Chapter 22: HDR Image Watermarking

Fabrizio Guerrini, Masahiro Okuda, Nicola Adami, and Riccardo Leonardi

22.1. A Brief Introduction to Digital Watermarking

This Section introduces the basic notions of digital watermarking, to give to the reader a self-contained, succinct yet quite complete coverage of digital watermarking basics while presenting the watermarking system in a way congenial to the rest of this Chapter.

Digital watermarking [Barni(2004)] [Cox(2008)] belongs to the *data hiding* field. Data (or information) hiding is a field as old as History itself [Cox(2008), Chapter 1]. One can hide an object, as a piece of information, for many possible reasons to obtain the more disparate intended results. The most common reason for hiding information is to protect it against inappropriate or prohibited or sometimes even perfectly licit uses from subjects who have not the authority to do so. A different but somewhat correlated reason is secrecy: one may want to hide informations to keep their very existence unknown (secrecy is correlated to protection since the former can be a mean for protection on its own and because secrecy and protection are often simultaneously present, and sometimes confused, in real world applications). Hidden information may also be utilized as in surveillance systems to trigger some actions in response to subjects, unaware of their presence, performing (potentially illicit) operations, this way constituting a repressive rather than a preventing way of protection. Last but not least, informations could be hidden because, in spite of their perhaps indispensable presence, should they be in plain sight in some cases they would degrade the perceptual value of the object they correspond to. An excellent introduction to data hiding history in its totality can be found in [Petitcolas(1999)]. For a mathematical analysis of the problem of secret communication, see the classic paper by Shannon [Shannon(1949)].

Digital data hiding is the direct derivation of these concepts into the digital world. Since digital content is only another representation of the same information the human senses naturally perceive, it is quite natural that the same needs as the ones previously cited arise in the digital world as well. Hence, digital information hiding applications account to hide some kind of digital information into digital documents.

Digital information hiding can be thought as a composition of three main techniques: cryptography, steganography and digital watermarking. The best known branch of information hiding is probably *cryptography* [Menezes(1996)], which is the art of hiding the content of a transmission between two subjects to a potentially malicious eavesdropper by making it unintelligible to all except the intended recipients. Another very old, though less known than cryptography, information hiding technique is *steganography* [Provos(2003)], sometimes also called covert communication. In steganography, the very existence of the communication is hidden; the information is conveyed by proper, imperceptible modifications of an innocent-looking object. The most recent information hiding technique, *digital watermarking*, is the subject of this Chapter, applied in the high-dynamic range imaging context.

Loosely speaking, digital watermarking tries to introduce information in a certain domain (this process is called *watermark embedding* [Barni(2004), Chapter 4]) inside a digital object while preserving its perceptual content (like steganography) and while being in a "hostile" environment (like cryptography), i.e. populated by intelligent attackers (embodying the so-called *watermark channel* [Barni(2004), Chapter 7]) interested in disrupting the watermarking system operativeness while preserving the perceptual quality of the host object. This information then will be retrieved by another entity, or perhaps the same one

F. Guerrini, N. Adami and R. Leonardi are with the Signals & Communications Laboratory - Department of Information Engineering, University of Brescia, 25123 Italy. Email: {firstname.surname}@ing.unibs.it.

M. Okuda is with the Department of Information and Media Sciences, University of Kitakyushu, 808-0135 Japan. Email: okuda-m@env.kitakyu-u.ac.jp.

that performed the embedding process, thus performing the so-called *watermark recovery* [Barni(2004), Chapter 6].

The need for digital watermarking first rose as an answer to the inherent deficiencies of the existing information hiding techniques to counter the digital multimedia piracy problem. In this case, a digital content owner wants to protect it, e.g. against unauthorized copying. Steganography is useless in this case since the potential pirate is well aware of the owner intentions and methods, thus failing the main hypothesis of the steganographic model; and cryptography can only make sure the encrypted content is not eavesdropped during its distribution but can do nothing more when the content is finally decrypted for consumption by its intended recipient. Digital watermarking, then, could offer a solution to this problem, at least in principle, since it is tied within the content itself.

Later on, it has become clear that digital watermarking can also be used in totally different application contexts [Cox(2008), Chapter 2] [Barni(2004), Chapter 2]. For example, broadcast monitoring refers to the ability by a broadcaster to have precise reports of which shows are aired and how often they are, for reasons ranging from royalties collecting to marketing studies. It includes as applicative environments digital TV broadcasting as well as Internet TV services and has recently surfaced as a critical problem due to the proliferation of available channels. The main target of these recently released products are the identification of copyrighted content made available on Internet by online viewers or P2P networks, at the very least to ask for their remotion.

Other possible watermarking applications do not deal with intellectual property rights protection at all. As the watermark represents a side channel to convey information that is attached to the content, it could also carry useful metadata instead. For example, metadata watermarks could simplify content-based retrieval of multimedia objects. A related application which is experiencing a growing interest is embedding linking information to enable a range of e-commerce services. The distinct advantage of using watermarking to attach metadata to a piece of content is its robustness to D/A and A/D conversions, processes that usually result in losing any other type of metadata such as header-based ones. The ability to insert as much information as possible in the host object is the highlight requirement, along with issues related to implementation such as complexity and speed of execution.

A rather ambitious proposed application of digital watermarking is enhanced coding of digital content, both source coding and channel coding. It is not clear from a theoretical point of view whether it could boost coding performance, especially in practical scenarios; nevertheless, some authors argue that some advantage is to be expected by applying data hiding techniques for content coding. For the source coding part, it has been proposed that the watermarking could help the compression process, achieving better compression by substituting (a lossless operation) the imperceptible part of the content with information data about the content that therefore have no more to be stored in the bitstream. Digital watermarking could also be used for channel coding, i.e. to counter transmission errors that are especially harmful to compressed content. It could be a valuable alternative to error concealment at the decoder side (to approximate lost information by means of some sort of filtering) and/or redundant coding at the encoder side (via error correcting codes), which usually suffer from backward compatibility issues. In the case of using a digital watermark to embed redundant information, compatibility is automatically achieved since it can be safely ignored by the decoder. It is not clear at this point if the PSNR distortion (the traditional way to evaluate the goodness of a coding algorithm) introduced by the watermark is better than those achievable by means of the other techniques.

The first domain considered by the digital watermarking community, and undoubtedly the most studied even today, is the one pertaining to still digital low dynamic range (LDR) images. Digital watermarking has then been applied to other domains as well, especially audio, but also video (often by using the methods designed for still images) and more exotic domains such as text and 3-D meshes. As we shall see, the most recent entry in this list is still digital high dynamic range (HDR) images.

A. *Digital Watermarking Requirements*

The requirements of a digital watermarking system, whose simultaneous presence distinguishes it from the other data hiding techniques, are *capacity*, *robustness/security* and *imperceptibility*. Each of these requirements are discussed in the followings.

As is often the case, the system designer must handle an application-dependent trade-off between the requirements. To picture their conflicting nature, they are often drawn on a so-called trade-off triangle, as in Fig. 22.1, to stress the fact that trying to favor one of these requirements always damage to some extent one or both of the other two. Note that robustness and security have been drawn on the same vertex; this can be acceptable in general, given that the boundary between this two requirements is very subtle (and even not considered by some authors). However, to be specific, security and robustness are sometimes themselves conflicting requirements, when treated as separate objectives (and we argue below that this should be indeed the case), so that the trade-off triangle could actually be a trade-off tetrahedron.

The capacity [Barni(2004), Chapter 3] is the quantity of information (usually measured in bits) that the watermark is able to convey. It is generally dependent on the size of the host object, thus sometimes the capacity is expressed as a relative entity (e.g. in the case of images the unity is bit of information per pixel or bpp). This way of expressing capacity is particularly common in steganography, where capacity is often the primary concern. Vice-versa, the capacity of a watermark more strongly depends on its application; for example, an image watermark capacity could range from a single bit in the case of detectable watermarking (see below) to some thousand of bits for a single host image, while in steganography it is usually some fraction of a bpp or more (which, given the number of pixels of an ordinary image, is several order of magnitude higher). Hence, the watermark capacity must be considered tailored for the intended application aim.

Once the watermark has been embedded into a host object, sooner or later it must be retrieved. How this can be achieved even after the host watermarked object has been possibly processed is referred under the watermark security and watermark robustness terms. There is some confusion in the early literature about the definition of these two requirements. The philosophy adopted here, that is to clearly distinguish security and robustness as separate and conflicting requirements, is probably the most acceptable in the authors' opinion.

Robustness [Cox(2008), Chapter 9] refers to the ability of the watermark to survive non-malicious data processing the host object happens to undergo. These processing are not intended to remove the watermark but are applied to the host object for some other purpose. The set of acceptable data processing must be decided prior to designing the watermarking system and obviously depends on the nature of the host object. For classical LDR images, lossy compression, noise addition, D/A and A/D conversion, geometric transformations (such as zooming, rotation and cropping), linear and non-linear filtering for

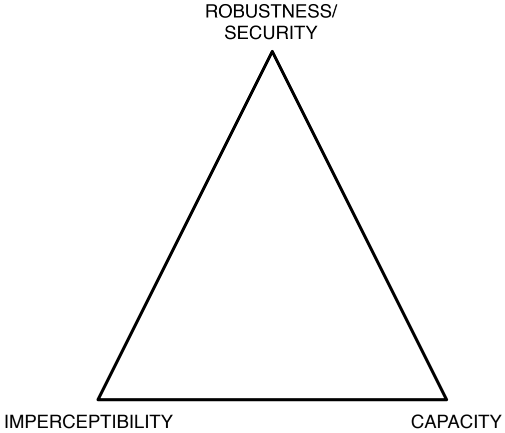


Figure 22.1: Digital watermarking trade-off triangle, depicting how the requirements on the vertices are conflicting.

image enhancement and histogram modifications are all examples of possibly non-malicious processing that can occur.

Security [Cox(2008), Chapter 10] instead is related to the inability by an hostile entity (referred to as the attacker) to remove the watermark or disable in some way its recovery. The most basic instance of security, that is usually present even in watermarking systems which do not explicitly address security, is that only authorized entities are allowed to embed in or recover watermarks from host objects. This in turn implies that both the embedding and the recovery processes must in some way depend on the knowledge of some *secret key*. In a well-designed system, security must obey Kerckhoffs' principle [Kerchoffs(1883)] [Menezes(1996)] much like in cryptography, that is the security of the system must be assured considering the attacker aware of all the system details; the only thing he/she ignores is the secret key.

