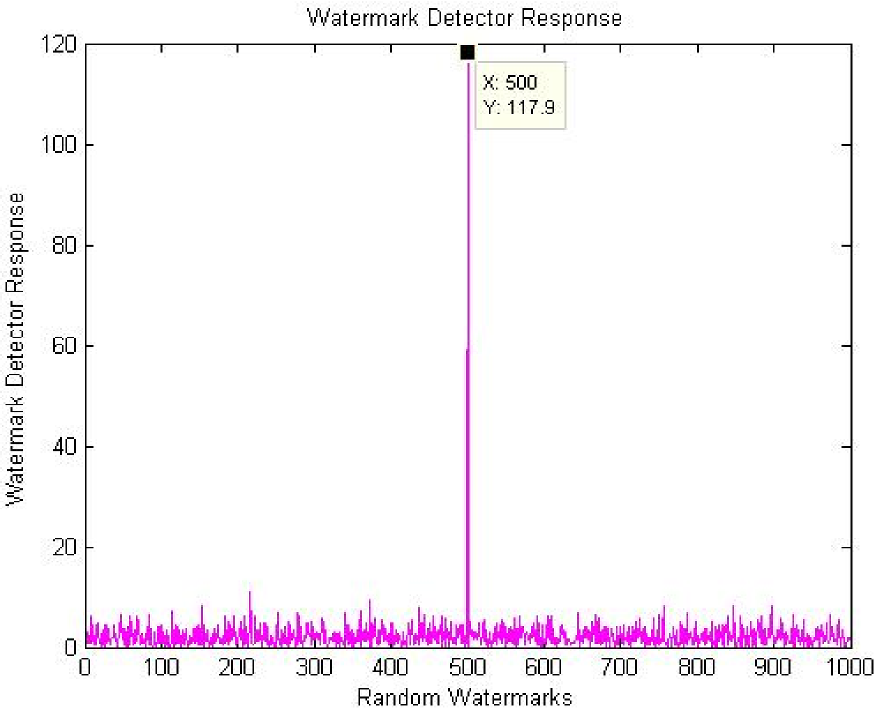


Figure 43(h): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.6) gray-scale image.

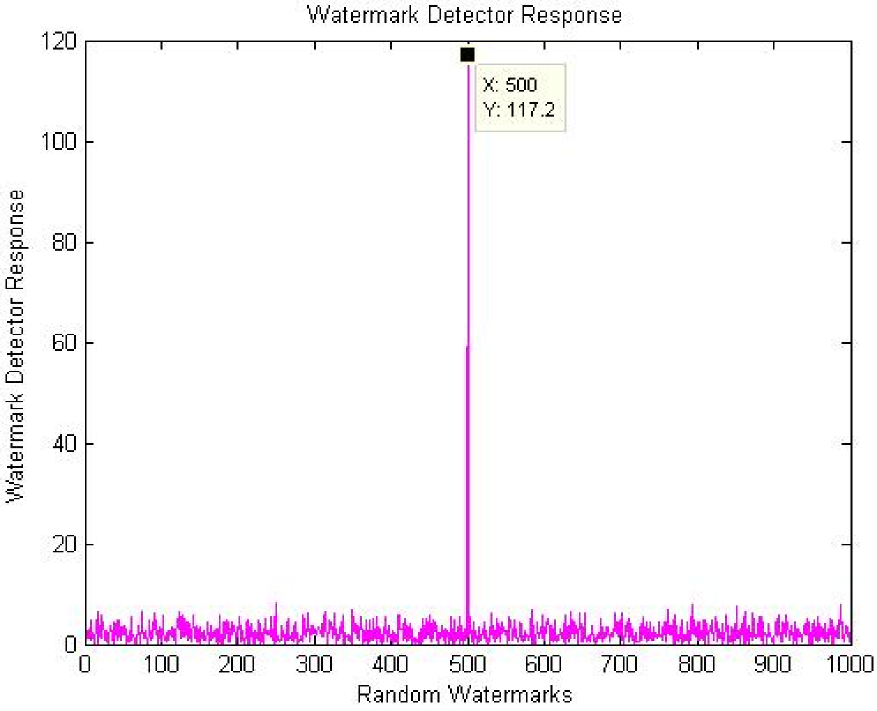


Figure 43(i): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.8) gray-scale image.

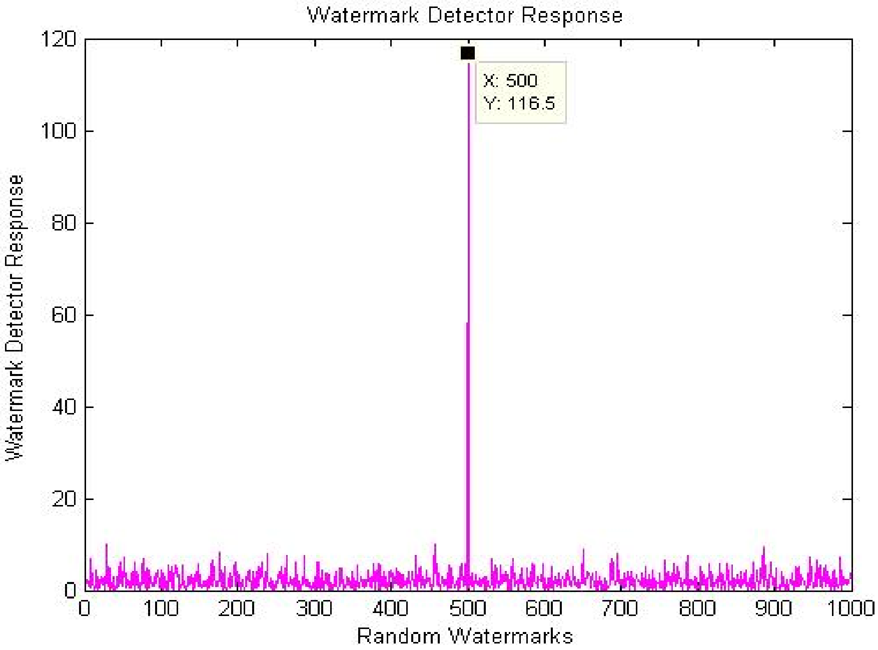


Figure 43(j): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 2.0) gray-scale image.

Figures 44(a) to 44(l) show the detector's response to PSC-compliant watermark embedded in 256x256 color image of "Bird". And figures 45(a) to 45(j) show the response to Wiener attacks on it.

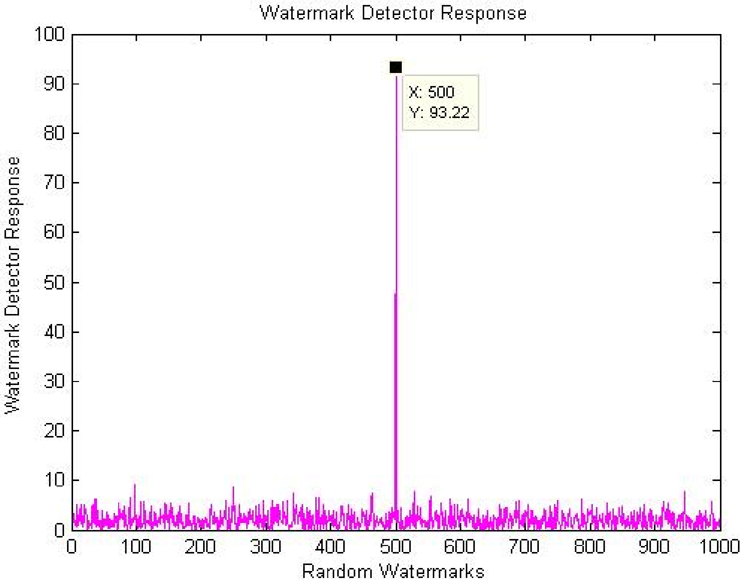


Figure 44(a): Detector response to PSC-compliant watermark embedded in color image.

