WATERMARKING WITH KNOWLEDGE OF IMAGE DATABASE

Sujoy Roy

School of Computing
National University of Singapore
sujoyroy@comp.nus.edu.sg

Ee-Chien Chang

School of Computing
National University of Singapore
changec@comp.nus.edu.sg

ABSTRACT

The goal of this paper is to study how a-prior knowledge of the image database could be exploited for better watermarking performance. Unlike most formulations, where the encoder and detector only know the distribution of the images, under our formulation, the **actual** *set of images to be watermarked are known, either in a static or dynamic setting. To achieve better performance, instead of choosing a random watermarking key or predefined code-book as is the usual practice, we derive the watermarking keys from the database. We study two settings, static and dynamic. In the dynamic setting, the image database starts from a single image and grows as more images arrive. Thus the watermarking keys have to be updated frequently. This setting can be applied to applications where the detector has access to the Internet. To demonstrate the main idea, we extend a variant of spread-spectrum method to a few schemes, and analyze their performance. Interestingly, the requirements on false-alarm, robustness and distortion can be traded-off with the size of the watermarking keys. We perform our experiments on both natural images and Gaussian source. Our analysis and experiments show promising improvement in performance by exploiting the a-prior knowledge of the image database, specifically for fixed robustness and false alarm we achieve significant reduction of distortion. Similar idea can be incorporated into other watermarking methods.*

1. INTRODUCTION

In the recent years, watermarking has emerged as an active research field. Many theoretical models, and applications have been proposed. In several watermarking models, the watermark detector can communicate with a server. For example, the detector in Digimarc MediaBridge Reader [3] makes use of the Internet to lookup more information based on the extracted message. Theoretical models like zero-knowledge proof [2] and public watermarking also exploit communication to enhance security.

In this paper, we propose another way to exploit communication. In the proposed model, the encoder knows the actual images that are to be watermarked. Based on this knowledge, the encoder tailor-makes an encoding routine (a set of watermarking keys) that is suitable for the database. The watermarking keys are then stored in the server. To determine whether an image is watermarked, the detector first obtains the watermarking keys from the server, which is in turn used in the detection (Fig. 1). Although our discussion centers on images, the main idea can be extended to other multimedia source.

Unlike the model of informed watermarking [5], and watermarking as communication with side information [4] where both the encoder and decoder know the distribution of the images, in the proposed model, we exploit the additional assumption that the encoder knows the **actual** images to be watermarked, and the decoder knows a compact but partial description of the actual images (Fig. 1). Note the fundamental differences between a-prior knowledge of the distribution and knowledge of the actual database. The actual images are samples from the distribution, and can be used to estimate the distribution. On the other hand, knowing the image distribution is not sufficient to determine the actual images. Also note that in practice, the image distribution is usually assumed to be Gaussian, which is an over simplification for natural images.

Compact description of database. Given the database $\mathcal{I}$, a possible but inefficient scheme takes the whole database as the keys, $\mathcal{W} = \mathcal{I}$. Thus nothing is done during encoding and zero distortion is achieved. To decide whether an image is watermarked, the detector searches for the image in $\mathcal{W}$. If it is within the proximity of a key in $\mathcal{W}$, then it is declared to be watermarked. Although this scheme achieves zero distortion, the number of keys is too large and thus inefficient. This brings forth the interesting issue of finding a trade-off between efficiency (number of keys) and performance (distortion, false alarm, and robustness).

Static vs. Dynamic setting. We study two settings. In the *static* setting, the database remains unchanged throughout the encoding and detection process. In the *dynamic* setting,

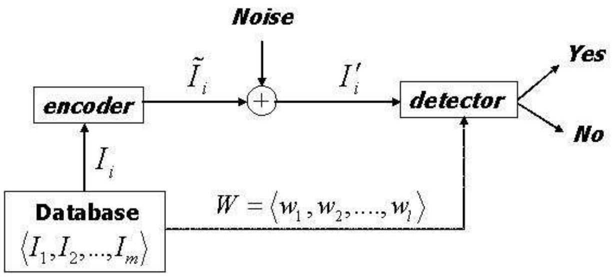
Noise
encoder
$\tilde{I}_i$
I'_i
detector
Yes
No
I_i
Database
$\langle I_1, I_2, \ldots, I_m \rangle$
$W = \langle w_1, w_2, \ldots, w_l \rangle$

Fig. 1. A schematic diagram of the proposed system in the static setting with ℓ keys.

new images can be added to the database. The database starts from a single image, and grows as new images arrive. The encoding must be done in a "on-line" manner, that is, when a new image I arrives, the encoder must immediately encode I before the arrival of the next image. This setting is practically applicable when a watermarking service provider has a dynamic database of images i.e., the database can be updated with new images. The detector, having access to the Internet, can contact the server and retrieve the updated keys.