Some author refers to security in a way related to cryptanalysis, which means it should be impossible for an attacker to guess the secret key. However, we prefer to use our previous definition of security since the former is only a (particularly harmful) instance of the latter, since an attacker could also be interested in simply disabling or removing any watermark present without knowing the secret key.

Finally, it is worth noting that repeated or strong use of a certain processing tool initially considered as non-malicious (maybe the one against which the system is the most vulnerable) could be effectively considered a security attack, making the boundary between security and robustness very fuzzy. Therefore it can be tempting to put both these requirements on the same vertex of the trade-off triangle, but we believe that neglecting the more sophisticated intelligence of a determined attacker can hurt the deployability of a watermarking system in a security-critical application. In our view robustness and security are conflicting requirements in the sense that robustness tends to exploit any possible perceptual niche of the host object to survive processing, while security tries to make the watermark characteristics and location as unpredictable as possible.

Last, the watermark has to be imperceptible [Cox(2008), Chapter 8], that is it should not degrade the perceptual content of the host object. Starting from this definition, it is obvious that the perceptibility is a subjective matter so that it is impossible to give a universal measure of perceptibility. The best way to handle perceptibility is to study how human perceive the nature that he/she is surrounded with by means of models that approximate the mechanisms underlying perception. These models, primarily the HAS (Human Auditory System) and the HVS (Human Visual System), have been extensively studied in the past for the field of digital compression. As a matter of fact, multimedia data compression tries to remove perceptually irrelevant parts of the original data to decrease the amount of information that must be conveyed to reproduce an acceptable "quality" of the data, so it is very important to include perceptual cues. In a sense, digital watermarking and data compression could be thought as dual problems: the former in fact, since it has to be imperceptible, must reside in the field of imperceptible data, just the ones that compression tries to eliminate from the original data. Unfortunately (or luckily depending on the point of view) the perfect perceptual compression is not achievable and therefore watermarks could be accommodated in imperceptible "niches" left by compression. Hence, to achieve imperceptibility digital watermarking could exploit a whole mass of knowledge borrowed from multimedia compression.

Many systems do not explicitly use any perceptual model (which are usually difficult to implement), but instead rely on more classic approaches based on standard metrics (e.g. PSNR) to minimize perceptual impairments; however, care has to be taken when comparing human perception with these kind of absolute distortion measures. It is also very common to guarantee imperceptibility by the mere selection of an appropriate watermark domain.

An exception of the general rule of imperceptibility is *visible watermarking*, where the watermark is rendered perceptible to assess its presence (maybe for informative purposes, much like a logo) while retaining all the other watermark characteristics. In any case, even if the watermark is visible, there is a certain amount of distortion that the embedding process cannot exceed on the host object, so that even in this case it is possible to define the imperceptibility requirement with some slight modification.

B. Watermarking System Examples

To help conveying the basics of watermarking systems as described above, we first provide a brief exposition of a couple of classic algorithms before introducing the watermarking system structure in general and abstract terms in the following Sections.

The work in [Cox(1997)] introduces a very used watermarking paradigm known as spread-spectrum watermarking. The simplified flowcharts of both the watermark embedding and recovery processes are depicted in Fig. 22.2. The watermark is constructed by seeding a pseudorandom number generator with the secret key and then extracting a Gaussian random sequence w. It is argued that to render the watermark robust against common processing such as compression, the best way is to insert the watermark in the most perceptually significant portion of the host object. Referring to the still image case, a full-frame 2-D DCT is computed and represents the watermark domain. Next, a perceptual mask is computed to identify the most perceptually significant coefficients, referred as the vector v. The watermark is then introduced to obtain the watermarked coefficients v' applying one of the following formulas:

$$v'_i = v_i + \alpha_i w_i \tag{22.1}$$

$$v'_i = v_i(1 + \alpha_i w_i) \tag{22.2}$$

for $i = 1, \ldots, n$ where n is the watermark length. The scaling factors α_i should be selected so as to ensure imperceptibility and thus can be dependent of the value v_i. The scheme's target is only to assess if a particular watermark is present or not (see detectable watermarking below). The above formulas are sometimes referred to as additive spread-spectrum and multiplicative spread-spectrum watermarking respectively.

The watermark recovery is non-blind (see below), that is the original, unwatermarked object is necessary to the entity that performs the recovery. To determine if the watermark is present, the received watermarked object (which is possibly further processed) undergoes the same 2-D DCT transform and the obtained coefficients are then subtracted to the original ones to obtain the recovered coefficients w^* and then a

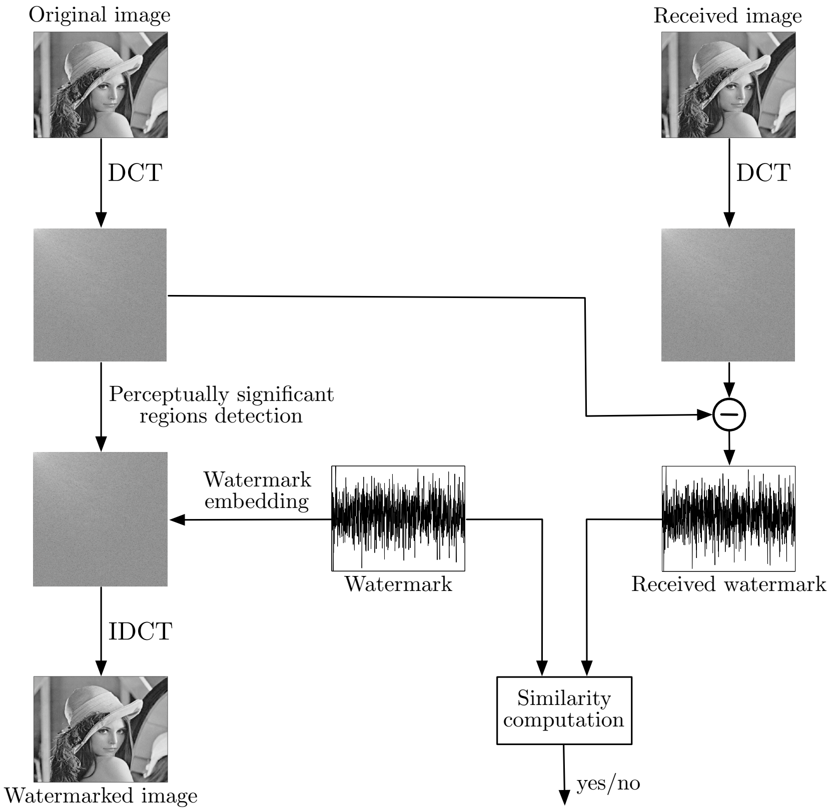


Figure 22.2: Example of a spread-spectrum watermarking process, the one presented in [Cox(1997)].

normalized correlation (or similarity measure) is computed as:

$$\mathrm{Sim}(w, w^*) = \frac{w \cdot w^*}{\|w\| \cdot \|w^*\|} \tag{22.3}$$

Using Gaussian distributed watermark coefficients increases security because it counters so-called collusion attacks, in which the attacker averages several watermarked images in the hope of obtaining an unwatermarked object: in this case all the watermarks are still present simultaneously. However, it should be noted that an attacker can apply the same perceptual mask to identify which coefficients carry the watermark and then perform more subtle attacks in the DCT domain. Choosing just a random portion of all coefficients would increase security in this sense, but would surely hurt robustness.

The second watermarking system example we'd like to discuss is Quantization Index Modulation, or QIM, the precursory of many techniques that have appeared and are still appearing in the literature. It has been proposed by Chen and Wornell in [Chen(2001)]. QIM consists in a quantization of the host object features using a particular codebook Q associated to a given watermark. To be more specific, suppose we have a set U of $2^{|\mathbf{b}|}$ different quantizers, each identified by a specific string associated to a binary string $\mathbf{b}$ of length $|\mathbf{b}|$. If one wish to embed the watermark code $\overline{\mathbf{b}}$ in the host object, the latter's features must be quantized using the correspondent quantizer from the set U obtaining the quantized (watermarked) features. When the watermark is to be retrieved, the received object features are re-quantized using the entire codebook set U (since the retriever does not know in advance which quantizer has been used in the first place) and then identifies to which particular codebook the quantized value belongs by taking the one in the entire set U with a quantized value at minimum distance, thus retrieving $\overline{\mathbf{b}}$.

A very commonly found subset of QIM algorithms are SQIM (Scalar QIM), also referred as DM (Dither Modulation) watermarking. In these systems the reconstruction values associated to all the codebooks forming U are arranged in a regular, rectangular lattice; in this way, one could perform scalar feature quantization, one feature at at time, thus avoiding vector quantization processes. A rearrangement of QIM to make it adhere to the SQIM paradigm is illustrated in Fig. 22.3, where two features are represented on each axis. As observable, now it is possible to perform feature quantization separately along the two axis, with in general two different quantization steps, as every bit of $\mathbf{b}$ is embedded in a single feature. In this case, the watermark code $\mathbf{b}$ is only 2 bits long, so U consists of 4 distinct quantizers, each represented by a different symbol. The legend on the right describes the relation between each watermark code and its

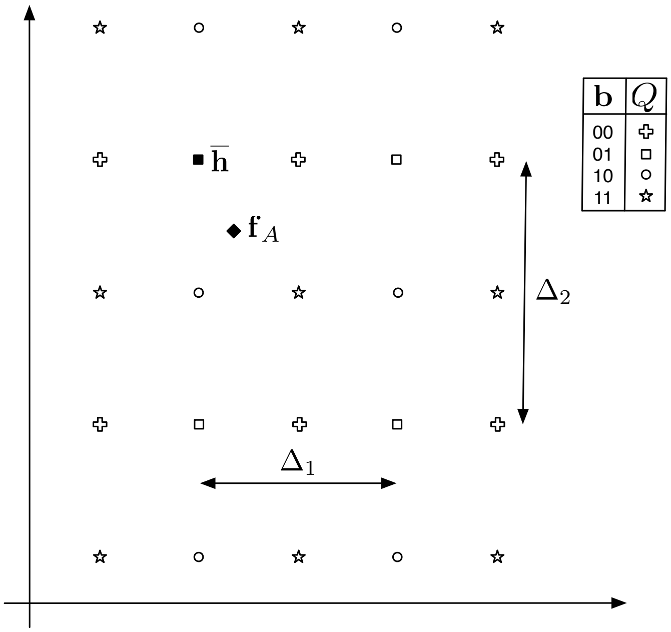


Figure 22.3: QIM example, specifically a 2 bits scalar QIM is depicted.

correspondent quantizer. The host feature point $\mathbf{f}_A$ is represented by the filled diamond. Assuming that we wish to embed the code $\overline{\mathbf{b}} = 01$, only the quantizer individuated by the squares is used for the embedding and so the filled square is selected as the quantized feature $\overline{\mathbf{h}}$ (with the watermark embedded *because* it is a square). Note that by no means the quantized feature point $\overline{\mathbf{h}}$ is the closest to $\mathbf{f}_A$ for the entire codebook U but only for the suitable quantizer Q. When, due to the watermark channel, this feature point will be moved (hopefully in a close neighborhood), its re-quantization using all the symbols performed by the decoder will output which symbol was used during the embedding process, hence allowing to identify $\overline{\mathbf{b}}$. Note that as long as the codebook is known during the recovery phase, there is no need to use the original unwatermarked object (blind recovery, see below).