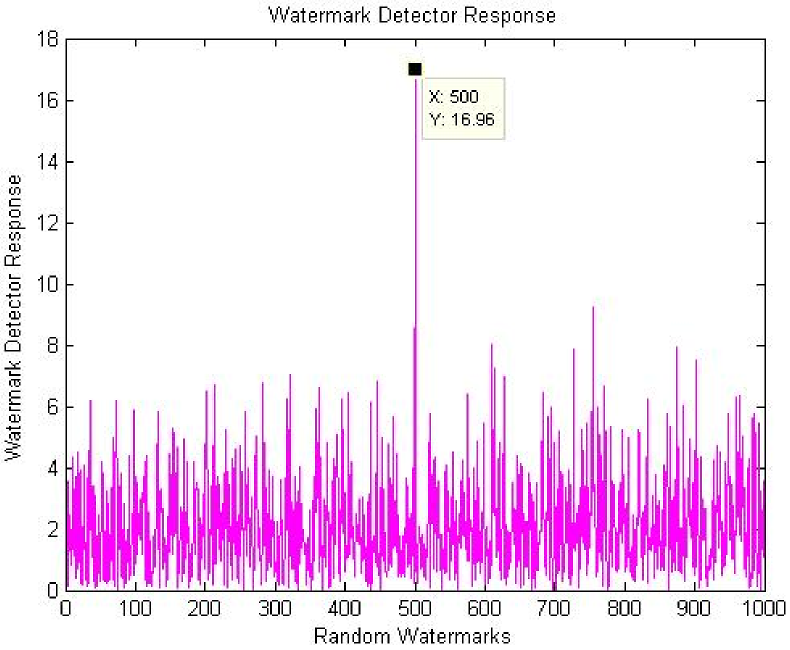


Figure 44(b): Detector response to PSC-compliant watermark in scaled to half and rescaled color image.

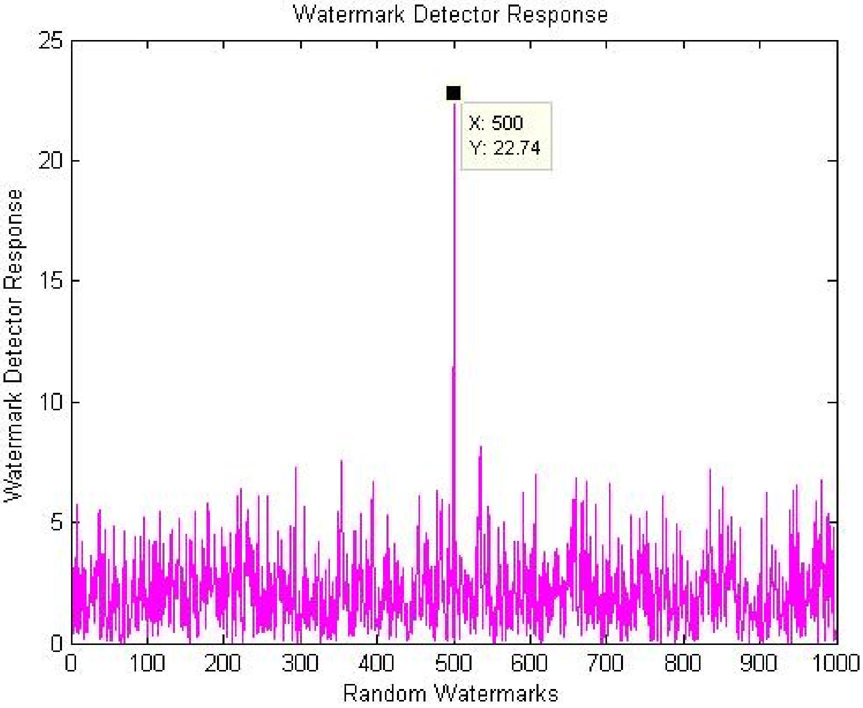


Figure 44(c): Detector response to PSC-compliant watermark in JPEG compressed (5% quality) color image.

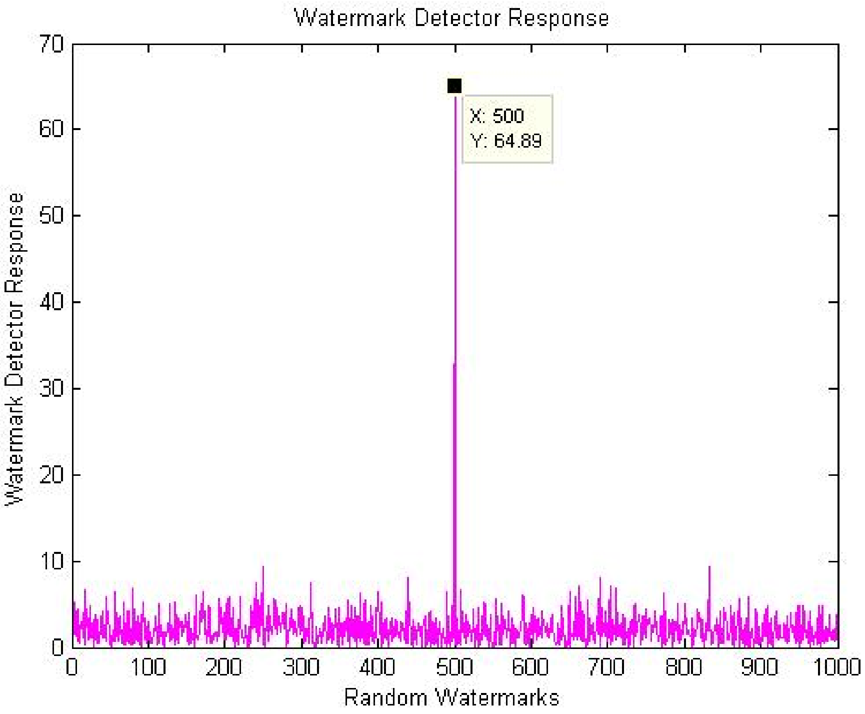


Figure 44(d): Detector response to PSC-compliant watermark in JPEG compressed (10% quality) color image.

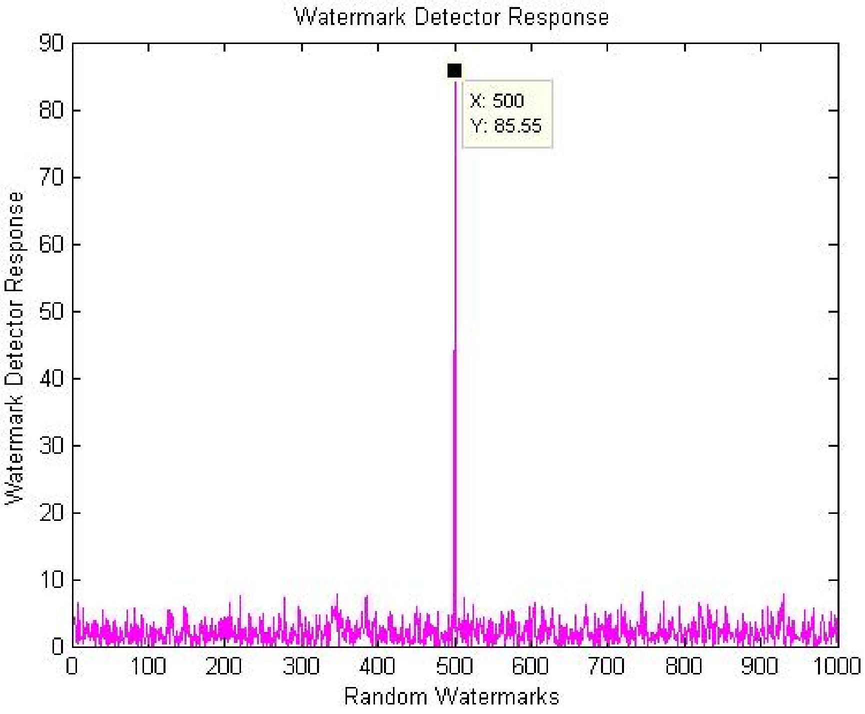


Figure 44(e): Detector response to PSC-compliant watermark in JPEG compressed (25% quality) color image.

