INFORMED EMBEDDING: EXPLOITING IMAGE AND DETECTOR INFORMATION DURING WATERMARK INSERTION

Matt L. Miller, Ingemar J. Cox, and Jeffrey A. Bloom

NEC Research Institute
4 Independence Way
Princeton, NJ 08540

ABSTRACT

Usually watermark embedding simply adds a globally or locally attenuated watermark pattern to the cover data (photograph, music, movie). The attenuation is required to maintain fidelity of the cover data to an observer while the watermark detector considers the cover data to be "noise". We refer to this as *blind* embedding. In [1], it was observed that the cover data is not noise, i.e. it is not random but completely known at the time of embedding. This knowledge, along with knowledge of the detection algorithm to be used, allows a new category of *informed* embedder to be realized. In this paper, we describe a simple watermarking algorithm and then compare the performance of blind embedding with three types of informed embedding. Note that in all four cases, the watermark detector is unchanged, only the embedder is altered. Experimental results clearly reveal the improvement of informed over blind embedding.

1. INTRODUCTION

In discussions of watermarking systems, a distinction is generally made between *blind detectors*, which have no knowledge of the original, unwatermarked media, and *informed detectors* (or *non-blind detectors*), which use the original media to assist in detection. A similar distinction can be made between designs of watermark embedders. Although all watermark embedders receive the unwatermarked media as input, many ignore this input when deciding on the watermark pattern to be added. These *blind embedders* work by adding a weak signal to the cover media [2, 3, 4, 5, 6, 7]. Other embedders make partial use of the unwatermarked media to locally attenuate the watermark pattern in regions where it will be perceptable [8, 9, 10]. However, these embedders still ignore the effect of the media on watermark detection.

In [1], it is observed that a watermark embedder can be made more effective if it is designed to exploit the information it has about the cover media, together with knowledge of the watermark detector to be used. This observation leads to a view of watermarking as an example of communication with side information [11]. We refer to such a system as an *informed embedder*.

In this paper, we apply the ideas of [1] to a simple image watermarking system, and compare four different strategies for using the available information. The first of these strategies is simple, blind embedding, with a constant signal-to-noise ratio (SNR). The next two also yield the same constant SNR, but employ information about the cover media and detector to obtain better performance. The fourth strategy employs this information to maintain more constant performance across images, at the expense of variable SNR.

Section 2 of this paper reviews the approach to designing watermark embedders outlined in [1]. Sections 3 and 4 describe the simple watermarking systems we used for our tests. Experimental results are given in Section 5.

2. GENERIC INFORMED EMBEDDING ALGORITHM

We can think of each piece of media as a point in a K-dimensional *media space*. With respect to a given watermark and piece of cover data, the watermark detector and the human observer define two regions of media space. The *region of acceptable distortion* is the set of all points that a human will perceive as being acceptably close to the original cover media. The *watermark detection region* is the set of all points that the watermark detector will categorize as containing the watermark. Ideally, the watermark embedder should output a signal that lies in the intersection of these two regions.

For a watermark embedder to identify the region of acceptable distortion, it must have a perceptual model [10], which is often expressed with a perceptual distance metric. We will use $D(A, B)$ to denote this metric, where A and B are pieces of media.

Identifying the watermark detection region is straightforward. Most detection algorithms can be divided into two basic steps. First the detector extracts a vector, v, from the media content. This vector lies in an N-dimensional *watermark space*, where $N \leq K$. Next, the extracted vector is compared to a predesignated vector, w, that identifies the watermark we are testing for. The comparison produces a *detection statistic*, and if this exceeds a threshold, T, then a watermark is present in the media.

Our informed embedder performs the following steps: (1) extract a signal, v, from the unwatermarked cover media, I. We use $X(I)$ to denote this signal extraction process.

(2) apply a *mixing function*, $f(v, w, I)$, to produce a new vector, v', that is perceptually similar to v, but is inside the watermark detection region around w. (3) modify I to obtain I' using an *inverse extraction function*, $X^{-1}(v', I)$. The inverse extraction function produces a piece of media that yields v' when the signal extraction process is applied to it, and which is perceptually close to I.

Given a watermark detection algorithm, the functions used in steps 1 and 3 are usually straightforward to design. We define a detection algorithm, along with these two functions in the next section. Designing the "mixing function" of step 2 is more subtle. Section 4 is devoted to four different ways this function can be defined for our simple example.

3. A SIMPLE WATERMARK

The watermark detector we develop here is intended only to serve as a clear example of how to implement an informed watermark embedder. This system makes no use of such ideas as embedding in mid-frequencies, correcting for geometric distortions, or sophisticated perceptual modeling.