The trade-off triangle is easily observable in QIM systems. Robustness, heuristically, depends on the mutual distance between the reconstruction values belonging to different quantizers; imperceptibility depends on the quantization step Δ because it is related to how much the host feature point is moved; and capacity refers to the number of different codebooks available. As it is obvious they are conflicting requirements, since increasing robustness means to move away reconstruction values of different codebooks, but this in turn affects imperceptibility; and increasing capacity increases the density of symbols, worsening robustness, whereas lowering the density increases the quantization step, harming imperceptibility. As a final note, security in QIM systems is usually achieved by shifting the codebook U by a random quantity, extracted from a uniform distribution to increase the uncertainty on its value, and depending on the secret key, so that an attacker could not tell which symbol has been used in the quantization process. Since adding this shift, which can be shared thanks to the knowledge of the secret key, has no effect on the other requirements and is easily implemented, this solution is very commonly adopted.

C. Structure of a Watermarking System

In the literature the watermarking system structure has been proposed in many ways; here we will adopt the view of considering the watermarking game as a digital communication one. The basic flowchart of a digital watermarking system is illustrated by Fig. 22.4. In this high-level flowchart only mandatory variables, i.e. variables always present in every watermarking system, are depicted (an exception is the dashed optional input, which is represented because it is almost always employed).

The message (which can be represented as a bit string $\mathbf{m}$ without loss of generality) is the main input to the system; the objective of the watermarking game is to guarantee that it is correctly received at the end of the transmission chain. The watermark embedder introduces the message into the host object A following the suitable mix of watermarking requirements a priori selected by the system designer, producing a watermarked object A_w. As previously mentioned, this process is very often driven by a secret key K to ensure a certain amount of security: for example, the seed of a pseudorandom generator of the watermark as in spread-spectrum watermarking. After the embedding stage, the watermarked object A_w possibly undergoes some processing (both malicious and non-malicious attacks, or no attacks at all) which is overall modeled by the so-called watermark channel. Finally the watermark recovery is performed

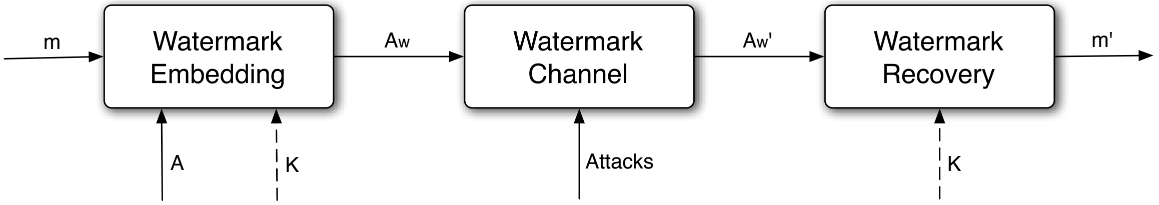


Figure 22.4: Elementary structure of a digital watermarking system. A watermark message $\mathbf{m}$ is embedded into a host object A, optionally using a secret key K. The watermarked object A_w goes through the channel in which it possibly undergoes some attacks. The recovery is then applied on the resulting $A_{w'}$. The output is the estimated watermark message **m'**.

on the "received" object A'_w, aided by the secret key K which was shared with the embedder (if used), giving as output a bit string $\mathbf{m}'$ which represents an estimate of the original message.

1) Watermark embedding: The embedder block objective is to produce the watermarked object A_w; this can be summarized by the following formula:

$$A_w = \mathcal{E}\left(\mathbf{m}, A[, K]\right) \tag{22.4}$$

where $\mathcal{E}(\cdot)$ is referred to as the *embedding function*: Eq. (22.1) constitutes an example of an embedding function. Notice how we consider the secret key K as an optional variable (enclosed in square brackets), to be coherent with Fig. 22.4. Depending on how the task of Eq. (22.4) is implemented, we can distinguish between two different types of watermark embedding (and for extension of watermarking systems altogether): *waveform-based watermarking* and *direct embedding watermarking*.

Fig. 22.5 depicts the typical steps of a waveform-based embedding stage. Additive and multiplicative spread-spectrum watermarking can be classified as based on a waveform-based embedding process. First the message code $\mathbf{m}$ is coded into a bit string $\mathbf{b}$ (the watermark code or simply the *watermark*) using a code $\mathcal{C}$; this operation is not always present and in this latter case $\mathbf{m} = \mathbf{b}$.

After the preliminary message coding step, the embedding function is applied. Usually, the watermark domain is different from the host object domain, that is to say the watermark embedding is accompanied by a feature extraction process $\mathcal{F}(\cdot)$ which transforms the host object A into a set of original host features $\mathbf{f}_A$ (the feature space is the *watermark domain*). For example, the full-frame 2-D DCT coefficients represent the watermark domain in our example spread-spectrum system [Cox(1997)]. Correspondingly to $\mathcal{F}(\cdot)$, a *watermark coding* $\mathcal{W}$ must take place beforehand, which transforms the watermark $\mathbf{b}$ in a suitable *watermark signal* $\mathbf{w}$ (alternatively called watermark waveform, see Eq.(22.1) and Eq.(22.2) for examples), expressed in the feature domain, which is well suited to be embedded in the description of the host object A carried by its features. The embedding function $\mathcal{E}(\cdot)$ then can be thought in this case as a *mixing* $\oplus$ of some kind of the watermark signal $\mathbf{w}$ with the host features $\mathbf{f}_A$ to obtain the watermarked features $\mathbf{f}_{A_w}$. Note that not all the host features $\mathbf{f}_A$ need to be mixed with the watermark signal $\mathbf{w}$ (i.e. the watermark signal dimensionality need not to be the same of that of the host features). The watermarked object A_w is finally obtained by a reverse mapping function $\mathcal{F}^{-1}(\cdot)$ from the watermarked features $\mathbf{f}_{A_w}$ to the host object domain (the inverse DCT in our spread-spectrum example). When the watermark domain coincides with the host object domain (for example, image watermarking in the pixel domain which works directly on pixel values), the feature extraction and its inverse revert to an identity function. This whole procedure could be illustrated as (optional arguments are again enclosed in square brackets):

$$\mathbf{b} = \mathcal{C}(\mathbf{m}) \tag{22.5}$$

$$\mathbf{w} = \mathcal{W}(\mathbf{b}, [A, K]) \tag{22.6}$$

$$\mathbf{f}_A = \mathcal{F}(A[, K]) \tag{22.7a}$$

$$\mathbf{f}_{A_w} = \mathbf{f}_A \oplus_{[K]} \mathbf{w} \tag{22.7b}$$

$$A_w = \mathcal{F}^{-1}(\mathbf{f}_{A_w}[, K]) \tag{22.7c}$$

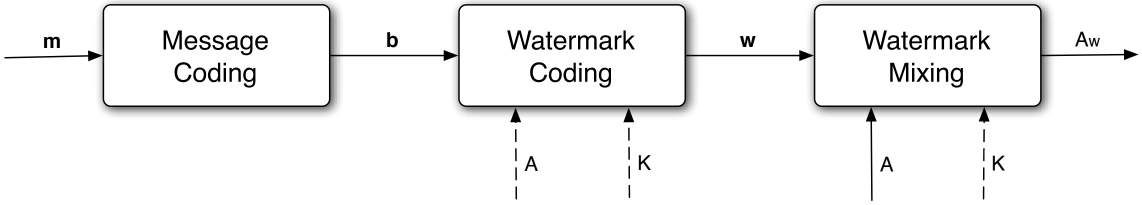


Figure 22.5: Waveform-based watermark embedding steps.

Eq. (22.5) and Eq. (22.6) both pertain to the message coding and watermark coding blocks depicted in Fig. 22.5 respectively, while the operations referred to as Eq. (22.7) are performed by the mixing block.

The secret key K could drive both the watermark coding and the watermark mixing processes. For example, the watermark signal $\mathbf{w}$ could be randomly selected from a set of possible waveforms according to K; the secret key could also randomly select which features the watermark signal have to be mixed with and which not or randomize the feature extraction process itself out of a predetermined set.

On the other hand, in direct embedding techniques there is no watermarking signal defined prior to the host features manipulation. They are instead described by Fig. 22.6, where with respect to Fig. 22.5 the watermark coding step is missing. Hence, the bit string $\mathbf{b}$ is embedded directly into the host object A by modifying in a controlled way the host features $\mathbf{f}_A$. The QIM paradigm that we described above falls into this category. The set of equations describing the direct embedding paradigm is given here:

$$\mathbf{b} = \mathcal{C}(\mathbf{m}) \tag{22.8}$$

$$\mathbf{f}_A = \mathcal{F}(A[, K]) \tag{22.9a}$$

$$\mathbf{f}_{A_w} = \mathcal{E}'(\mathbf{b}, \mathbf{f}_A[, K]) \tag{22.9b}$$

$$A_w = \mathcal{F}^{-1}(\mathbf{f}_{A_w}[, K]) \tag{22.9c}$$

Message coding of Eq. (22.8) is the same and serves the same purpose as Eq. (22.5). In this case, therefore, what is really different with respect to the waveform-based approach is the absence of any operation similar to Eq. (22.7b). Instead of mixing with a pre-defined signal $\mathbf{w}$, there is a function $\mathcal{E}'(\mathbf{b}, \mathbf{f}_A)$, described by Eq. (22.9b), which moves the host features to the watermarked features $\mathbf{f}_{A_w}$ in a way depending both from the initial position $\mathbf{f}_A$ and from $\mathbf{b}$. Referring to Fig. 22.3, the host feature point $\mathbf{f}_A$ must be moved to the new watermarked feature point $\overline{\mathbf{h}}$ without explicitly defining a watermark signal $\mathbf{w}$. This process usually involves minimizing a cost function tied to the introduced perceptual distortion, possibly using an iterative algorithm (for SQIM, searching for the "nearest" quantized value).