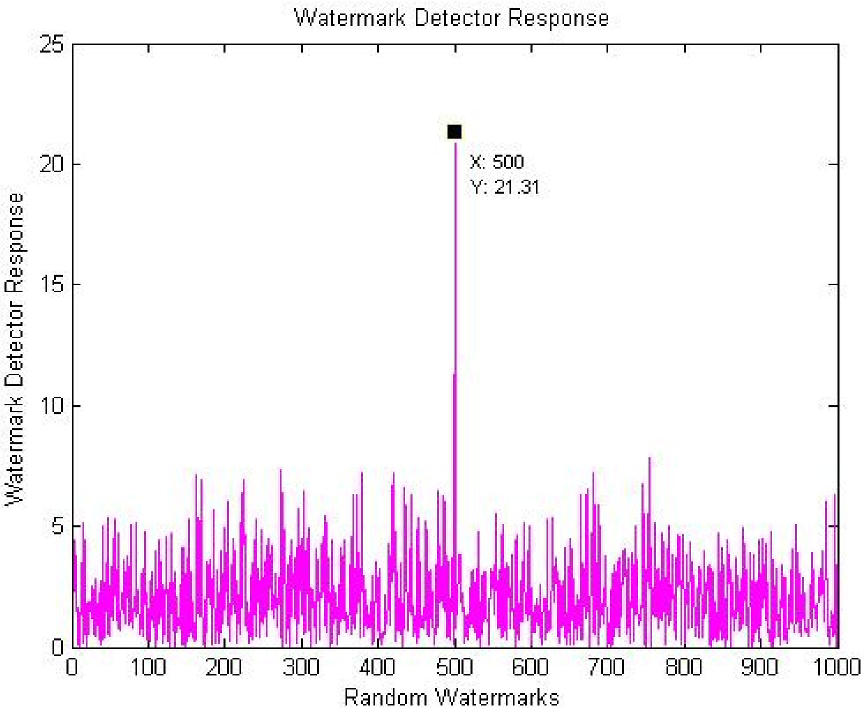


Figure 44(f): Detector response to PSC-compliant watermark in rotation-corrected, cropped and rescaled color image.

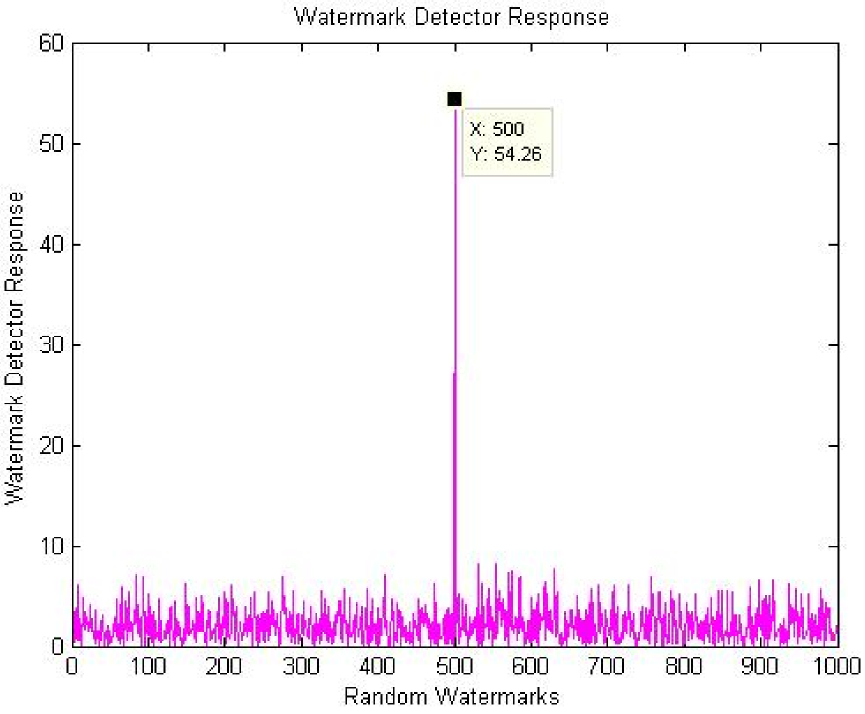


Figure 44(g): Detector response to PSC-compliant watermark in Deblurred color image.

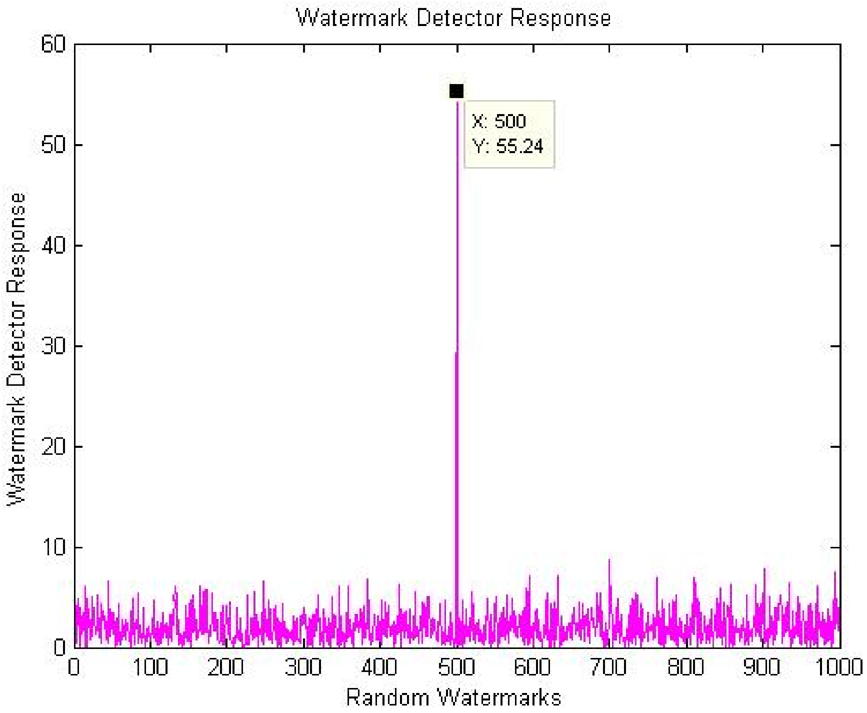


Figure 44(h): Detector response to PSC-compliant watermark in "Salt-n-Pepper Noise" affected color image.

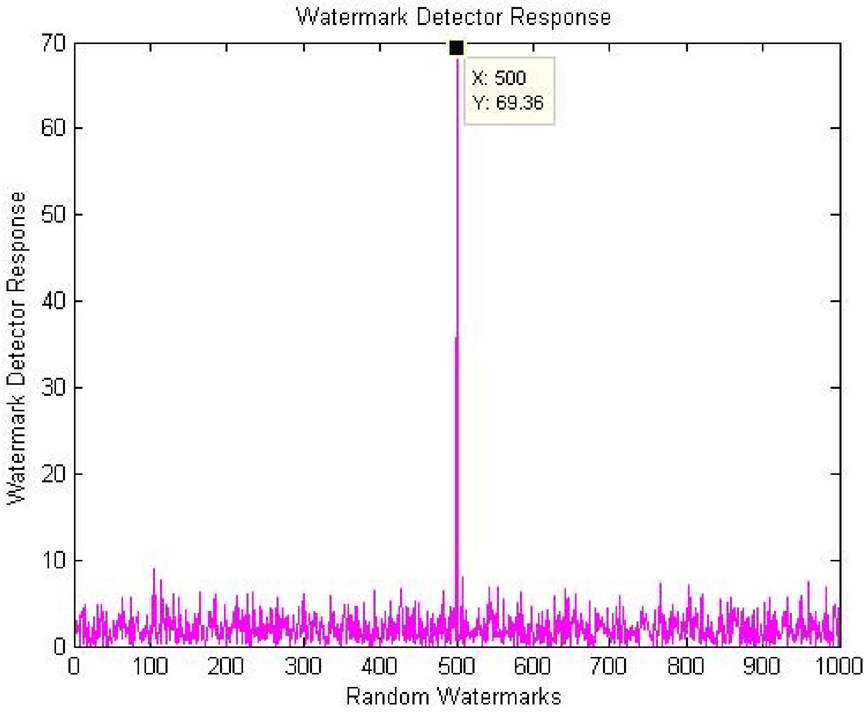


Figure 44(i): Detector response to PSC-compliant watermark in "A.W.G. Noise" affected color image.