2. FORMULATION

Let $\mathcal{I} = \langle I_1, I_2, \ldots, I_m \rangle$ be a database of m images. Each image is a sequence of d coefficients which is generated from an underlying source distribution. We study two settings, *static* and *dynamic*.

Static Setting. Given the image database $\mathcal{I}$, the encoder derives a set of keys $\mathcal{W} = \langle w_1, w_2, \ldots, w_\ell \rangle$ and the set of encoded images $\widetilde{\mathcal{I}} = \langle \tilde{I}_1, \tilde{I}_2, \ldots, \tilde{I}_m \rangle$. The keys once determined remains unchanged throughout the process. Given an image I' and the keys $\mathcal{W}$, the detector declares whether it is watermarked (output Yes) or not watermarked (output No). The detection should be robust under the additive white Gaussian noise (AWGN).

Dynamic Setting. The database grows as new images are added. Let $\mathcal{I}_t = \langle I_1, I_2, \ldots, I_t \rangle$ be the database with the first t images, and $\widetilde{\mathcal{I}}_t$ be the corresponding set of encoded image. The encoding is done in an online manner, that is, the image I_t must be encoded before I_{t+1} arrives. Once an $\tilde{I}_t$ is obtained it cannot be recalled for modification.

Similar to the static setting, detection is done based on a set of keys. However, because the database dynamically changes, the keys are updated once a new image arrives. Let $\mathcal{W}_t = \langle w_1^t, w_2^t, \ldots, w_\ell^t \rangle$ be the keys after I_t has arrived and encoded. An additional requirement is that the set $\mathcal{W}_t$ has to be *backward compatible*, i.e., the detector must be able to detect $\tilde{I}_s$, for any $s \le t$ based on $\mathcal{W}_t$. Same as the static setting, the detection should be robust under AWGN.

Performance measures. For completeness and clarity, we give a formal description of *false alarm*, *robustness* and *distortion*. The encoder takes each $I \in \mathcal{I}$ and gives an encoded $\tilde{I}$. We call the average distance

$$\frac{1}{m} \sum_{I \in \mathcal{I}} \|I - \tilde{I}\|_2^2$$

the distortion. The false alarm is the probability of a randomly chosen sequence (from the image distribution) to be declared as watermarked by the detector. The scheme is robust if, under the influence of AWGN, an encoded image is still declared to be watermarked with high probability (the actual probability is not a concern in this paper). We take the variance of the noise as the measure of robustness.

Variant of spread-spectrum method. For the purpose of comparison, we consider a variant of the well-known spread-spectrum method [5]. However, our idea is not restricted to this method. Similar idea of exploiting the a-prior knowledge of image database can be extended to other methods.

This variant is parameterized by a watermark key w, a constant threshold T and a constant K. The encoding of I giving $\tilde{I}$ is achieved by

$$\tilde{I} = I + \max(0, K - I \cdot w) w. \tag{1}$$

The w is normalized so that $\|w\|_2^2 = 1$. An I' is declared to be watermarked if

$$w \cdot I' \ge T. \tag{2}$$

The false alarm, robustness and distortion of this scheme can be obtained analytically.

3. WATERMARKING SCHEMES

This section generalizes the spread-spectrum method to the static and dynamic setting. Their performance is compared with the spread-spectrum method, which can be viewed as an equivalent scheme with no a-priori knowledge of the image database. The false alarm and robustness is fixed in all schemes, which amount to the parameters K and T being fixed. Thus by comparing the distortion we can determine which scheme attains better performance.

3.1. Static with single key

This section gives two schemes (*static* and *static iterative*). The encoder of the first scheme, (*static*), computes the normalized sum of the database, that is,

$$w = \sum_{I \in \mathcal{I}} I / \| \sum_{I \in \mathcal{I}} I \|_2^2, \tag{3}$$

and takes this as the key i.e., $\mathcal{W} = \langle w \rangle$.