2) Watermark channel: The watermark channel represents all the transformations applied to the watermarked object A_w before the watermark recovery is performed, so that the latter is actually done on an attacked object A'_w. As was previously cited, the attacks applied on a watermarked object could be either malicious (targeting watermarking security) or non-malicious (targeting robustness instead), although this separation is not always simple to do. Nevertheless, holding this overlapping of meanings in mind, it is possible to define non-malicious attacks as *robustness attacks* and malicious attacks as *security attacks*. In turn, malicious attacks could be further classified in *blind security attacks*, in which the attacker does not exploit any knowledge about the watermarking system but instead attacks the watermarked object with some operation which is not considered usual, hoping from the attacker's point of view that the system designer did not take them into account, and *non-blind security attacks*, in which the attacker knows all of the watermarking system, except the secret key employed, and exploits this knowledge to attack the system in its weak spots, e.g. by mounting special attacks allowed by the particular system implementation (such as the availability of watermark recovery tools and the ability to apply them to arbitrary objects) and/or exploiting knowledge about watermark localization or spectral properties to alter synchronization between the embedder and the recovery block or to filter out the watermark. These latter attacks are the

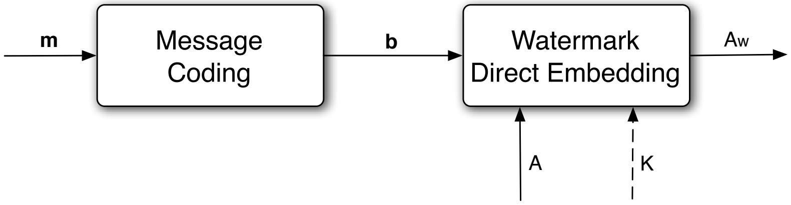


Figure 22.6: Direct watermark embedding steps.

most dangerous and at the same time they cannot be ignored by the system designer if the already cited Kerckhoffs' principle is to be respected; however, in some application, it is reasonable to safely ignore such attacks simply because the intentional watermark removal or disabling is against the attacker's own interest.

Robustness attacks are defined as all the transformations which are applied on the host object without aiming explicitly at disabling or removing the watermark; consequently, they comprise all processing which belong to the "normal" course of life of the host object. Which attacks to consider as non-malicious is a matter strongly dependent on the nature of the host object A; moreover, also the application intended for the digital watermarking system at hand plays an important role since many robustness attacks have more importance than others in certain contexts. There are so many types of robustness attacks that usually the system designer takes into account only a certain number of them (if any) and then hopes for the best for all the others.

Common signal processing, usually aimed at digital objects' perceptual content enhancement, are another form of robustness attacks. They could be some kind of simple features manipulation, e.g. the so-called constant gain attack (CGA) where the watermarked features are multiplied by a certain factor, or some more complex processing (for example for digital images histogram modification is a classic type of image processing similar to CGA); they could also be represented by filtering, either linear (e.g. low-pass filters) and non-linear (e.g. noise-suppressing filters).

Another very important class of robustness attacks are geometric manipulations, where geometry refers to any coordinate of the host object (spatial coordinates for images, temporal coordinate for audio and both spatial and temporal coordinates for video): for this reason they are also called synchronization attacks. These attacks could be very tricky as they tend to break the consistency between the coordinates of the embedder and those used during the recovery process. To undo synchronization attacks the system could either try to be as invariant as possible to coordinates modification (losing robustness with respect to other types of attacks) or leave the task of recovering the original coordinate configuration to the recovery block (usually by exhaustive search).

Finally, object editing processes, such as cropping for images, could be considered robustness attacks; they usually intrinsically contain some amount of geometric manipulations when they are applied.

Among blind security attacks, the exhaustive use of some robustness attack is the first thing that comes into mind, as the attacker may want to break the system by going beyond the robustness the watermark may tolerate.

Another way of attacking the system without delving into its details is to treat the watermark as noise and hence to use some noise-suppressing process to completely remove it in the simplest cases (in particular when the watermark is well modeled by an additive noise) or at times at least roughly estimate the watermark itself to remove it or to illicitly embed it into other objects (another example of how security and robustness concepts overlap). Other, more sophisticated threats should also be taken into account, e.g. the already cited collusion attack where many watermarked copies of the same host object or many objects with the same watermark embedded are compared to estimate the watermark.

Regarding non-blind security attacks, the first observation that has to be made is that the secret key K is the most sensitive parameter of the system, since its unwanted leakage to an attacker aware of the system design could signify the nullification of the watermark intended task. This is effectively a very tough problem in some applications where the recovery process is to be performed by any user willing to do so (and this means they have to use the secret key). This prompted the rise of asymmetric key schemes which use different keys during the embedding and recovery stages, although they introduce other problems (most notably a robustness decrease tendency).

Furthermore, when the attacker can repetitively perform the recovery process on any object, the so-called sensitivity attack could be adopted. In this attack, the attacker modifies little by little the watermarked object and then perform the watermark recovery; this will give a rather precise estimation of the detection/decoding boundaries, allowing to choose the most convenient unwatermarked object (e.g. the one that minimizes distortion), and maybe to learn some information about the secret key employed.

Using complex detection boundaries or making the attack construction unfeasible thanks to computational complexity issues are the most common countermeasures to the sensitivity attack.

Obviously, a non-blind attacker could also use one of the attacks previously classified as blind if it is identified as a weak point of the system after an appropriate analysis of the system operations. As a last note, to better counter security attacks on a critical application, it is arguably mandatory to couple watermarking and cryptography technologies on a protocol level. Several examples are found in security-oriented applications, where cryptography is usually used to secure content distribution and to limit as much as possible unauthorized access to the object at hand and watermarking is used to tie security information with the content perception. A good survey on security issues in digital watermarking can be found in [Cayre(2005a)] and [Cayre(2005b)].

3) Watermark recovery: The recovery stage is responsible for the extraction from the possibly attacked object A'_w of an estimate $\mathbf{m}'$ (the bit string representing m') of the original message string $\mathbf{m}$. Hence the most general point of view of the recovery process is described by the following equation:

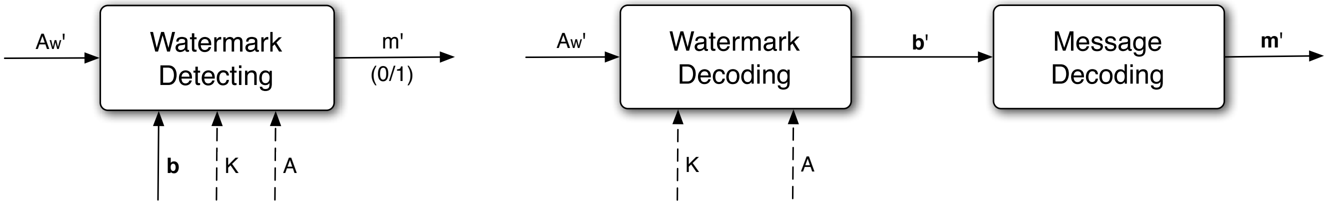
$$\mathbf{m}' = \mathcal{D}(A'_w[, K]) \tag{22.10}$$

A *recovery function* $\mathcal{D}(\cdot)$ is applied on A'_w to obtain the recovered message $\mathbf{m}'$. The form of the recovery function $\mathcal{D}(\cdot)$ pretty much depends on the nature of the watermarking algorithm, that is whether the recovery consists in assessing that a certain watermark is present or in choosing which watermark among those possible has been embedded. Therefore, Eq. (22.10) can be specialized into two different forms belonging respectively to *detectable* and *decodable* watermarking, that are separately depicted in Fig. 22.7.

In detectable watermarking, the watermark recovery process (now called the detector block) can be schematized as in Fig. 22.7a. Here we are only interested in assessing the presence or the absence of the watermark, so that the message m can be reduced to a binary variable (thus $\mathbf{m} = m$ as the string has unitary length); the fact that we are embedding a watermark inside a host object A means that $m =$"1". Consequently, the coding $\mathcal{C}(\cdot)$ of Eq. (22.5) is better described by a codeword selection (such as bit repetition rather than a channel code), as the watermark $\mathbf{b}$ is embedded into A only to mean that A_w is watermarked, regardless of the meaning of $\mathbf{b}$. The detector, then, looks for the watermark $\mathbf{b}$ into the attacked object A'_w and takes a decision about its presence or not, thus outputting an estimated message m' which is 1 in case the detector believes the watermark $\mathbf{b}$ has been embedded into A'_w and 0 otherwise. Eq. (22.3) exemplifies this process: a threshold selected on the similarity measure between the supposedly embedded watermark and the received one embodies the detection decisor. Therefore, the watermark detection could be expressed as:

$$\mathcal{D}(\mathbf{b}, A'_w[, A, K]) = m' \in \{1, 0\} \tag{22.11}$$

Notice how in this case the detector must know in advance $\mathbf{b}$ so it could achieve its task. In some literature, it is stated that since the message m conveys only 1 bit of information (the watermark presence or its absence), thus detectable schemes are also defined as 1-bit watermarking. This could be confusing: it would be more correct to say that the *detector* obtains 1 bit of information, but it has to be kept in mind that the possible message m is only one.



(a) Watermark detector.

(b) Watermark decoder.

Figure 22.7: Watermark recovery process forms.

On the contrary, decodable watermarking is depicted in Fig. 22.7b. In this case, the recovery block, conveniently called the decoder, does not know in advance the watermark $\mathbf{b}$ so that it has to read it from the attacked object A'_w (for this reason, this scheme is also called readable watermarking) to form an estimated watermark $\mathbf{b}'$. Here the original message m is meaningfully represented by the string $\mathbf{m}$, and the message coding $\mathbf{b} = \mathcal{C}(\mathbf{m})$, if present, could be for example a channel coding of the message $\mathbf{m}$ into the watermark $\mathbf{b}$. The first operation of the decoder is to decode an estimated watermark $\mathbf{b}'$ from the attacked object A'_w (using a function which is referred as $\mathcal{D}'(\cdot)$); then, given that the first stage of the embedding process is the message coding, the last stage of the decoding process is obviously the message decoding of the recovered string $\mathbf{m}' = \mathcal{C}^{-1}(\mathbf{b})$. As is easily imagined, decodable watermarking is also called multi-bit watermarking. Fig. 22.3 is an example of multi-bit watermarking. Depending on the symbol over which the received feature point is quantized, a different pair of bits is decoded (hopefully, if we embedded the "square" as depicted by $\overline{h}$, the received feature point will still be nearer to it than to the other symbol to ensure correct decoding). The whole process is illustrated as follows:

$$\mathcal{D}'(A'_w[, A, K]) = \mathbf{b}' \tag{22.12a}$$

$$\mathbf{m}' = \mathcal{C}^{-1}(\mathbf{b}) \tag{22.12b}$$

Looking at Fig. 22.7, one may notice that the host object A plays the role of an optional input to the recovery process. As it turns out, in some applications the recovery process has access to the original host object A (perhaps because the embedding and the recovery are performed by the same entity), so that in Fig. 22.7 one may add the original host object A as an additional input to the detector/decoder block. If this is the case, the watermarking scheme (or equivalently the recovery process) is said to be *non-blind* (and the optional input A in Eq. 22.11 and Eq. 22.12 has to be considered); otherwise, if the recovery block does not have any knowledge about the host object A, it is called *blind*. The spread-spectrum watermarking technique described above and in [Cox(1997)] is an example of non-blind watermarking, since the original unwatermarked image is needed during the recovery phase. On the other hand, the QIM paradigm as illustrated in [Chen(2001)] is based on a blind watermark recovery stage.