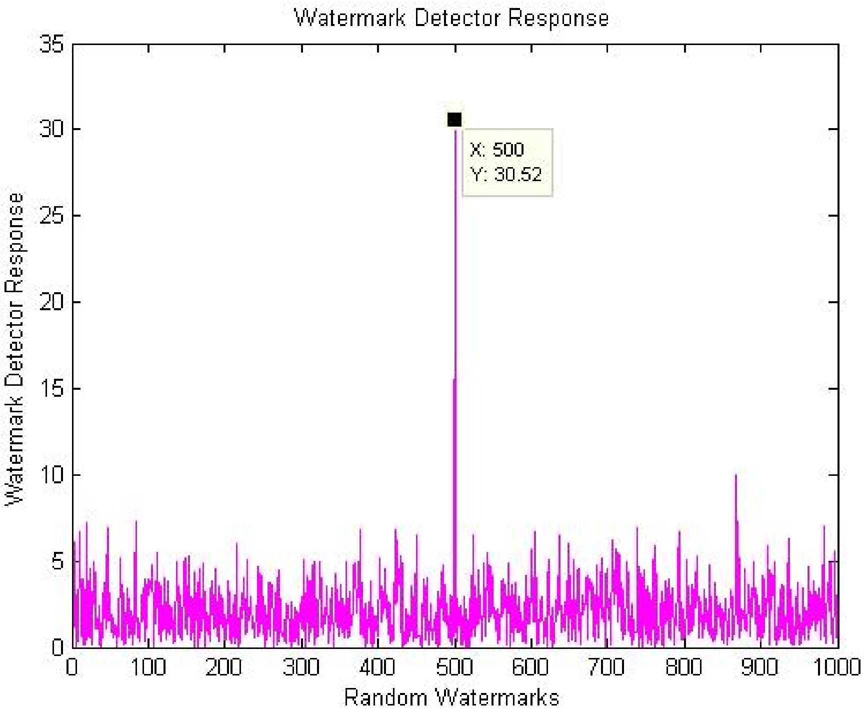


Figure 44(j): Detector response to PSC-compliant watermark in Average Filtered color image.

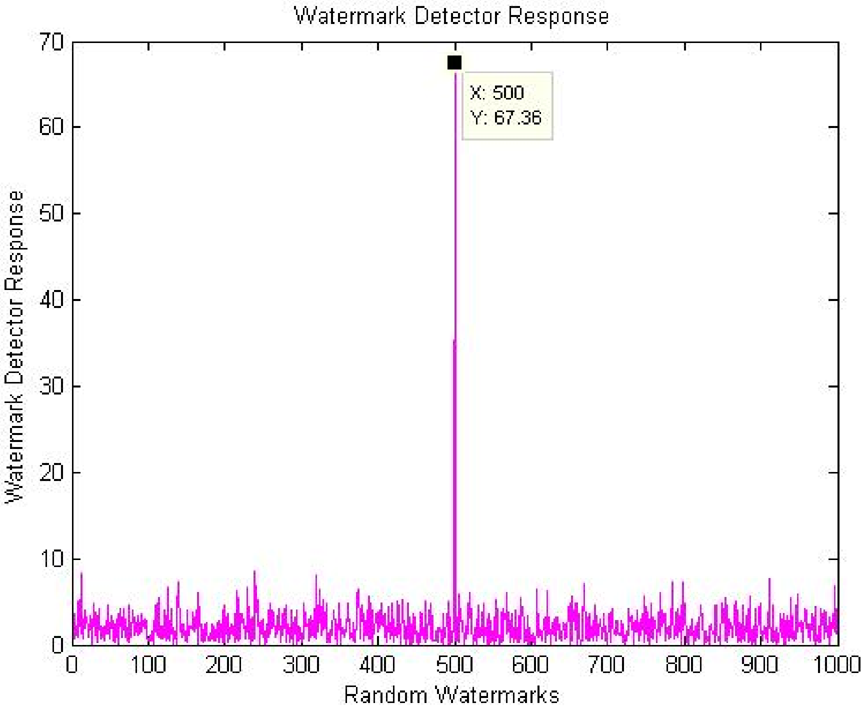


Figure 44(k): Detector response to PSC-compliant watermark in Median Filtered color image.

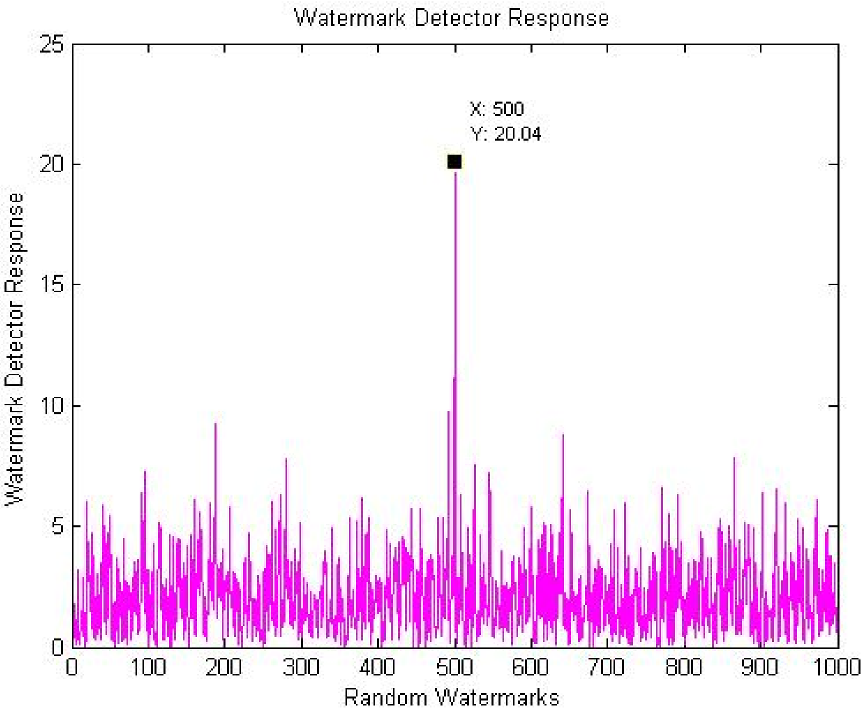


Figure 44(l): Detector response to PSC-compliant watermark in Dithered color image.

