Watermarking with Retrieval Systems

Ee-Chien Chang Sujoy Roy
School of Computing
National University of Singapore
{changec, sujoy}@comp.nus.edu.sg

ABSTRACT

We are interested in the problem of associating messages to multimedia content. This problem can be addressed by a watermarking system which embeds the associated messages into the multimedia content (also called Works). A drawback of watermarking is that the content will be distorted during embedding. On the other hand, if we assume that the database is available, the problem can be addressed by a retrieval system. Although no undesirable distortion is created, searching in large databases is fundamentally difficult (also known as the dimensionality curse).

In this paper we present a novel framework which strikes a tradeoff between watermarking and retrieval systems. The framework avoids the dimensionality curse by introducing small distortions (watermark) into the multimedia content. From another perspective, the framework improves the watermarking performance, marked by significant reduction in distortion, by introducing searching ability in the message detection stage. To prove the concept, we give an algorithm based on the proposed notion of "clustering by watermarking".

1. INTRODUCTION

The recent invasion of digital multimedia in an entire range of our everyday life has brought forth two active areas of research, namely, retrieval systems and watermarking technology. Although both these areas are seemingly motivated to address different aspects of multimedia management, a unifying element that brings them together is that both can be used for *identification* of multimedia content [2].

The problem of identification of multimedia content can be understood through the following example. Given a database of images and a query image I we would like to search for I in the database. Once the index of I within the database is found, we can use it to access more information, like author name, creation, date etc. Because we own the image database, we can modify it before it is released to the public.

Both watermarking and retrieval systems can be used for content identification. In watermarking the identifying mark or message is embedded into the content. The identification is done by extracting the message from the content. Recently some efforts have been made in the direction of using watermarking for multimedia content identification. Digimarc's MediaBridge Reader [3] uses the concept of "smart images" wherein the watermarked message includes pointers to some knowledge structure on a local database or on the Internet. The watermark (message) detector extracts the message and hence extracts more information from the database. In retrieval systems, identification is done by extracting essential perceptual features from the content and using some similarity matching technique to search through the contents in the database, represented by their pre-computed feature values [5]. Kalker et. al [2] brings out some relationship between watermarking and retrieval systems (using perceptual hashing), but no concrete technique to combine them is specified.

Motivated by the ideas in watermarking and retrieval systems, we reformulate the identification problem to one that allows us to look into the tradeoff between watermarking and retrieval systems (Section 3). Our approach in achieving such a tradeoff is by applying *clustering by watermarking* (Section 4).

2. RETRIEVAL VS WATERMARKING

Here are two views that lead to the proposed framework of looking for a tradeoff: from a watermarking and retrieval perspective.

Watermarking Perspective. There are a number of watermarking models. Under non-blind watermarking, the decoder has the original Work of the watermarked Work. For blind watermarking, the decoder does not have the original Work. However, it knows the underlying distribution of the original Works. For instance, the results by Costa [4] require both the encoder and decode know the distribution. This motivates us to investigate situation where the encoder and decoder know the **actual** Works database. That is, the encoder knows the set of Works to be watermarked. With this additional information, the encoder can tailor-make watermarking codes suitable for this particular database. Unlike the predefined codes, these tailor-made codes are not well-structured, and thus searching is required during decoding.

Retrieval Perspective. In some scenarios, the decoder can communicate with server (for e.g. Digimarc MediaBridge Reader[3] and zero knowledge watermark detection[1]), and thus can receive more information on the wa-

Workshop on Multimedia and Security, ACM MM '02 Juan-les-Pins, France
Copyright 2001 ACM ...$5.00.

termarked Works. With the additional knowledge, alternative to the direct usage of a watermarking technique should yield higher performance. An alternative is a retrieval system. The retrieval system stores the content-message pairs, $(I_1, m_1), (I_2, m_2), \ldots$ in a database and takes the original Works $I_1, I_2, \ldots$ as keys. Given a query image, the associated message is obtained by searching the query image in the database.

However, the retrieval method has two limitations. Firstly, it is inefficient because searching in high dimension space is fundamentally difficult. The computational requirement increases rapidly as the dimension increases. This phenomenon is generally referred to as the dimensionality curse. Our identification problem is usually applied to Works which have very high dimension. For example, the dimension of images can ranges from 500 to millions, depending on the underlying image transformations and features space.

Another limitation of the retrieval method arises when some of the original Works are similar. In the worst case, all Works are identical, say $I = I_1 = I_2 = \ldots$. Now, given a query I, it is impossible to decide which is the associated message. The watermarking method solves this ambiguity naturally. Under watermarking, the messages $m_1, m_2, \ldots$ are embedded into I separately, giving different watermarked $\widetilde{I}_1, \widetilde{I}_2, \ldots$. Given $\widetilde{I}_i$ as query, the decoder can correctly output the message m_i without ambiguity.