Using realistic assumptions on the system, non-blind systems surely achieve better robustness over their blind counterpart, but one must keep in mind that such a framework is not always applicable in real-world applications; moreover, the performance gap is not as large as one could intuitively expect.

To summarize, the recovery function can assume one of the forms listed in Table 22.1.

D. Watermarking system evaluation

Once a watermarking system is implemented, its performance should be evaluated as objectively as possible [Cox(2008), Chapter 7]. Although additional parameters can play a role, such as complexity or memory usage, here we will focus on the satisfaction of the main requirements that we discussed earlier.

Imperceptibility of the watermarked image refers to its indistinguishability from the original, unwatermarked image and it can be evaluated either subjectively or objectively. The former usually involves user tests performed in controlled environments, but they are seldom used in watermarking contexts. The latter relies on the computation of objective metrics which may or may not be driven by HVS models. In almost all the works subject of this Chapter, imperceptibility is either evaluated through PSNR, that, although providing a rough estimate of the embedding distortion, is not very well correlated with actual human perception, or more sophisticated metrics for HDR data such as HDR-VDP [Mantiuk(2005)] and its successor HDR-VDP-2 [Mantiuk(2011)] (see Chapter 17 of this book).

	Blind recovery	Non-blind recovery
Detectable watermarking	$\mathcal{D}(\mathbf{b}, A'_w[, K]) = 1/0$	$\mathcal{D}(\mathbf{b}, A'_w, A[, K]) = 1/0$
Decodable watermarking	$\mathcal{D}(A'_w[, K]) = \mathbf{m}'$	$\mathcal{D}(A'_w, A[, K]) = \mathbf{m}'$

Table 22.1: Watermark recovery function forms.

Evaluating security is a much more challenging issue and is outright ignored in many works, even in security-critical applications such as copyright protection. In most cases, the secret key is only used as the seed of a pseudorandom number generator responsible of constructing the watermark sequence. In this sense, only recipients with the correct secret keys can read the watermark, however all the other security aspects are not covered by this approach. For example, an intelligent attacker can be interested in disabling the watermark recovery by simply trying to delete the watermark, e.g. by embedding a spurious watermark, which is always possible if one assumes that he/she knows the details of the watermarking system and the location of the watermark into the embedding domain is invariant. Although theoretical analysis can be carried out when the problem is simplified, most of the time evaluating the security of a watermarking system is done empirically.

Last, evaluating robustness is related to the fact of how the watermark is recovered. In every system, in some part of it a threshold (or more than one) needs to be chosen to differentiate between watermarked and unwatermarked images, or to decode bits of the embedded message. The threshold should be selected by minimizing a loss average, where the loss is a function describing the damage caused if an error occurs in the decision process. In particular, for decodable watermarking, the bit error rate (BER) of the recovered watermark is usually considered. In the case of detectable watermarking, if the system believes that the watermark is present even when it is not the case, we are in presence of a *false alarm*; conversely, if the system wrongly believes that the watermark is absent a *miss* has occurred. The goal of the designer is to estimate the probability of false alarm and the probability of miss of the watermarking system given any host object and any other input (e.g. any secret key). Watermarking systems are usually designed according to the Neyman-Pearson criterion [Neyman(1933)], where the threshold is selected such that the false alarm probability is less than a given target figure and then evaluating the resulting miss probability. Both probabilities are then depicted on a ROC (Receiver Operating Characteristic), usually obtained by letting the false alarm probability vary, and then calculating the correspondent threshold (the order is inverted when experiments correspond to practical rather than theoretical evaluations) and finally observing the miss probability. Sometimes, the equal error rate (EER), which is the point in the ROC where miss probability and false alarm probability are equal, is provided. An example of ROC will be shown in the next Section.

22.2. Digital watermarking for HDR images

As HDR images are gaining ground in day-to-day applications, early efforts into more sophisticated requirements for applications have been pioneered. Digital watermarking has carved itself a niche in security-oriented applications but its deployment could represent a challenging problem, so it should not be surprising that at the time of this writing only a handful of works have been published in the literature on the subject. In what follows, we will give a brief description of these works, following an approximate temporal order, that includes their classification according to the criteria described earlier to give a better idea on how they relate to each other. Before that, we will discuss in general terms what are the requirements for HDR image watermarking.

A. Requirements for HDR image watermarking

In the case of HDR images, some peculiarities exist with respect to the classical LDR images. These should be taken into account when designing a watermarking system. For this reason, it is generally impossible to directly port an established LDR image watermarking technique into the HDR domain since both imperceptibility and robustness would suffer greatly. It is in fact obvious that a small modification in the LDR domain could become huge when ported back in the HDR since the range of the pixel values there is far more extended with respect to the usual 8-bit, [0, 255] range. Also, the pixel value itself may be redundant, that is, by suitably modifying the exponent and mantissa part of the floating point representation, sometimes there are more than one way to express a given pixel value.

The pixels' value range difference in the two domains also has far-reaching implications into how humans perceive HDR images, so that perceptual models should be corrected when dealing with them. That generally implies tuning the specific parameters of the usual perceptual masks, considering how sharper and richer the visual representation of high dynamic range images is with respect to LDR images. For example, the contrast masking effect of the human visual system is surely higher in the HDR domain given its richness in details.

One way to tackle these issues is to first transform the HDR image into a LDR one, watermark it using a LDR image watermarking process and then revert back to the HDR domain by some inverse transformation, as many systems do. In doing so, particular caution should be exercised to guarantee that the modifications in the LDR domain would not be perceptible in the HDR domain. An example of such reasoning can be found in the following when we describe Fig. 22.8.

Moreover, it is worth noting that the high visual value of HDR images is always put at a premium. Therefore, when considering which processing high dynamic images can (or should) undergo, only those that do not severely alter their perceptual value should be studied. As a matter of fact, most systems try to be robust against a single type of manipulation: tone-mapping operators. They are common nowadays to permit the rendering of HDR images on LDR displays, so naturally they are considered common signal processing in HDR imaging. Therefore, watermarking systems should be robust against tone-mapping and possibly permit watermark recovery in both the original HDR image domain and the low dynamic range of its tone-mapped versions, whatever specific operator has been employed.

More comments on the applicability of digital watermarking in the HDR domain can be found in the last Section, where we draw some remarks based on the current state of the art that we review next.

B. *Survey of the current state of the art*

The work in [Guerrini(2008)], further expanded in [Guerrini(2011)], is to the best of our knowledge the first published paper on data hiding for HDR images. It is a blind detectable watermarking method, so its capacity is a single bit. The capacity is sacrificed to favor the other requirements: robustness against tone-mapping and noise addition, security and imperceptibility.

The rationale under this watermarking scheme is depicted in Fig. 22.8. Tone-mapping operators are well modeled through a logLUV process (L), so the HDR image is first transformed into the logLUV domain and then a LDR watermarking robust against non-linear attacks and invariant to constant gain modifications is applied to the luminance component only. The HDR watermarked image is obtained by exponentiation L^{-1}. It is argued that each tone-mapping operator (TM^i) is only a mild non-linear transformation away from the logLUV image, so in the end $I_{TM^i}^W$, the tone-mapped version of the HDR watermarked image, still retains the watermark.

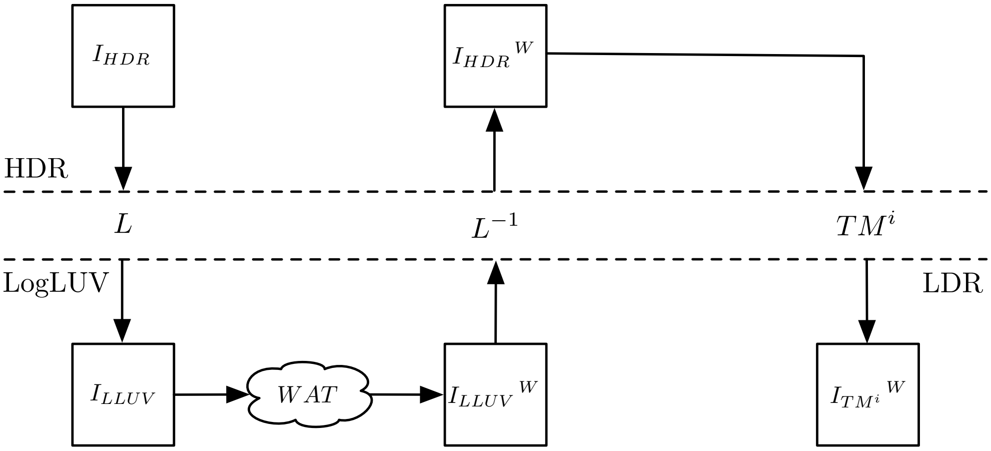


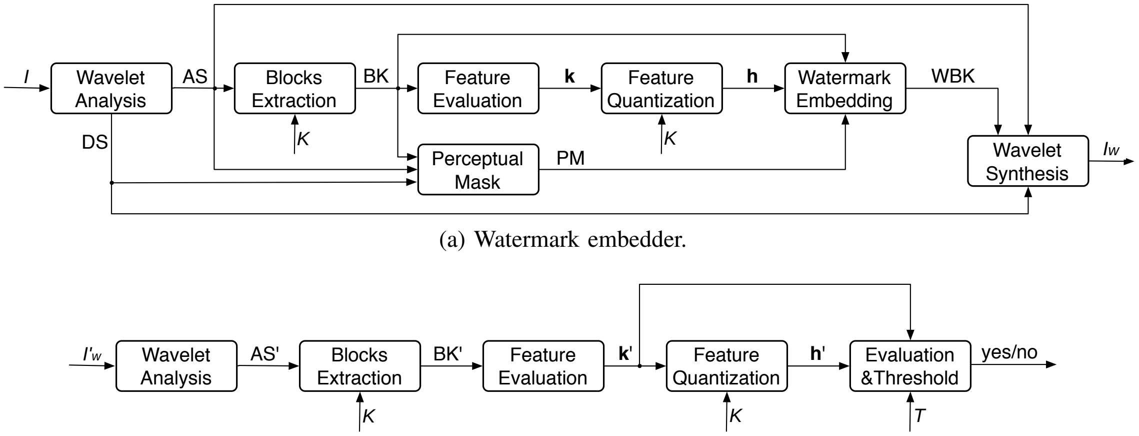
Figure 22.8: Conceptual representation of the framework in which the watermarking system of [Guerrini(2011)] is expected to operate.