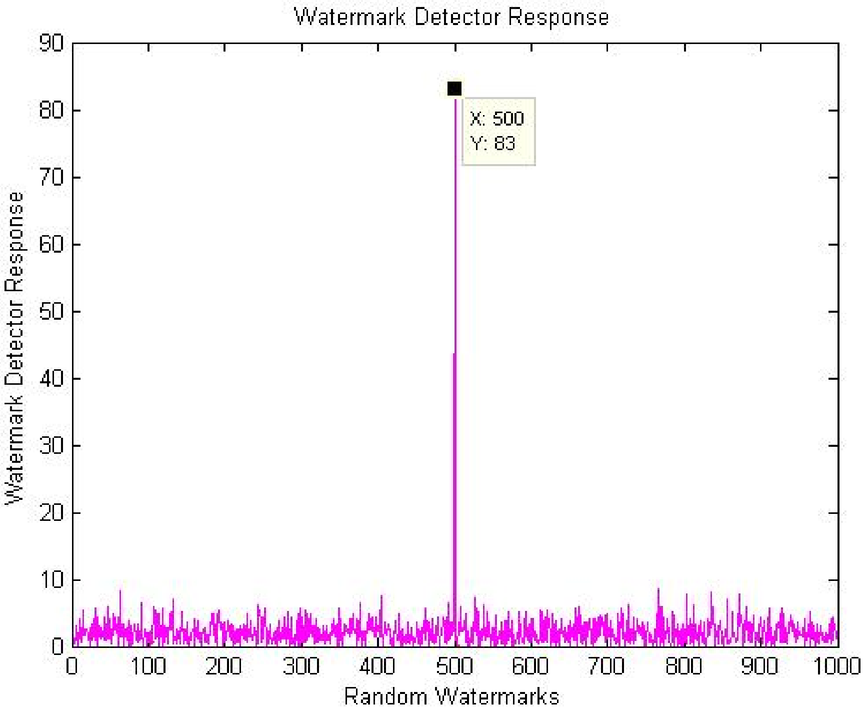


Figure 45(a): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 0.2) gray-scale image.

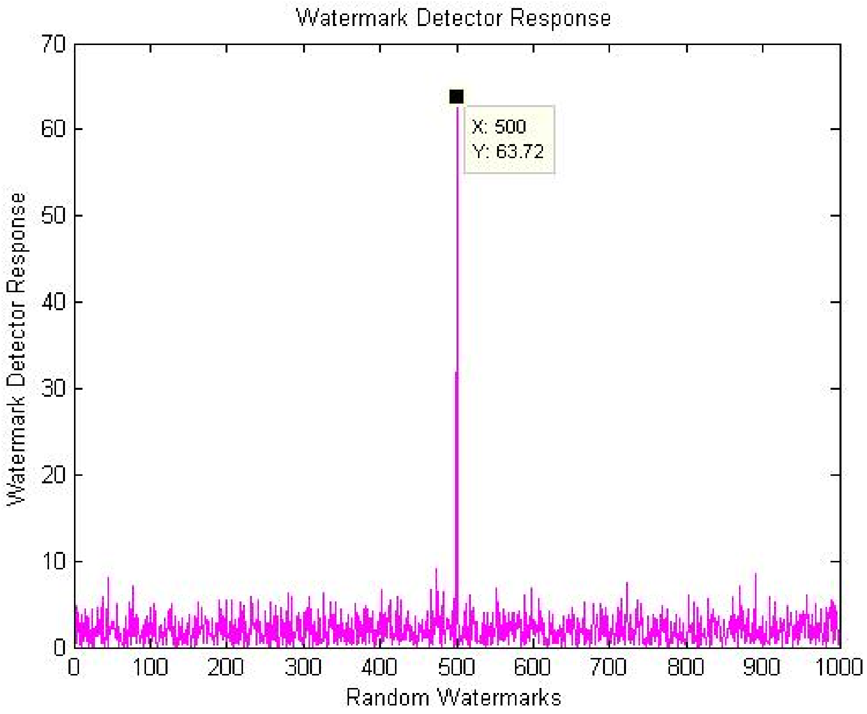


Figure 45(b): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 0.4) gray-scale image.

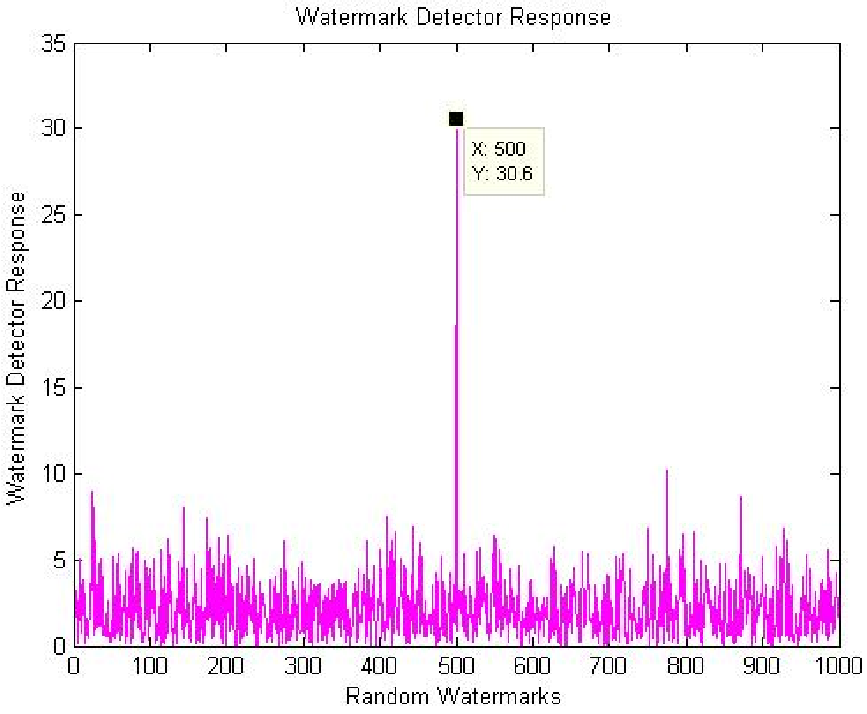


Figure 45(c): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 0.6) gray-scale image.

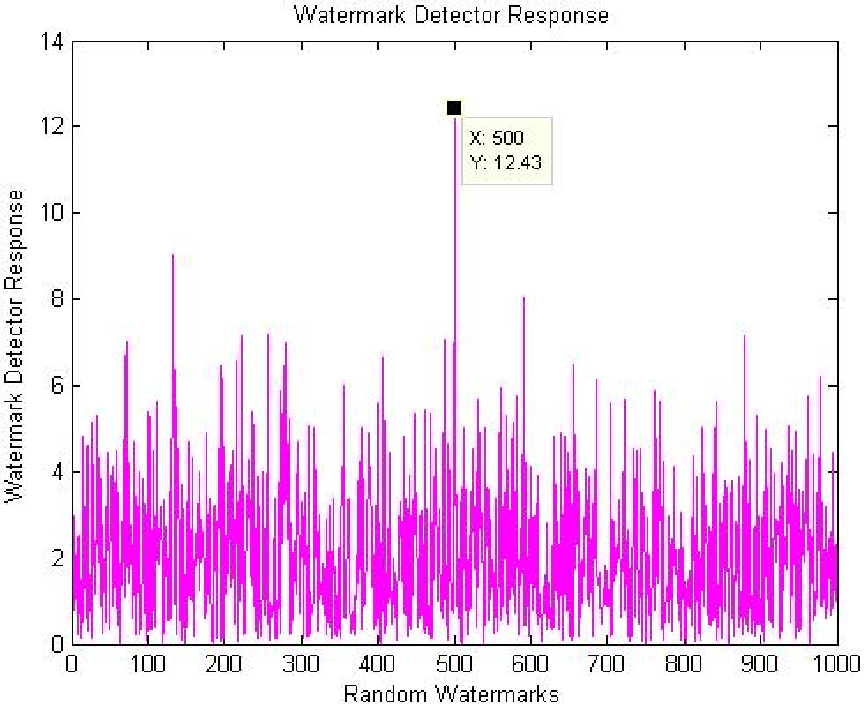


Figure 45(d): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 0.8) gray-scale image.