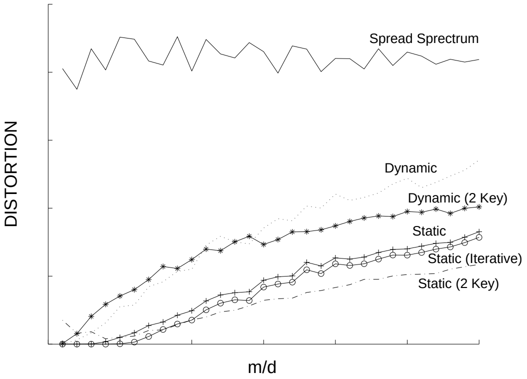

Fig. 2. Distortion verses (m/d) for images from a Gaussian distribution. The number of coefficients is fixed at $d = 1000$, and $K = 0.1414$. For the 2-key setting, the size of the database is twice of that for the single key settings. For example, at $m/d = 0.2$, the number of images is 200 for the single key setting, and 400 for the 2-key setting. (For images from a Gaussian Distribution.)

The encoding and detection is same as the spread spectrum method given in (1) and (2). Unlike the spread spectrum method where the watermarking key is randomly chosen, in this scheme, w is computed from the image database. Note that if both parameters K and T are the same as in the spread spectrum method, then the false alarm and robustness of this scheme are the same as that of the spread spectrum method. By fixing K and T, we want to know which scheme provides lower distortion.

To analyze the reduction in distortion, let us define the *baseline* B as the average of $I \cdot w$ among the images in the database,

$$B = \frac{1}{m} \sum_{I \in \mathcal{I}} I \cdot w.$$

The baseline is a good indicator of the reduction in distortion. If w is randomly chosen (assuming Gaussian distribution with unit variance), then the baseline is expected to be 0. In our scheme, the expected baseline is $1/\sqrt{m}$. Since the baseline is raised to $1/\sqrt{m}$, the distortion required to "push" the image over the threshold K is reduced.

Fig. 2 shows the result of an experiment, confirming the gain in performance of using w obtained from (3) against a randomly chosen key. The experiment is done on images generated from Gaussian distribution. In this experiment, we fix the robustness and false-alarm (that is K and T). The curve labelled as "Static" in Fig. 2. indicates the reduction in distortion from the spread spectrum method. The distortion is reduced by $1/\sqrt{m}$, confirming the above analysis.

The second encoder (*static iterative*) in this section, further improves the performance by searching for the key $\tilde{w}$ which minimizes the average distortion. The minimization is done in an iterative manner. In Fig. 2 (the curve labelled as "Static(iterative)"), noticeable but less significant improvement is achieved by the second encoder.

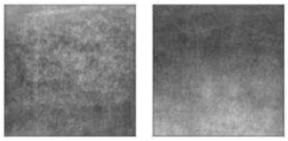

Fig. 3. Keys for the 2-key static setting, for the image database. Note that the 2 keys are almost complementary to each other.

3.2. Static with Multiple keys

In this scheme, instead of using a single key, the detector uses a set of keys $\mathcal{W} = \langle w_1, w_2, \ldots, w_\ell \rangle$. A sequence I is declared to be watermarked if there is an i such that $(w_i \cdot I) > T$.

We now give two encoders for $\ell = 2$. Other values of ℓ can be easily generalized.

The first encoder randomly partitions the database into two subsets of equal size. The key for each subset is generated using (3), by treating each subset as a single database. It is easy to verify that the baseline will improve by a factor of $1/\sqrt{2}$ and this implies that the distortion will also improve by approximately the same factor. On the other hand, the false alarm will increase by a factor of approximately 2. This is because in the 2-key situation, a sequence I is declared to be watermarked if either $(I \cdot w_1) \geq T$ or $(I \cdot w_2) \geq T$. However, constant factor growth of the false alarm is insignificant, because the false alarm decreases exponentially as the threshold T increases linearly. Thus, it is desirable to allow more keys, if efficiency is not a consideration.

The second encoder searches for a good partition of the the database using the well-known 2-mean algorithm. Our experimental results show that the 2-means algorithm gives an improvement in distortion compared to the first encoder which randomly partitioned the database. In Fig. 2, the graph for static 1-key setting can be treated as the performance of the first encoder. The distortion is higher than the graph labelled as "Static (2 key)".