The embedding method is quite complicated: the flowchart is reported in Fig. 22.9a. It is based on the Quantization Index Modulation paradigm previously explained, which in this case encodes information into the shift of a non-uniform quantizer. The quantization is applied to the kurtosis of the approximation coefficients resulting from a wavelet decomposition, taken into randomly positioned, randomly shaped blocks. Imperceptibility is aided by using a HVS-derived perceptual mask, computed using the detail subbands as well (that are left untouched by the watermarking process), taking into account brightness, neighborhood activity and presence of edges. Security is very high, since it relies on both the shift of the quantizer and the random position and shape of the blocks on which the kurtosis feature is computed, with the attacker needing to employ an attack at least visually perceptible as the mask to disable the watermark. Watermark recovery, illustrated in Fig. 22.9b, is relatively straightforward. The secret key drives the blocks extraction and the computed kurtosis feature in each of them is quantized using the same codebook employed in the embedding phase. If the number of blocks correctly decoded is higher than a threshold T, the image is judged as watermarked.

The experiments have been performed on 15 HDR images and using 7 tone-mapping operators. The imperceptibility is measured using HDR-VDP, the average of which is below 0.5%, signifying that the watermark is almost impercetible. Fixing the false alarm probability at 10^{-5}, the miss probability can be as high as 10^{-2} for small images but goes well below 10^{-6} for larger images, which is the most common case for HDR imagery. One of the obtained ROCs can be found in Fig. 22.10, for an image watermarked using $N = 700$ blocks (which is reasonable for a medium sized HDR image). As is usual for ROCs, the axis are usually drawn in logarithmic scale and the curve in the cases of various robustness attacks (tone-mapping operators in this case) is obtained varying a detection threshold. It is worth noting that no non-realistic assumptions are made on the distribution of errors; instead, a binomial distribution based on the actual block decoding error probability is assumed. Fig. 22.10 also suggests how hugely different can be the scale of introduced distortion for different tone-mapping processes.

The work in [Guerrini(2008)] went mostly unnoticed until the appearance of [Guerrini(2011)]. Meanwhile, other works appeared which were actually targeted towards steganography, so caution should be adopted when evaluating them for watermarking purposes. Nevertheless, it's interesting to report here those efforts that, exploiting the characteristics of HDR images to hide data, have inspired later works on HDR image watermarking.

The work in [Cheng(2009)] is the first that tackled the subject of steganography for HDR imagery. As it is costumary for steganography, the emphasis is put on capacity and imperceptibility at the expense of robustness. In this case, no manipulation is even expected between the message embedding and its subsequent recovery, so robustness has not been tested. The method is based on the so-called LSB embedding, in which the least significant bit or bits carry the embedded message. LSB embedding can be considered as a waveform-based, additive scheme dependent on the original image pixel values and



(a) Watermark embedder.

(b) Watermark detector.

Figure 22.9: Watermark system flowchart for [Guerrini(2011)], embedding and recovery parts respectively.

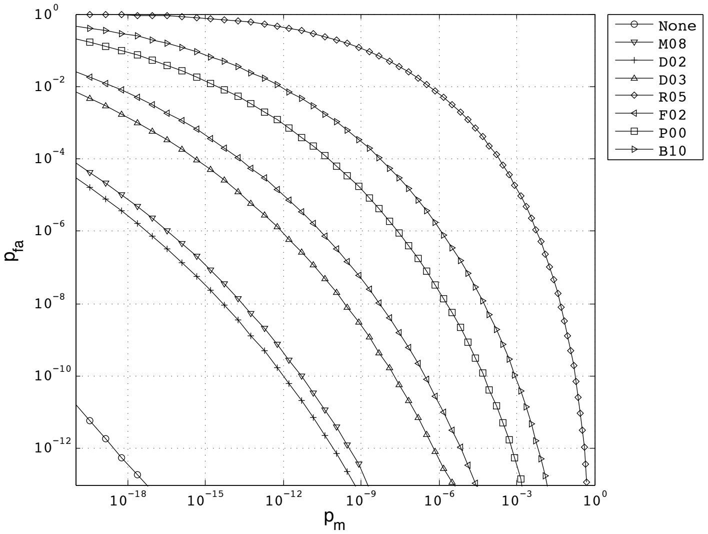


Figure 22.10: Example of a ROC. This is obtained using the method in [Guerrini(2011)] setting the number of blocks $N = 700$. Each curve represents a different tone-mapping operator (see [Guerrini(2011)] for further details).

is a popular choice in steganographic applications. Early watermarking techniques also proposed LSB embedding until robustness and security issues mostly barred further efforts in this direction. The embedded message recovery is blind, another common feature of steganography.

Imperceptibility is pursued through some heuristics: no explicit HVS model is used. First, 32-bit, RGBe format pixels [Reinhard(2005)] are classified into flat and non-flat image classes and different embedding methods are used for each. This classification is performed by comparing the exponent terms in neighboring pixels. Then, the number of least significant bits that are modified to carry the message is computed in a way to be higher for dark and contrast image areas and smaller for bright and smooth areas. Also, RGB channels are weighted to reflect the higher sensitivity of the human eye to the red and green channels.

Security requirements are basic in this case. The (symmetric) secret key is used to encrypt and decrypt the plain text message and a form of authentication through message digesting (e.g. MD5) is also advised and implemented.

Given that this work is not concerned with robustness, the experiments focus on imperceptibility using PSNR as metric and capacity. The tests are performed on 7 standard HDR images and report values going from 1 to 3 bits per color channel per pixel (with total capacity per image in the Mbit range) and PSNR above 30 dB.

The work in [Li(2011)] is inspired by [Cheng(2009)] and therefore embraces most of its premises. For instance, the suggested application is still steganography and, as such, in this work too, robustness is not taken into account. Security concerns are neglected as well, but one can assume that the same kind of basic requirements can be employed for this scheme. Of course, if the same type of message digesting security as in [Cheng(2009)] is to be included, the net capacity is bound to decrease through the inclusion of security information in the embedded bits.

Again, no HVS model is considered. Instead, a simple assumption is made: to minimize the embedded information perceptibility the total variation of each pixel's value should be minimized. The HDR format considered is logLUV TIFF [Reinhard(2005)], where the luminance value is floating point, and each image is first normalized to a given set of luminance exponent values before embedding to cope with variable HDR ranges across different images.

The embedding strategy is once again based on the LSB paradigm. The information is embedded in the mantissa for luminance values and the exponent is then selected to minimize the difference between the new luminance value and the original one. The chrominance channels have a direct value representation, hence classical LSB is employed there. The best trade-off between capacity and imperceptibility is reported as 6 bits embedding for the luminance channel and 10 bits for each chrominance channel.

The experiments once again only consider capacity and imperceptibility, using a testbed of 10 HDR images. Moreover, the PSNR is still chiefly used to measure imperceptibility. The authors of [Li(2011)] report an increase in capacity (more than doubled) and PSNR (up 2 to 3 dB) with respect to [Cheng(2009)]. However, the set of test images (10 in this case) is different, so these conclusions should be taken with caution. For a single image, HDR-VDP$_{75}$ and HDR-VDP$_{95}$ are also computed and both reported under 1%.

The work in [Yu(2011)] is again on HDR images steganography, but it considers a different set of requirements. Robustness is neglected in this work as well, and basic security is achieved by simple pseudo-random scrambling of the pixels order. The same assumptions as above about increasing security at the expense of net capacity still apply. Furthermore, the capacity could still be decreased when the presence of an intelligent attacker trying to detect the presence of a message is assumed because only a subset of the available pixels is used (see below). The embedded message recovery is blind.

With respect to the previously described methods, imperceptibility in this case is almost maximized, but the capacity is much lower. This method exploits the redundancy in the RGBe representation format for a given pixel, that is the fact that adding 1 to the exponent and halving the other channels (or subtracting 1 to the exponent and doubling the other channels), with some obvious extra assumptions on over- and underflowing and rounding, does not change the pixel value. Embedding information into pixels, therefore, consists in choosing one of these equivalent representations, once they are sorted using the exponent value, using the embedded bits as index. Obviously, not all pixels admit equivalent representations and the number of embeddable bits depends on the number of these representations.

The experiments are only concerned with capacity, as imperceptibility is all but guaranteed by this technique. Depending on the intended application the authors of [Yu(2011)] reported two different average capacity. In the less security critical environment of image annotation, it is in the 0.1 bpp range. When an intelligent attacker is assumed, the message is embedded only in a random, secret key driven subset of the available pixels. In particular, the selected pixels are such that altering them do not alter the statistics of the image with respect to the original image. In this case, the capacity is in the 0.001 bpp range.

The work in [Wang(2012)] directly builds upon that in [Yu(2011)]. The proposed variation is to group pixels for embedding purposes into so-called segments, instead of considering pixels one at a time, and it also uses a more sophisticated approach to scramble the pixel order to construct the segments. This mechanism makes a more effective use of the number of equivalent representations for groups of pixels and improves the capacity by just around 5%, so there is probably not much more room for improvement on the capacity side.

Following the above papers, other watermarking methods started to appear in 2011. As already stated, the most common requirement is robustness against tone-mapping operators, while the other requirements are variably addressed.

The work in [Xue(2011)] discusses two approaches to implement a blind detectable (1-bit capacity) watermarking scheme. Both techniques aim at robustness against tone-mapping and do not consider security. Also, imperceptibility is not addressed explicitly but relies on the watermark embedding domain to ensure it is achieved. Both proposed techniques are based on the multiplicative spread-spectrum watermarking paradigm. Hence, correlation-based blind recovery is employed in both, with a basic modicum of security provided by the secret key acting as the seed of the pseudorandom watermark.