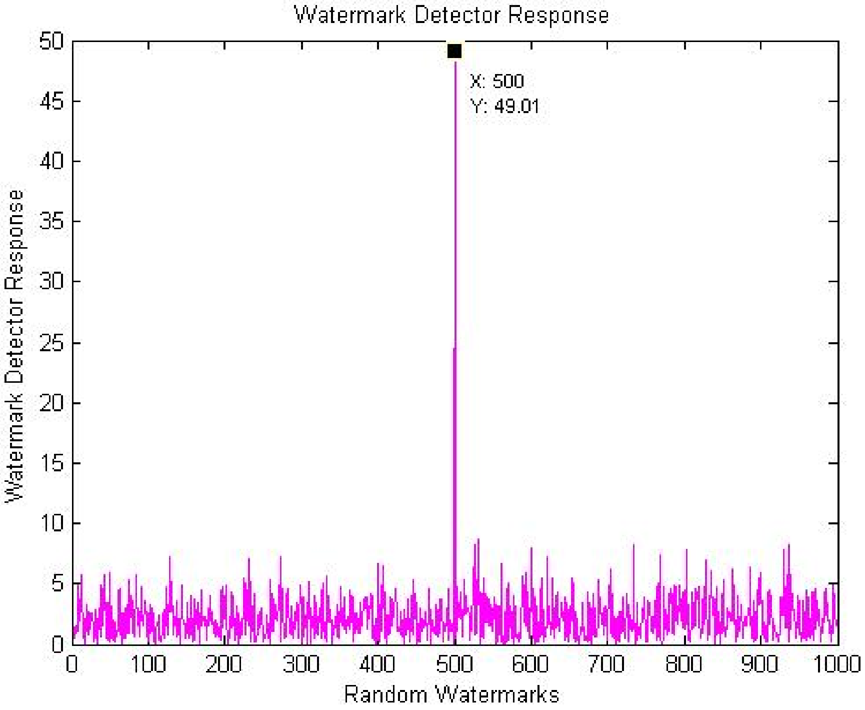


Figure 45(e): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.0) gray-scale image.

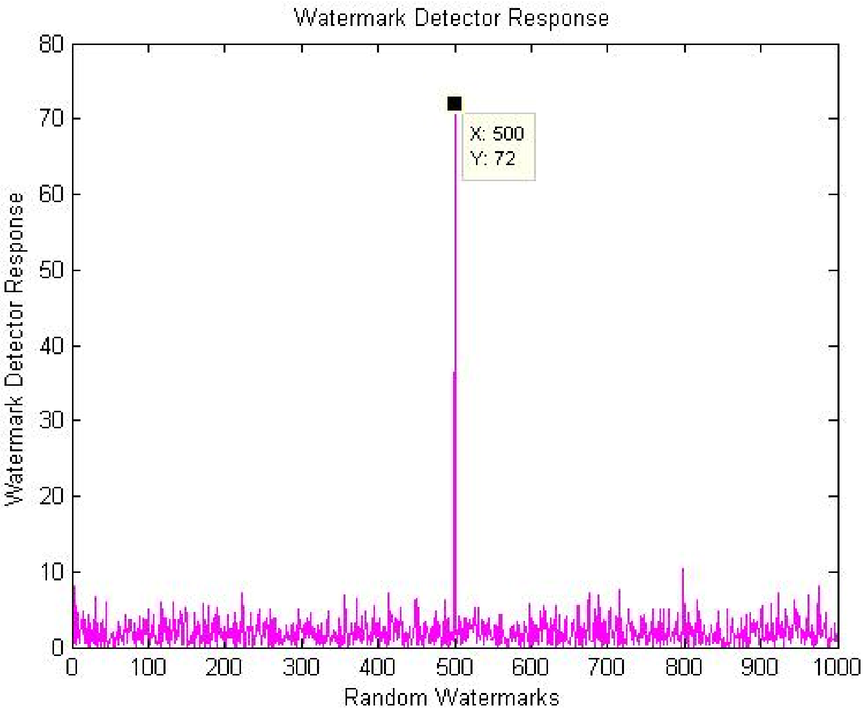


Figure 45(f): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.2) gray-scale image.

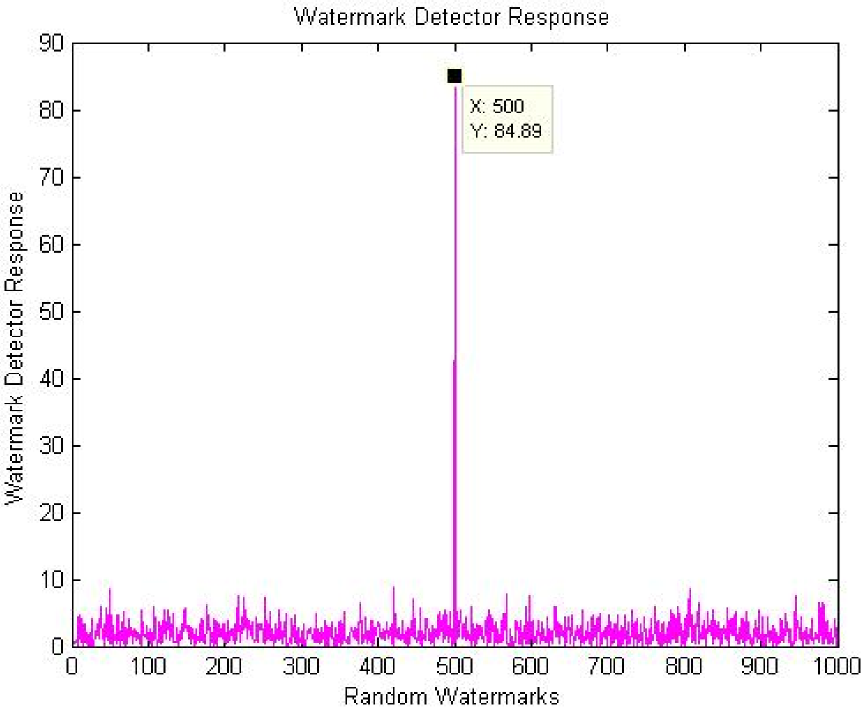


Figure 45(g): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.4) gray-scale image.

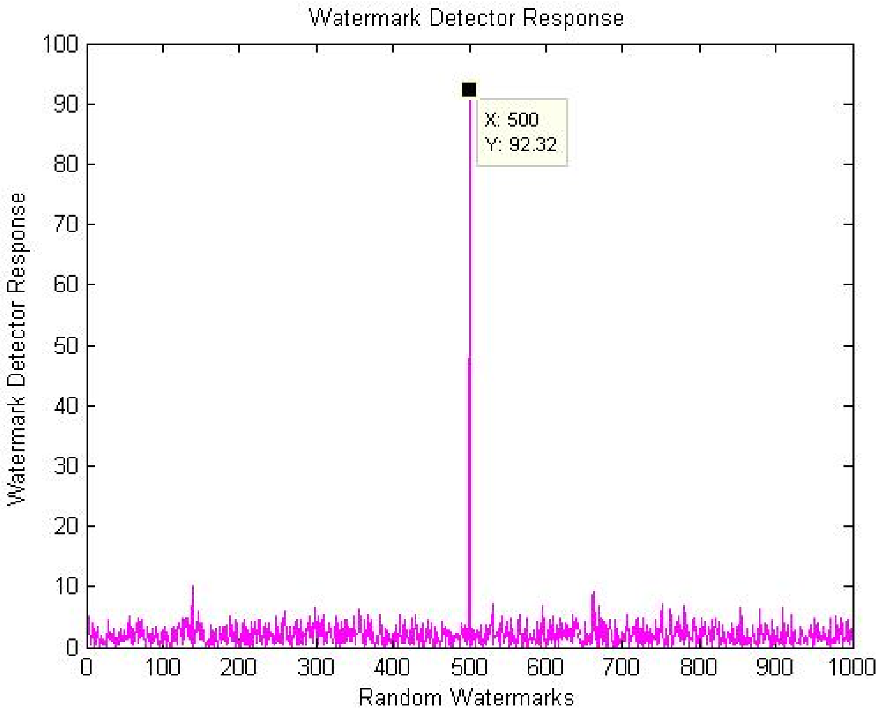


Figure 45(h): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.6) gray-scale image.