Our detector first extracts a 64-dimensional vector, v by computing the inner product between the image's pixel intensities and 64 random, orthogonal matrices:

$$v_i = X(I) = \sum_{x,y} I_{x,y} M^{[i]}_{x,y} \tag{1}$$

where $I_{x,y}$ is the pixel intensity at location x, y, $M^{[i]}_{x,y}$ is the value of the i'th matrix at location x, y, and v_i is the i'th element of the extracted signal vector, v. Each of the matrices, $M^{[i]}$, consists of 0's and an equal number of 1's and -1's. At each pixel, exactly one of the matrices has a non-zero value.

The second step, vector comparison, is performed using correlation coefficient, c,

$$c = \frac{\hat{v} \cdot \hat{w}}{\|\hat{v}\| \|\hat{w}\|} \tag{2}$$

where $\hat{v} = (v -$ the mean of $v)$, $\hat{w} = (w -$ the mean of $w)$, and $\hat{v} \cdot \hat{w}$ is the linear correlation (or inner product) between the two vectors. Finally, we choose a threshold value based on the model described in [12] to predict the false positive probability.

The above detector design directly gives the signal extraction function to be used in step 1 of our embedder. We now define the inverse signal extraction function, $X^{-1}(v', I)$, of step 3. If we have an image that yields an extracted vector, v, and we want to change it to yield a different extracted vector, v', we can just add each of the matrices, $M^{[i]}$, to the image, scaled by an amount proportional to $v'_i - v_i$:

$$I' = X^{-1}(v', I) = I + \sum_i \frac{(v'_i - v_i)}{n_i} M^{[i]} \tag{3}$$

where n_i is the number of non-zero values in $M^{[i]}$.

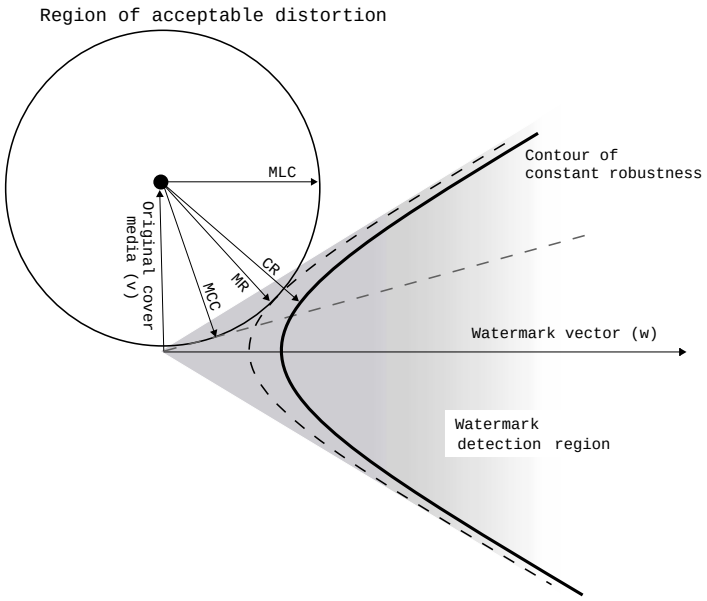

Figure 1: Four watermark embedding strategies.

In practice, our inverse extraction function cannot be implemented perfectly, because of problems with round-off and clipping. Our implementation uses a simple form of error-diffusion to produce an image that yields a close match to v', at the expense of a small fidelity impact.

Finally, to facilitate the design of the mixing functions described in the next section, we must define a method of measuring perceptual distances in media space. We define $D(I', I)$, to be the mean squared error (MSE) between I' and I, scaled by the mean square of I. This yields a value that is equivalent to the signal-to-noise ratio[1], expressed in dB as $10 \log_{10}(1/D(I', I))$.

4. EMBEDDING STRATEGIES

In this section, we explore four different embedding strategies, defined by the mixing function, $f(v, w, I)$. These strategies differ in the values they hold constant and their optimization criteria. The first three hold the distortion, $D(I, I')$, constant while trying to optimize a value related to detection. The fourth method holds a robustness measure constant, while minimizing distortion.

We observe that, in all four cases, the optimal v' will lie in the plane of watermark space that contains the origin, v, and w. Thus, we can restrict ourselves to the two-dimensional problem of finding the optimal point in this plane. Figure 1 illustrates the plane for one combination of extracted vector, v, and watermark, w. Referring to this figure, we now consider each mixing strategy in turn.