3.3. Dynamic with single key

In the dynamic setting, the images arrive sequentially. Let w_1^t be the key computed after the arrival of I_t. The encoding and detection is similarly performed as in (1) and (2). The key satisfies the additional backward compatibility requirement, i.e., $\tilde{I}_s \cdot w_1^t \geq K$, for any $s < t$. There are two interesting issues. The first issue is concerned with how backward

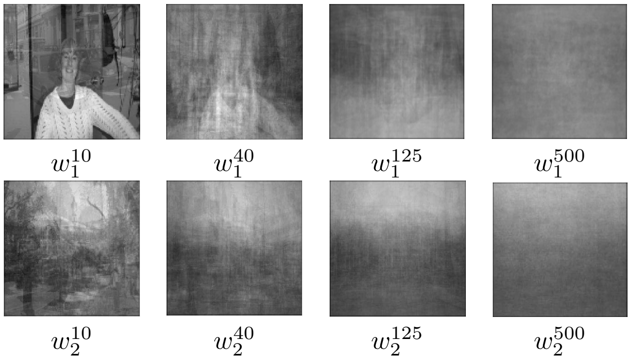

Fig. 4. Evolving of keys in the dynamic 2-key case, for the image database. First row depicts the evolving of the first key for 10, 40, 125 and 500 images. The second row depicts evolving of the second key.

compatibility is to be enforced. It is also interesting to study the reduction in performance when information is available in the on-line manner, as opposed to the static setting where full knowledge of the database is readily available.

On arrival of the $t-$th image, the following iterative method searches for the new key w_1^t. It is important to choose the weighting function as $(1/\sqrt{t})$ in §1.

§1. Let $w_1^t = w_1^{t-1} + (1/\sqrt{t})I_t$.

§2. If there is a $r < t$ such that $I_r \cdot w_1^t < K$, then update $w_1^t = w_1^t + (K - I_r \cdot w_1^t)I_r$.

§3. Repeat step 2 until no such r is found.

3.4. Dynamic with multiple key

In this setting, more keys are allowed as is in the static setting (Section 3.2). The encoder employs a combination of the encoders in the dynamic setting and the 2-key static setting. Details are omitted here. The graph in Fig. 2 shows that this encoder performs better than a 2-key encoder which partitions the database randomly.

4. EXPERIMENT WITH IMAGE DATABASE

We conducted an experiment on a database of 500 natural images from [1]. The main goal of this experiment is to test our idea on non-Gaussian image distributions. Because image representation is not the focus of this paper, we represent each image by its rescaled 128 by 128 gray level image. Fig. 5 shows 3 of these images.

We tested our database using the static single key encoder. The average distortion is 0.41233, which is better than the estimated distortion of 1.0946 if we assume the images are Gaussian. The improvement is due to strong coherence among images in the database.

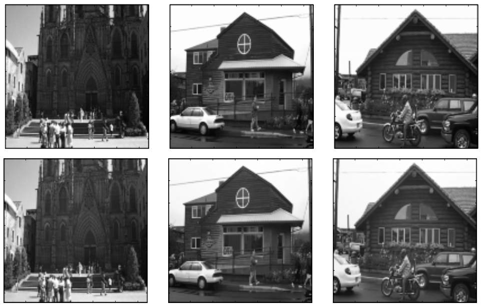

Fig. 5. Three images in the database with the respective encoded images below them.

We also tested our database in the 2-key static and dynamic setting. The keys generated for the static 2-key and dynamic 2-key setting are illustrated in Fig. 3 and Fig. 4 respectively. The keys in Fig. 3 are complementary in the sense that both point towards the opposite direction if treated as vectors in the 128 by 128 dimensional space. This is characteristic of the 2-mean algorithm. The keys in the dynamic 2-key (Fig. 4) setting are updated as new images arrive.

5. CONCLUSION

In this paper, we propose a watermarking formulation which exploits the a-prior knowledge of an image database. Such a formulation is realistic because in some applications, the detector has access to the Internet. We also give few schemes for various settings and analyze their performance based on the assumption that the image and noise are Gaussian. We also test our main idea on non-Gaussian images, which is a set of natural images. Our experiment and analysis show promising improvement in performance by using a-prior knowledge of the image database. Similar idea can be incorporated into other watermarking methods.

6. REFERENCES

[1] http://www.cs.washington.edu/research/imagedatabase.

[2] A. Adelsbach and A. Sadeghi. Zero-knowledge watermark detection. *4th Int. Workshop on Info. Hiding*, LNCS 2137:273–288, 2001.

[3] A.M. Alattar. Briding printed media and the internet via digimarc's watermarking techniology. 2000.

[4] M. Costa. Writing on dirty paper. *IEEE Trans. on Info. Theory*, 29(3):439–441, 1983.

[5] I.J. Cox, M.L. Miller, and J.A. Bloom. *Digital Watermarking*. Morgan Kaufmann, 2002.