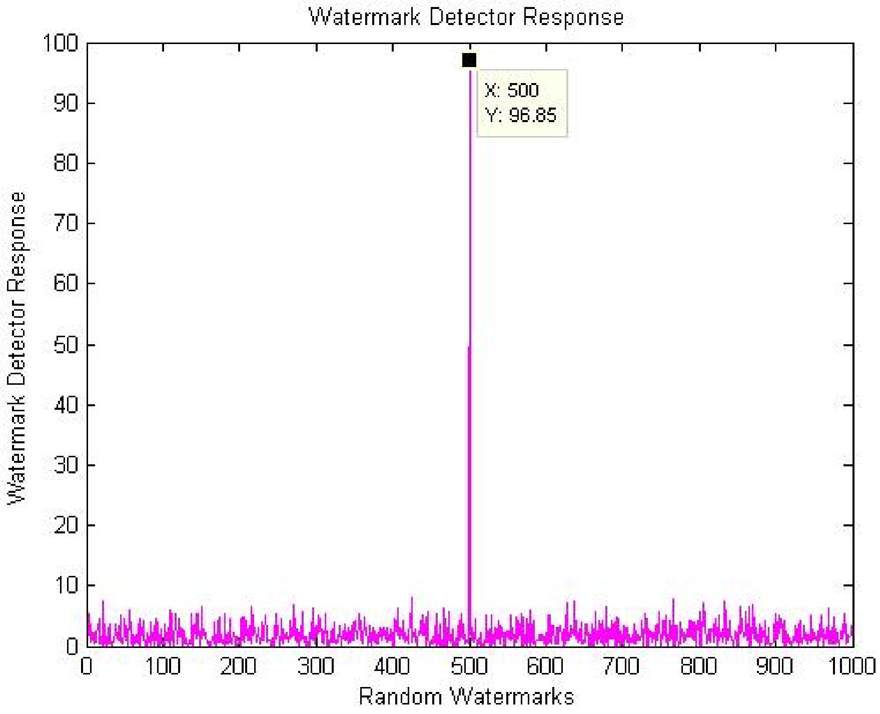


Figure 45(i): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 1.8) gray-scale image.

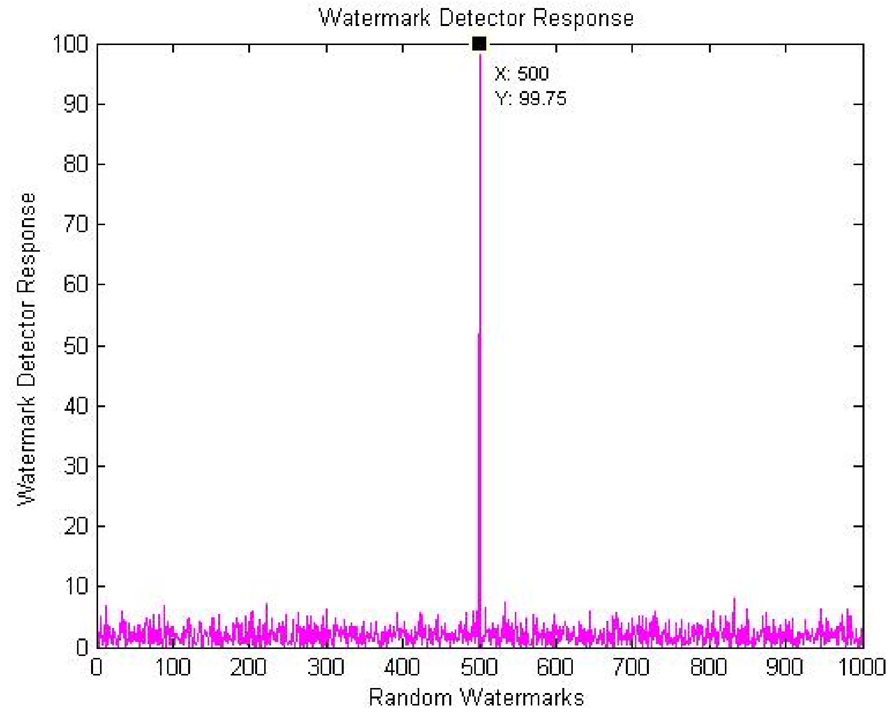


Figure 45(j): Detector response to PSC-compliant watermark in Wiener Attacked (γ = 2.0) gray-scale image.

The detector response for PSC-compliant watermarks in both gray-scale and color images are summarized in Table 3 and Table 4 for general attacks and Wiener attacks, respectively. The embedded watermark is having 2000 samples and scaling parameter α = 0.1.

S.No.	**Attack**	**Detected SNR (dB)** (Gray-scale)	**Similarity** (Gray-scale)	**Detected SNR (dB)** (Color image)	**Similarity** (Color image)
1	No Attack	19.2496	124.4864	18.0550	93.2236
2	Re-sized	16.2080	114.4424	17.5556	16.9589
3	JPEG (5%)	14.9469	92.6964	15.7821	22.7400
4	JPEG (10%)	16.1880	105.3236	16.8640	64.8941
5	JPEG (25%)	17.4149	119.4601	17.5353	85.5464
6	Rotation / Cropped	15.1288	95.7824	16.0379	21.3092
7	De-blurred	17.5262	114.5802	17.5676	54.2619
8	Salt-pepper Noise	11.9180	91.4669	14.1398	55.2356
9	AWG Noise	13.1925	91.9234	15.4693	69.3561
10	Average filtered	16.0043	109.6841	17.4254	30.5192
11	Median filtered	16.8101	115.1786	17.6450	67.3539
12	Dithered(8 colors)	0.8659	94.8945	17.6475	20.0447

Table 3: Experimental results obtained with 256x256 gray-scale and color images, using a PSC-compliant watermark.

Table 4 (shown below) summarize the watermark detector response to the Wiener attacked images, (both gray-scale and color), each with a PSC-compliant (energy-efficient) watermark. Here, γ (i.e. estimate weighing factor) is incremented in small equal steps of 0.2, and the similarity value remains much higher than the average threshold. Thus, results confirm that PSC-compliant watermarks are resilient to the Wiener attack.

S.No.	**Wiener Attack**	**SNR (dB)** (Gray-scale image)	**Similarity** (Gray-scale image)	**SNR (dB)** (Color image)	**Similarity** (Color image)
1	Wiener γ = 0.2	17.6259	124.1160	19.1911	82.9996
2	Wiener γ = 0.4	15.4946	123.3286	18.4053	63.7243
3	Wiener γ = 0.6	13.4935	122.3709	16.4118	30.5967
4	Wiener γ = 0.8	11.7518	121.3745	14.2840	12.4331
5	Wiener γ = 1.0	10.2475	120.4223	12.3851	49.0139
6	Wiener γ = 1.2	8.9384	119.5032	10.7451	72.0041
7	Wiener γ = 1.4	7.7868	118.6684	9.3289	84.8914
8	Wiener γ = 1.6	6.7618	117.8961	8.0923	92.3159
9	Wiener γ = 1.8	5.8392	117.1887	6.9987	96.8462
10	Wiener γ = 2.0	5.0031	116.5466	6.0215	99.7511

Table 4: Experimental results obtained with 256x256 gray-scale and color images, embedding a PSC-compliant watermark.

Table 5 shown below, compares the watermark detector response to the Wiener attacked images having a random (i.e. Non-PSC-compliant) and a PSC-compliant watermark, for both gray-scale and color images, respectively.