The first technique approximates the tone-mapping process with a μ-law function applied on the HDR image, obtaining a LDR image and a ratio image (the original HDR image divided by its tone-mapped version). Then, the LDR image is wavelet decomposed and the watermark is embedded into the vertical and horizontal detail subbands. Last, the watermarked HDR image is obtained by multiplying the wavelet reconstructed watermarked LDR image by the ratio image.

The second technique applies bilateral filtering to the HDR image to obtain a large scale part and subsequently a detail part by subtracting the large scale part to the HDR pixel values, after having taken the logarithm of both. Then, the detail part undergoes the same processing described above: wavelet decomposition and reconstruction with the spread-spectrum watermark embedding in between. Finally, the watermarked detail part is summed to the large scale part and, recalling we have applied the logarithm, the sum is exponentiated to obtain the watermarked HDR image.

The experiments conducted on 5 HDR images first address imperceptibility, giving PSNR figures in excess of 50dB for the first technique and lower figures (32dB in one image) for the second technique. Not surprisingly, the second technique is more robust against tone-mapping according to a limited set of experiments, although some correlation values appear to be under the threshold hence resulting in missed detection.

The work in [Wu(2012)] is a blind decodable watermarking technique that aims to embed a 4800 bits logo in HDR images. Aside from tone-mapping operators, robustness against noise addition, cropping and blurring is sought after, while security is completely neglected.

To be robust against tone-mapping, a prototype tone-mapping operator is first applied to the HDR image and the watermark embedding is performed on the resulting LDR image. The HDR watermarked image is obtained again by storing a ratio image by which to multiply the watermarked LDR image. The LDR watermarking method is a variation of the classical spread-spectrum method in the DCT domain, where instead of directly embedding the watermark with the additive rule of Eq. (22.1) on the DCT coefficients, the value difference between specific pairs of medium-to-low frequency coefficients is modified. The watermark blind recovery is based on the computation of a correlation coefficient.

The experiments are performed on a single HDR image at a time. Imperceptibility is evaluated through PSNR, which is reported around 70 dB. Robustness against attacks is given by reporting the correlation coefficient, that on average goes from 0.5 to 0.8 (with bit error rates approximately ranging from 12% to 23%), giving mostly human-readable recovered logos since the HVS tends to compensate for errors in logo images with those BERs.

The work in [Solachidis(2013a)] is directly derived in the HDR domain. A just noticeable difference (JND) mask is obtained from the HDR image, and then it is used to embed 128 bits. Again, the target is robustness to tone-mapping operators, and a minimum of security is obtained by using a secret key to scramble the data.

The imperceptibility is pursued by using a mask based on a contrast sensitivity function, on bilateral filtering and on the aforementioned JND. The embedding method is of the multiplicative spread-spectrum type, applied in the wavelet domain, and the blind recovery is based on a threshold. The mask is used to temperate the embedded information depending on the mask value. Only the luminance channel is used to embed the watermark in this work as well.

The experiments are performed on 3 HDR images and used a set of 7 tone-mapping operators. The watermark decoding can be performed on both HDR images and LDR images obtained through tone-mapping. The reported bit error rate is, in the worst case, around 5%. Miss and false alarm probabilities are then extrapolated assuming Gaussian distributions for the bit errors, and almost negligible miss probabilities for false alarm probabilities equal to 10^{-10} are reported. No experimental evaluation on imperceptibility is reported.

The same authors of [Solachidis(2013a)] proposed two other techniques. The first, that is proposed in [Solachidis(2013b)], is a blind detectable watermarking scheme (so the capacity is 1 bit) with again robustness against tone-mapping as the main target.

Here, the original HDR image is decomposed into a set of LDR images, each representing a subset of the original dynamic range. On each LDR image, a LDR image watermarking already proposed in the literature is employed and the watermarked HDR image is then obtained by combining the set of LDR watermarked images. The LDR watermarking scheme belongs to the additive spread-spectrum family, it works in the wavelet domain and it uses a HVS-based mask to achieve imperceptibility. Security resides in the pseudorandom sequence to be added to the original wavelet coefficients, so the watermark cannot be read but it can be disabled by an intelligent attacker. Collusion attacks are also not considered. The recovery can be performed on both the HDR image directly or on a LDR image obtained by scaling the dynamic of the HDR image.

The experimental results are performed on 6 HDR images. The strength of the watermark is adjusted so as to have a HDR-VDP-2 under 5% for 90% of the image pixels, which is assumed as a good value for imperceptibility. The miss and false alarm probabilities are very small, when assuming Gaussian distributions for the detection scores. The tests considered 5 different tone-mapping operators.

The second work [Solachidis(2013c)] is a blind decodable watermarking method. In this case, the employed method is completely different, although it also aims at being robust against tone-mapping operators.

The authors propose to consider one of the first-level wavelet subbands of the logarithm of the luminance (logL) component of the HDR image, separating it into blocks and then applying the Radon-DCT transform [Do(2003)], leaving the chrominance channels unmodified. Then, a QIM watermarking process is applied, hence classifying this method into the direct embedding family. The features to be quantized are the most energetic directions, so as to embed the information into the edges of the image to ensure maximum imperceptibility. The secret key controls the quantizer shift and is therefore the only security mechanism present. The capacity ranges in the tens of kbits and depends on the number of blocks present in the image. In this case as well, the watermark blind recovery can be done on both the watermarked HDR image or on a tone-mapped LDR image.

Experiments have been carried on 7 images and 5 different tone-mapping operators are considered. The imperceptibility is at its best when using the HH subband, where the HDR-VDP-2 gives a 5% probability of detecting a modification on around 5% of the image. However, on the average for the HH subband, the bit error rate is as high as 22%.

The work in [Autrusseau(2013)] is another work focusing on robustness against tone-mapping. It is a detectable watermarking technique, so its capacity is 1 bit. Imperceptibility relies once again on the use of wavelet decomposition.

A non-linear variant of the classical multiplicative spread-spectrum watermarking paradigm is employed on all the first-level subbands of a wavelet decomposition. Thus, security only consists of the seeding for pseudorandom noise, which is the watermark. The blind watermark recovery is, as usual in this watermarking methods, based on correlation between the extracted and the original watermark.

The experimental results on robustness, considered for 8 HDR images and 6 tone-mapping operators, do not give any actual figures, but they report that false detection occurs more frequently than true detection for at least some combinations of original HDR image and tone-mapping operator. The imperceptibility is in this case as well evaluated through HDR-VDP and is reported around 95% (except for a single image with 85%).

C. Summary of HDR image watermarking systems

In Table 22.2 we have reported a summary of the proposed HDR image watermarking systems described above. The point of the table is not to allow experimental comparisons (see next Section). Instead, here we just want to list the various approaches and point out a few characteristics that can be inferred by comparing them.

The first observation is how uneven the proposed embedding schemes are, including purely steganographic systems. In addition, some systems are of the detectable kind and others embed a variable quantity of information in the host image. This is not necessarily a bad thing, but indicates how a clear favorite application for HDR image watermarking (and a suitable requirements mix for it) has not emerged yet.

The second observation is that, besides a single work, security is mostly neglected and relies on simple strategies, like using a secret key to construct the watermark sequence. While this prevents the unauthorized decoding or detection of the watermark, it does not guarantee that the watermark recovery cannot be impaired by more sophisticated attacks aimed directly at the recovery process. The latter fact in general forbids the deployment of the watermarking scheme in security-critical applications such as the ones dealing with copyright protection.

Another couple of observations can be made with respect to how imperceptibility is handled. First, there is a discrepancy in the use of metrics to assess the perceptual distortion introduced by the watermark

Reference	Embedding Domain and Algorithm	Recovery	Watermark requirements			
			Capacity	Robustness	Security	Imperceptibility
Guerrini(2008), Guerrini(2011)	QIM (direct embedding) on the kurtosis of AS coefficients in logLUV domain (2-level wavelet decomposition)	Blind	1 bit (detectable)	7 TMOs, Gaussian noise (masked). Evaluated with ROC	Features computed in blocks of random shape and in random locations, random quantization shift	Use of a perceptual mask, based on brightness, activity and edges. Experiments with HDR-VDP
Cheng(2009)	LSB embedding in HDR domain, RGBe format	Blind	3-9 bpp	None (steganographic)	In the clear, MD5 for message authentication	Experiments with PSNR
Li(2011)	LSB embedding in HDR domain, logLUV TIFF format	Blind	26 bpp	None (steganographic)	In the clear	Experiments with HDR-VDP
Yu(2011), Wang(2012)	Using redundant representation of HDR pixel values in RGBe format	Blind	0.001 bpp	None (steganographic)	Pixel scrambling	Completely imperceptible by design
Xue(2011)	Bilateral Filtering of HDR image, multiplicative spread-spectrum in wavelet domain	Blind	1 bit (detectable)	4 TMOs. Evaluated with detection scores	Pseudorandom watermark	Experiments with PSNR
Wu(2012)	LDR domain (after TMO), variation of additive spread-spectrum watermarking applied to medium-low frequency DCT coefficients	Blind	4800 bit	4 TMOs, noise, blurring and cropping. Evaluated with BER	None	Experiments with PSNR
Solachidis(2013a)	HDR domain, multiplicative spread-spectrum applied to wavelet coefficients	Blind	128 bit	7 TMOs. Evaluated with BER	Pseudorandom watermark	Use of a perceptual mask, based on JND, contrast and bilateral filtering. No assessment
Solachidis(2013b)	LDR domain, obtained through bracket decomposition, additive spread-spectrum watermarking applied to wavelet coefficients	Blind	1 bit (detectable)	5 TMOs. Evaluated with EER obtained assuming Gaussian error distributions	Pseudorandom watermark	Experiments with HDR-VDP-2
Solachidis(2013c)	QIM (direct embedding) of most energetic Radon-DCT directions, applied on the logL LDR domain	Blind	Tens of kbit	5 TMOs. Evaluated with BER	Quantizer shift	Experiments with HDR-VDP-2
Autrusseau(2013)	Wavelet transform of HDR pixel values, non-linear variant of multiplicative spread-spectrum	Blind	1 bit (detectable)	6 TMOs. Evaluated with BER	Pseudorandom watermark	Experiments with HDR-VDP

Table 22.2: Summary table of the current state of the art in HDR image watermarking.

embedding process. Some of the works use the PSNR, which is probably unsuitable when applied to the high quality content at hand: the breadth of work dedicated to the development of suitable perceptual metrics for HDR image quality is a clear proof of this fact (see Chapter 17). Also, it is worth noting that no papers have even considered the possibility of implementing visible watermarking, but all aim to render the watermark imperceptible to retain the image content quality. In addition, most works just rely on the watermark domain (e.g. using the detail subbands of a wavelet decomposition) to achieve imperceptibility. Perceptual masks, possibly developed ad-hoc for the HDR domain, can possibly provide a needed upgrade to guarantee that the watermarked image retains high visual quality.