Although the retrieval method is computationally expensive and introduces ambiguity, it achieves zero distortion. This is in contrast to the watermarking solution, which generates undesirable distortions, but achieves fast retrieval and resolves ambiguity. Hence, the interesting question is, whether we can combine both techniques and find the right tradeoff for better performance. This is the focus of this paper.

3. PROBLEM FORMULATION

We re-formulate the identification problem as a variant of the classical nearest-neighbor search in high dimensions, but with the additional freedom of modifying (that is, watermarking) the data points.

Formulation. In this paper, each Work is represented as a point in $\mathbf{R}^d$, and the distance measure between the Works is the Euclidean 2-norm. Given the original database $\mathcal{I} = \langle I_1, I_2, \ldots, I_n \rangle$, a distortion constraint ϵ and robustness σ^2, we want to preprocess $\mathcal{I}$ to obtain the watermarked $\widetilde{\mathcal{I}} = \langle \widetilde{I}_1, \widetilde{I}_2, \ldots, \widetilde{I}_n \rangle$ and an index tree. The watermarked $\widetilde{\mathcal{I}}$ satisfies the distortion constraint ϵ, that is,

$$\frac{1}{n} \sum_{i=1}^{n} \|I_i - \widetilde{I}_i\|_2^2 < \epsilon. \tag{1}$$

The index tree facilitates searching such that given the query $\widetilde{I}_i$, we can output i efficiently. The searching is robust in the sense that if $\widetilde{I}_i$ is corrupted by additive white Gaussian noise (AWGN) with power σ^2, the output is correct with high probability. Specifically, suppose

$$I' = \widetilde{I}_i + z,$$

where $z = (z_1, z_2, \ldots, z_d)$ and each z_j is independently drawn from the normal distribution $N(0, \sigma^2/d)$. Then, taking I' as the query, the algorithm gives the correct output (which is i) with probability at least $(1 - 1/d)$.

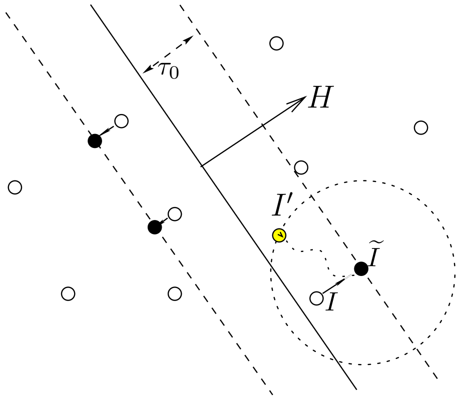

Figure 1: Each circle represents a Work. Each filled circle represents the corresponding watermarked Work, if it is different from the original. The region between the two dotted lines is the buffer zone, and its width is τ_0. The point I is an original Work, $\widetilde{I}$ is the watermarked Work and I' is a corrupted query. The normal of the separating hyperplane H can be viewed as the "watermark". Those Works on the left halve contain the watermark $-H$, while those on the right contain watermark H.

In the above formulation, the messages associated to the Works are actually its indices. This is different from our original description where the messages m_i could be a string. This difference is not critical because the actual message m_i can be easily looked up from a table.

Coding. A solution to our problem has to address two issues. The first is regarding coding. If $I_1 = I_2 = \ldots = I_n$ are identical, then the problem is same as informed watermarking, that is, watermarking with original Works available at the decoder. Because there is only one Work, we can use it as the reference point. This reduces the problem to finding the watermarked $\widetilde{I}_1, \widetilde{I}_2, \ldots, \widetilde{I}_n$ that are far apart but subject to the distortion constraint $\sum_i \|\widetilde{I}_i - I_i\|_2^2 \leq n\epsilon$. This is essentially channel coding, where ϵ is the power constraint and σ^2 is the noise variance. Note that high dimensionality is required to encode large number of messages.

Searching. The other issue is the computational aspect of searching. Compared to the classical nearest-neighbor approach, in our proposed framework the data points can be slightly modified for better searching performance. In the extreme, with unlimited distortion, the problem is trivially solved by aligning the watermarked Works along a straight line. Since distortion is undesirable, we want to minimize the distortion while supporting fast retrieval.

4. CLUSTERING BY WATERMARKING

In this section, we propose an indexing algorithm based on hierarchical clustering. This size of the index tree is linear and the search time is $O(\log n)$.