Blind embedding or Maximizing linear correlation (MLC). This is the simplest strategy. We assume that the user has specified a limit on fidelity loss. To keep fidelity constant, we scale the watermark vector, w, to a magnitude that corresponds to our fidelity limit, and add it to the extracted signal, v. This strategy maximizes the

[1]Here, the image is the "signal", and the watermark is the "noise".

linear correlation between the resulting signal and w. It is illustrated in Figure 1 by the arrow labeled "MLC".

Note that Figure 1 shows the MLC strategy failing to reach the detection region. In practice, with lower detection thresholds, this occurrence is not as common as implied by this figure. However, MLC will fail to embed the watermark even in some images who's regions of acceptable distortion overlap with the detection region.

Maximizing correlation coefficient (MCC). This strategy maximizes the actual detection statistic used. Thus we want to choose the point along the surface of the region of acceptable distortion that has the highest correlation coefficient with the watermark vector. This is equivalent to finding the vector that forms the smallest angle with the watermark vector. In Figure 1, the arrow labeled "MCC" shows what is added to v by this version of the mixing function.

Since this function explicitly considers the detection statistic, it should succeed in embedding a watermark whenever the region of acceptable distortion overlaps with the detection region. However, it will often choose a vector that is very short (has low amplitude). Even a small amount of noise added to such a vector can easily move it outside of the detection region. Thus, this strategy is expected to embed fairly fragile watermarks.

Maximizing robustness (MR). Here, we explicitly optimize the robustness of the watermark. To do this, we need a formula that gives us some measure of how robust a given signal is, assuming a given watermark and detection threshold. One such formula is suggested in [1][2]:

$$r^2 = \frac{(v' \cdot w)^2}{T^2 \|w\|^2} - \|v'\|^2 \tag{4}$$

where v' is the vector being tested, w is the watermark vector, T is the threshold, and r^2 is a measure of the amount of noise that can be added to v' before its correlation coefficient with w is expected to fall below T.

In the two-dimensional plane of Figure 1, the set of all points that have a given value of r^2 is a hyperbola. Thus we want to find the point on the perimeter of the circle of acceptable distortion which lies on the hyperbola that is deepest within the detection cone. The vector that this mixing method adds to v is illustrated in Figure 1 by the arrow labeled "MR". The dashed hyperbola shows the contour of points that have equivalent robustness to the selected point.

Constant robustness (CR). In this strategy, the user specifies a desired robustness measure, rather than a maximum acceptable distortion. The mixer examines points that all have the given robustness measure, and chooses the one that is closest to the extracted signal, v. In Figure 1, the solid hyperbola shows all the points that have some, specified robustness value, and the arrow labeled "CR" indicates the vector that is added to v by this version of the mixer.

[2]In [1], a typographic error caused the formula to be printed incorrectly. The expression presented here is correct.

This strategy should succeed in embedding a watermark in every image, assuming that the watermark detector uses the same detection threshold that is used to compute r^2 during embedding. However, the fidelity that it achieves will vary.

5. RESULTS

To test the performance of our four embedding strategies, we implemented four versions of the watermark embedder. All the code for these implementations was identical except the portion that implements the mixing function. We then tested them on 2000 images drawn from a Corel image database [13]. Most of these are natural photographs, but there are also some paintings and computer-generated texture images.

Four different watermarked copies of each image were obtained, one for each embedding method. The MR and CR methods were run using an expected detection threshold, T, of 0.55. Next, all the watermarked images were JPEG compressed with a quality factor of 85, using DeBabelizer Pro, and decompressed. The resulting watermarked and attacked images were then run through the watermark detector using three thresholds.

The results of these tests are summarized in Table 1. For each of the three thresholds, the estimated probability of false positive is given as P_{fp}. The results for images after JPEG were computed based only on the images that had watermarks successfully embedded in them. Thus, these results reflect only the effect of JPEG on the detectability of the watermark.

From this table, we make the following observations:

Although the MLC, MCC, and MR embedders try to maintain constant fidelity, there is some variation in the actual fidelity obtained. Similarly, the CR strategy tries to keep constant robustness, but there is some variation in the actual r^2 values obtained.

The blind embedding strategy (MLC) fails to embed a watermark more frequently than any of the others, at all thresholds. This is expected, since MLC ignores the detection region during embedding.

The MCC strategy succeeds in embedding more frequently than MLC or MR at the highest threshold. It has nearly the same success rate as MR at thresholds of 0.45 and 0.55.

Watermarks embedded by MCC are more fragile than those of MR and CR, at thresholds equal to or below the ones used during embedding. This is expected, since MCC will sacrifice robustness in order to maximize correlation coefficient.

Watermarks embedded by MR survive attack better than those of MLC and MCC, when the detection threshold is equal to or less than the one used during embedding. This is expected.

The CR strategy succeeds at embedding watermarks in 100% of the images at every threshold.