There are some common points as well that arise from Table 22.2. For example, watermark recovery is always blind, a scenario justified for those applications in which it is assumed that the entities performing watermark embedding and recovery differ. Also, all the proposed techniques that aim to achieve robustness (that is, excluding the steganographic systems), recognize the importance of being robust against the tone-mapping operators which constitute the most widely diffused form of processing the HDR images are currently expected to undergo.

22.3. Concluding Remarks

In this Chapter we briefly introduced the data hiding branch known as digital watermarking, with a particular focus on the still image case. Turning our attention to the HDR domain, we wanted to give a high-level overview of the current state of the art. To conclude our discussion, in this final Section we offer some remarks on the present status of digital watermarking applied to HDR.

First, as can be inferred from our discussion in the preceding Section, referring in particular to Table 22.2, we avoided making any comparison between the described HDR image watermarking techniques. We did this purposefully: we want to highlight here how difficult is to properly conduct critical evaluations of the current state of the art for a number of reasons. First, the sheer number of available works is still too low to draw conclusions, suggesting that much more work is needed in the field. Second, there is a distinct lack of recognized "standard" HDR image databases, and the few readily available consist in a small number of images. In both these aspects, it is easy to conclude that HDR image watermarking is completely out of pace with respect to the huge amount of research poured into the LDR image field. Instrumental to this detachment is also what we have already previously pointed out: that it is in general not possible to directly transpose a LDR image watermarking into the HDR domain.

These are not the only problems preventing a critical comparison between available works. More importantly, there is a feeling that no research group agrees on the requirements a HDR image watermarking should satisfy. This latter fact is probably dictated by the complete absence of any deployment of actual watermarking system in real-world scenarios. Once the need to explore watermarking in a real application will arise, it is likely that all these problems will be dealt with simultaneously and comparisons between proposed techniques shall be made possible.

Speaking of watermarking requirements, looking again at Table 22.2, it is somewhat surprising that all the proposed technique are based on the blind recovery paradigm. It is pretty easy to imagine applications catering to the perceptual quality of HDR content that might make use of a non-blind recovery stage, considering the advantages in terms of robustness achievable by such framework. For example, some company could sell their HDR images after having properly and imperceptibly watermarked them. Then, an interested party might want to verify the authenticity of the content of a tone-mapped version of such image. To do that, it could upload that LDR image to a site controlled by the selling party, that would then extract the watermark using a non-blind watermark recovery system since it possesses the unwatermarked original image. The only security requirement in such an application would be the impossibility to completely remove the watermark to prevent misappropriation - simply disabling its recovery would only damage the image quality and prevent its authentication. This application is perfectly viable, so we suggest that more effort should be poured into non-blind watermarking. We also want to point out that, with these assumptions in place, steganography could still arise as the killer application, and that only time will tell.

In addition, as one can observe from Table 22.2, imagining critical security free applications such as the one described above also matches the poor attention that security has enjoyed until now in the literature. Techniques with tighter security can of course be also proposed, for example letting the buying party in the scenario above perform the authentication itself. Such applications would likely require additional security infrastructure besides the one provided by the watermarking system, e.g. enlisting the aid of an asymmetric key cryptography framework: for example, a public key could be needed to recover the watermark while a private key could be necessary to embed it.

As a quick note, robustness against tone-mapping operators seems to be the present focus of the proposed methods. However, it is easy to predict that HDR image watermarking will have to be robust against other processing as well: transcoding, as HDR image coding solidifies itself, springs to mind.

As a last note, a major absent from this Chapter is the HDR video medium. In fact, to the best of our knowledge, no one has proposed a HDR video watermarking technique yet. However, given the infancy that even the still image field is still in at the present time, that fact should be hardly surprising. We are pretty certain that, soon after the still image case reaches an adequate level of maturity, works concerning HDR video watermarking would begin to appear.

References

[Autrusseau(2013)] F. Autrusseau, and D. Goudia, *”Non Linear Hybrid Watermarking for High Dynamic Range Images”*, Int. Conf. on Image Processing, pp. 4527-4531, 2013.

[Barni(2004)] M. Barni, and F. Bartolini, *”Watermarking Systems Engineering”*, Marcel Dekker Inc., 2004.

[Cayre(2005a)] F. Cayre, C. Fontaine, and T. Furon, *”Watermarking Security Part One: Theory”*, Proc. SPIE Security and Watermarking of Multimedia Contents VII, Vol. 5681, pp. 746-757, 2005.

[Cayre(2005b)] F. Cayre, C. Fontaine, and T. Furon, *”Watermarking Security Part Two: Practice”*, Proc. SPIE Security and Watermarking of Multimedia Contents VII, Vol. 5681, pp. 758-768, 2005.

[Chen(2001)] B. Chen, and G. W. Wornell, *”Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding”*, IEEE Trans. on Information Theory, Vol. 47, No. 4, pp. 1423-1443, 2001.

[Cheng(2009)] Y.-M. Cheng, and C.-M. Wang, *”A Novel Approach to Steganography in High Dynamic Range Images”*, IEEE Multimedia, Vol. 16, No. 3, pp. 70-80, 2009.

[Cox(1997)] I. J. Cox, J. Kilian, F. Leighton, and T. Shamoon, *”Secure Spread Spectrum Watermarking for Multimedia”*, IEEE Trans. on Image Processing, Vol. 6, No. 12, pp. 1673-1687, 1997.

[Cox(2008)] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *”Digital Watermarking and Steganography, 2nd ed.”*, Morgan Kaufmann Publishers Inc., 2008.

[Do(2003)] M. N. Do, and M. Vetterli, *”The Finite Ridgelet Transform for Image Representation”*, IEEE Trans. on Image Processing, Vol. 12, No. 1, pp. 16-28, 2003.

[Guerrini(2008)] F. Guerrini, M. Okuda, N. Adami, and R. Leonardi, *”High Dynamic Range Image Watermarking”*, Proc. of the Int. Conf. on Circuits/Systems, Computers and Communications, pp. 949-952, 2008.

[Guerrini(2011)] F. Guerrini, M. Okuda, N. Adami, and R. Leonardi, *”High Dynamic Range Image Watermarking Robust Against Tone-Mapping Operators”*, IEEE Trans. on Information Forensics and Security, Vol. 6, No. 2, pp. 283-295, 2011.

[Kerchoffs(1883)] A. Kerchoffs, *”La cryptographie militaire”*, Journal des Sciences Militaires, Vol. IX, pp. 5-38, 1883.

[Li(2011)] M.-T. Li, N.-C. Huang, and C.-M. Wang, *”A Data Hiding Scheme for HDR Images”*, Int. Journal of Innovative Computing Information and Control, Vol, 7, No. 5A, pp. 2021-2035, 2011.

[Mantiuk(2005)] R. Mantiuk, S. Daly, M. Myszkowski, and H.-S. Seidel, *”Predicting Visible Differences in High Dynamic Range Images - Model and Its Calibration”*, SPIE 17th Annual Symposium on Electronic Imaging, Vol, 5666, 2005.

[Mantiuk(2011)] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, *”HDR-VDP-2: A Calibrated Visual Metric for Visibility and Quality Predictions in All Luminance Conditions”*, ACM Trans. on Graphics, Vol. 30, No. 4, pp. 1-14, 2011.

[Menezes(1996)] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *”Handbook of Applied Cryptography”*, CRC Press, 1996.

[Neyman(1933)] J. Neyman, and E. S. Pearson, *”On the Problem of the Most Efficient Tests of Statistical Hypotheses”*, Philosophical Trans. of the Royal Society of London, Vol. 231, pp. 289-337, 1933.

[Petitcolas(1999)] F. A. P. Petitcolas, R. J. Anderson, and M. J. Kuhn, *”Information Hiding: A Survey”*, Proc. of the IEEE, Vol. 87, No. 7, pp. 1062-1078, 1999.

[Provos(2003)] N. Provos, and P. Honeyman, *”Hide and Seek: An Introduction to Steganography”*, IEEE Security & Privacy Magazine, Vol. 1, No. 3, pp. 32-44, 2003.

[Reinhard(2005)] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *”High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting”*, Morgan Kaufmann Publishers Inc., 2005.

[Shannon(1949)] C. E. Shannon, *”Communication Theory of Secrecy Systems”*, Bell Systems Technical Journal, Vol. 28, pp. 656-715, 1949.

[Solachidis(2013a)] V. Solachidis, E. Maiorana, and P. Campisi, *”HDR image multi-bit watermarking using bilateral-filtering-based masking”*, Proc. SPIE Image Processing: Algorithms and Systems XI, Vol. 8655, 2013.

[Solachidis(2013b)] V. Solachidis, E. Maiorana, P. Campisi, and F. Banterle, *”HDR Image Watermarking based on Bracketing Decomposition”*, Proc. Int. Conf. on Digital Signal Processing, 2013.

[Solachidis(2013c)] V. Solachidis, E. Maiorana, and P. Campisi, *”Robust Multi-Bit Watermarking for HDR Images in the Radon-DCT Domain”*, Proc. Int. Symposium on Image and Signal Processing and Analysis, 2013.

[Wang(2012)] Z.-H. Wang, C.-C. Chang, T.-Y. Lin, and C.-C. Lin *”A Novel Distortion-Free Data Hiding Scheme for High Dynamic Range Images”*, Int. Conf. on Digital Home, pp. 33-38, 2012.

[Wu(2012)] J. L. Wu, *”Robust Watermarking Framework for High Dynamic Range Images Against Tone-Mapping Attacks”*, Watermarking, Vol. 2, pp. 229-242, 2012.

[Xue(2011)] X. Xue, T. Jinno, X. Jin, M. Okuda, and S. Goto, *”Watermarking for HDR Image Robust to Tone Mapping”*, IEICE Trans. Fundamentals, Vol. E94-A, No. 11, pp. 2334-2341, 2011.

[Yu(2011)] C.-M. Yu, K.-C. Wu, and C.-M. Wang, *"A Distortion-Free Data Hiding Scheme for High Dynamic Range Images"*, Displays, Vol. 32, No. 5, pp. 225-236, 2011.