S.No.	**Wiener Attack**	**Similarity** (Gray-scale image) **(Non-PSC)**	**Similarity** (Gray-scale image) **(PSC**)	**Similarity** (Color-image) **(Non-PSC)**	**Similarity** (Color-image) (**PSC**)
1	Wiener $\gamma = 0.2$	27.8437	124.1160	26.0344	82.9996
2	Wiener $\gamma = 0.4$	20.3104	123.3286	19.5519	63.7243
3	Wiener $\gamma = 0.6$	14.3766	122.3709	14.6619	30.5967
4	Wiener $\gamma = 0.8$	10.3430	121.3745	11.3237	12.4331
5	Wiener $\gamma = 1.0$	7.5379	120.4223	9.0452	49.0139
6	Wiener $\gamma = 1.2$	5.6285	119.5032	7.4104	72.0041
7	Wiener $\gamma = 1.4$	4.1702	118.6684	6.2216	84.8914
8	Wiener $\gamma = 1.6$	3.0731	117.8961	5.2821	92.3159
9	Wiener $\gamma = 1.8$	2.2172	117.1887	4.5494	96.8462
10	Wiener $\gamma = 2.0$	1.5026	116.5466	3.9514	99.7511

Table 5: Comparison of detector response (similarity) between Non-PSC compliant watermarked images and PSC-compliant watermarked images.

Here, we observe that the watermark detector response to Non-PSC (random) watermark falls drastically as γ is increased in small steps; while the same remains much higher than the earlier one, for a PSC-compliant watermark. This confirms that Non-PSC-compliant watermarks are vulnerable to Weiner attacks, while PSC-compliant watermarks are highly resilient to such attacks.

SUMMARY AND CONCLUSION

As electronic distribution of copyright material becomes more prevalent a need for digital watermarking rises. In this project, the basic characteristics of a digital watermark are outlined; mainly including: fidelity preservation, robustness to common signal and geometric processing operations, robustness to attacks applicability to digital images.

To meet these requirements, Cox et al. [1] proposed a watermark whose structure consists of i.i.d. random numbers drawn from a distribution. The length of the watermark is variable and can be adjusted to suit the characteristics of the data. As recommended, the watermark must be placed in the perceptually *most* significant components of the image spectrum. This maximizes the chances of detecting the watermark even after common signal and geometric distortions. Further, modification of these spectral components results in severe image degradation long before the watermark itself is destroyed. Of course, to insert the watermark, it is necessary to alter these very same coefficients. However, each modification can be extremely small and, in a manner similar to spread spectrum communication, a strong narrowband watermark may be distributed over a much broader image (channel) spectrum.

I have used the scaling parameter ($\alpha = 0.1$) for my experiments with both gray-scale and color images. If we change α, it will surely affect the visual quality accordingly. It will ultimately be up to content owners to decide what image degradation and what level of robustness is acceptable. This may vary considerably from application to application.

Detection of the watermark then proceeds by adding all of these very small signals, and concentrating them once more into a signal with high SNR. Because the magnitude of the watermark at each location is only known to the copyright holder, an attacker would have to add much more noise energy to each spectral coefficient in order to be sufficiently confident of removing the watermark. However, this process would destroy the image fidelity.

In these experiments, I have embedded the watermark to the image by modifying the 1000 largest coefficients of the DCT (excluding the DC term). These components are heuristically perceptually more significant than others.

An important open problem is the construction of a method that would identify perceptually significant components from an analysis of the image and the human perceptual system. Such a method may include additional considerations regarding the relative predictability of a frequency based on its neighbors. The latter property is important in combating attacks that may use statistical analysis of frequency spectra to replace components with their maximum likelihood estimate.

I have analyzed, using the gray-scale and color images that Cox's algorithm can extract a reliable copy of the watermark from images that were degraded with several common geometric and signal processing procedures. An important caveat here is that any affine geometric transformation must first be inverted. These procedures include translation, rotation, scale change, and cropping. The algorithm displays strong resilience to lossy operations such as aggressive scale changes, JPEG compression, dithering and filtering etc.

Experimental observations highlight that Cox's Spread-Spectrum algorithm using a random watermark provides excellent features of imperceptibility and robustness with respect to many common attacks as discussed above. But, the randomly generated watermarks can be attacked by taking an MMSE estimation of the image as done in Wiener attacks. The simple models for watermarking and the Wiener attack yield insight into the structure of a watermark for improved robustness. An important assumption is the use of a fixed watermark detector that does not compensate for the effects of attack. These considerations lead to the idea of energy-efficient watermarking and provide a way to link the detectability of an attacked watermark to the distortion of the attacked signal. It then becomes possible to evaluate robustness in a meaningful way.

The key result is the power-spectrum condition (PSC), which states that a watermark is energy-efficient if and only if its power spectrum is directly proportional to that of the original signal. The watermark must be designed in accordance to the power spectrum of the original cover image. The results have proved that the PSC-compliant watermarks not only improve the watermark detector response, but also strongly defeat Wiener attacks. The PSC holds for any signals that meet the assumptions of the model. It may therefore be applicable to digital audio, images, and video, for example.

FUTURE WORK

The owner or watermarker and the attacker both follow the game theory. The one who acts smarter and can predict the other's move, poses a challenge to the opposite party and tries to win over. So, in future the attackers would surely try to improve upon their attacking strategy, or shall discover new types of attacks. Therefore, the watermarker must try to improve upon the proposed watermarking scheme. In-fact no watermark and no watermarking algorithm is perfect, that could be applicable to all types of digital data.

A PSC-compliant watermark is essentially a 2D watermark of the same size of the original cover image. However, I have implemented only a string of suitable length in my experiments and obtained satisfactory results. In future, one can take up the challenge to implement embedding and extraction of 2D watermarks of such a large size, and try to improve upon the algorithm in order to make it compatible with 2D watermarks.

Broader systems issues must be also addressed in order for this system to be used in practice. For example, it would be useful to be able to prove in court that a watermark is present without publicly revealing the original, unmarked document. It should also be noted that the current proposal only allows the watermark to be extracted by the owner, since the original un-watermarked image is needed as part of the extraction process. So, one should also research for improvements in the proposed watermarking system, in which the original image may not be essentially required during detection and extraction of the watermarks.

REFERENCES

[1] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, “Secure spread-spectrum watermarking for multimedia,” IEEE Trans. On Image Processing, Vol. 6, No. 12, December 1997.

[2] J.K. Su and Bernd Girod, “Power Spectrum Condition for Energy-Efficient Watermarking”, IEEE Trans. On Multimedia, Vol. 4, No. 4, December 2002.

[3] R.C. Gonzalez and R.E. Woods - *Digital Image Processing*; Pearson Education - Asia, 1992.

[4] J.G. Proakis and D.G. Manolakis - Digital Signal Processing – Principles, Algorithms and Applications; Third Edition; Prentice Hall of India, 2003.

[5] S.K. Mitra – Digital Signal Processing – A Computer Based Approach; Tata-McGraw Hill.

[6] S.J. Chapman – MATLAB ® Programming for Engineers, Second Edition; Thomson 2003.

[7] E. Koch and Z. Zhao, “Toward robust and hidden image copyright labeling,” in *Proc. 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, June 1995.

[8] R. L. Pickholtz, D. L. Schilling, and L. B. Millstein, “Theory of spread spectrum communications—A tutorial,” *IEEE Trans. Commun.*, vol. COMM-30, pp. 855–884, 1982.

[9] K. Tanaka, Y. Nakamura, and K. Matsui, “Embedding secret information into a dithered multi-level image,” in *Proc. 1990 IEEE* pp. 216–220.

[10] A. Piva, M. Barni, F. Bartolini, and V. Cappellini, “DCT-based watermark recovering without resorting to the uncorrupted original image,” in Proc. IEEE Int. Conf. Image Processing, Santa Barbara, CA, Oct. 1997, pp. 520–523.

[11] J. K. Su and B. Girod, “Power spectrum condition for energy-efficient watermarking,” in Proc. IEEE Int. Conf. Image Processing, Oct. 1999.

[12] C. I. Podilchuk and W. Zeng, “Image-adaptive watermarking using visual models,” IEEE J. Select. Areas Commun., vol. 16, pp. 525–539, May 1998.

[13] IEEE website : www.ieee.org