Watermarks embedded by CR are the most robust of the four strategies, for thresholds equal to or below the one

		MLC	MCC	MR	CR
SNR (dB) (min/mean/max)		28.3 / 36.5 / 37.8	28.3 / 36.5 / 37.7	28.3 / 36.5 / 37.8	24.8 / 36.0 / 43.2
Robustness Metric (min/mean/max)		0 / 184.3 / 1032.3	0 / 98.5 / 785.8	0 / 214.4 / 1072.4	55.4 / 77.0 / 92.7
$T = 0.45$, $P_{fp} \approx 10^{-4}$	No Attack	98.2%	99.2%	99.3%	100.0%
	After JPEG	63.2%	59.4%	77.0%	77.3%
$T = 0.55$, $P_{fp} \approx 10^{-6}$	No Attack	93.0%	97.4%	97.7%	100.0%
	After JPEG	40.1%	48.7%	58.6%	61.7%
$T = 0.80$, $P_{fp} \approx 10^{-15}$	No Attack	23.5%	83.2%	61.4%	100.0%
	After JPEG	7.4%	23.8%	11.7%	23.0%

Table 1: Fidelity and robustness results for various mixing functions.

used to compute r^2 during embedding.

All but two of these results are expected from the design of the four mixing strategies. The first exception is that MLC, MCC, and MR did not result in exactly constant SNR, and CR did not result in exactly constant robustness. The second exception is that MCC and MR did not have exactly identical failure rates at embedding for a detection threshold of 0.55. However, both of these unexpected results can be explained as a consequence of our imperfect implementation of the inverse extraction function, $X^{-1}(v', I)$.

6. CONCLUSION

From the results presented here, we draw three main conclusions.

First, in a watermarking system with blind detection, the original cover media should *not* be considered noise. Rather, the watermark embedder should exploit the fact that it has complete knowledge of the cover media to embed the watermark more reliably.

Next, the embedder should also employ an accurate model of the watermark detector to identify the exact detection region for the watermark being embedded. This contrasts with the practice of simply adding the watermark to the cover media, which implicitly assumes that, whatever the detection algorithm, detectability increases with linear correlation.

Finally, it is not enough for the embedder to simply try to maximize a detection statistic. We can obtain significantly better results by defining a measure of the expected robustness of the watermark, and maximizing this function (MR) or keeping its value constant (CR).

These principles have been applied to a simple watermarking system, and they significantly improved performance, without changing SNR. We believe they generalize to many other algorithms. However, their impact depends on how different the true detection region is from that created by thresholding linear correlation. This is an area for future research.

Another area to be explored in the future is the development of better robustness measures. The r^2 measure used here is based on a simple model of attacks as additive white Gaussian noise. But this does not accurately model the effects of image processing on our watermark, and measures based on more realistic models might yield better results.

7. REFERENCES

[1] I. J. Cox, M. L. Miller, and A. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE*, 87(7):1127–1141, 1999.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 35(3/4):313–336, 1996.

[3] G. Caronni. Assuring ownership rights for digital images. In *Proc. Reliable IT Systems, VIS'95*. Vieweg Publishing Company, 1995.

[4] L. Holt, B. G. Maufe, and A. Wiener. Encoded marking of a recording signal. UK Patent GB 2196167A, 1988.

[5] K. Matsui and K. Tanaka. Video-steganography. In *IMA Intellectual Property Project Proceedings*, volume 1, pages 187–206, 1994.

[6] G. B. Rhoads. Identification/authentication coding method and apparatus. *World Intellectual Property Organization*, IPO WO 95/14289, 1995.

[7] J. R. Smith and B. O. Comiskey. Modulation and information hiding in images. In R. Anderson, editor, *Information Hiding: First Int. Workshop Proc.*, volume 1174 of *Lecture Notes in Computer Science*, pages 207–226. Springer-Verlag, 1996.

[8] I.J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Trans. on Image Processing*, 6(12):1673–1687, 1997.

[9] C. I. Podilchuk and W. Zeng. Image-adaptive watermarking using visual models. *IEEE Trans. on Selected Areas of Communications*, 16(4):525–539, 1998.

[10] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp. Perceptual watermarks for digital images and video. *Proc. of the IEEE*, 87(7):1108–1126, 1999.

[11] C. E. Shannon. Channels with side information at the transmitter. *IBM Journal of Research and Development*, pages 289–293, 1958.

[12] M. L. Miller and J. A. Bloom. Computing the probability of false watermark detection. In *Proceedings of the Third International Workshop on Information Hiding*, 1999.

[13] Corel Stock Photo Library 3, Corel Corporation, Ontario, Canada