To build the index tree, the algorithm first finds a hyperplane that separates $\mathcal{I}$ into two balanced (within a constant factor) clusters. The Works are then watermarked so that none of them are located near the hyperplane. The normal of the hyperplane can be viewed as the watermark in the well-known spread spectrum method. Finally, each

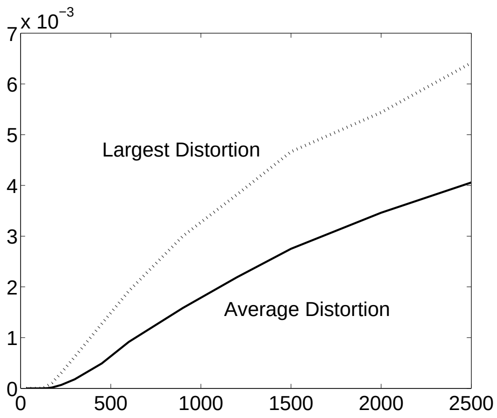

Figure 2: Distortion versus size of database.

cluster is recursively divided into sub-clusters. Let us call the slab (region between two parallel hyperplanes) that does not contain any watermarked Works the *buffer zone*, and the distance of the hyperplane to the buffer zone's surface the buffer zone's *width* (τ_0) (Figure 1).

The hierarchical clustering gives an index tree for searching. The internal nodes of this tree are the separating hyperplanes, and the leaves are the index of the only Work in the corresponding cluster. Given a query, say the watermarked $\widetilde{I}_i$, it is easy to transverse the tree from the root down to the correct leave (which is i) by comparing I_i with the internal nodes along the path. Under influence of AWGN, the query become $I' = \widetilde{I}_i + z$ where z is the noise. This additive noise might lead to error. Recall that the hyperplane is surrounded by a thick buffer zone. The width of this buffer zone is chosen so that the probability of I' crossing the hyperplane is extremely small. Thus, robustness is achieved.

Since the index tree contains of at most n hyperplanes, and each hyperplane can be represented by its normal and a point on its surface, the total size of the index tree is linear with respect to the size of $\mathcal{I}$. Because the tree is balanced, the depth of the tree is $O(\log n)$. We tested our algorithm on Works generated from Gaussian source and natural images. In our experiments, the index trees are always successfully built by the proposed heuristic algorithm.

There are a number of interesting technical issues, for example, (1) how to determine the correct buffer width for the given requirements on distortion, and (2) computational efficiency of the algorithm. The details can be found in the long version of this paper.

5. EXPERIMENTAL RESULTS

We conduct two sets of experiments. In the first set, the Works are generated from Gaussian source. In the second set, the database consists of natural images, resized to 64 by 64 gray-scaled pixels (Figure 3).

Random Works. In these experiments, Works are generated from a Gaussian source, more specifically, it is a multivariate Gaussian random variable $I = (x_1, x_2, \ldots, x_d)$ where each x_i is normally distributed with variance $1/d$.

Figure 2 shows the overall distortion (generated by multi-level clustering) as the number of Works increases. The width of buffer zones in all levels is kept at $\tau_0 = 3\sqrt{2/d}$. This value is chosen so that retrieval is robust under noise variance $\sigma^2 = 2$. That is, when the signal-to-noise ratio is at most 0.5. The distortion is generally very small. For example, at $n = 2048$, the distortion is 0.0035. This is much smaller than the energy of the Works (which is 1). It is also significantly smaller than the noise variance $\sigma^2 = 2$.

Natural Images. In this set of experiments, the database consists 2048 natural images. The purpose of these experiments is to test our idea on non-Gaussian source. These images are gray-scale image resized to 64×64 pixels. Thus, the dimension $d = 64^2$. The images are normalized so that each has unit energy. Because image representation is not a key issue here, we simply take the down-sampled images as the features to work in. Figure 3 shows samples from the database. Unlike the database of random Works, some of the images are similar. Similar images are more difficult to handle, because they should be separated to resolve ambiguities.

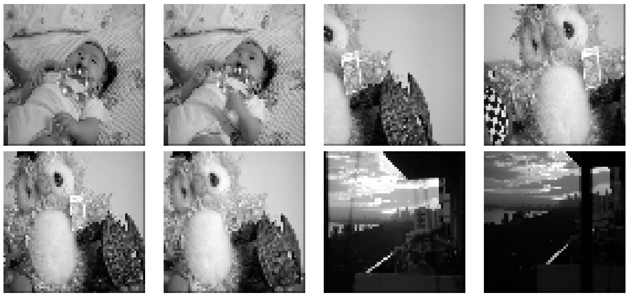

Figure 3: Eight sample images from the database.

The robustness σ^2 is chosen to be 2. This translates to the buffer zone's width of $\tau_0 = 3\sqrt{2/d}$. Figure 4 shows three instances of corrupted queries. Our algorithm successfully retrieves the correct index for (a) and (b), but not (c). The experiment is repeated for 1000 times, with same noise variance, but different noise instances. With noise variance of 1 and 2, our algorithm outputs the correct index for all instances. With noise variance of 4, it gives correct index in 655 instances.

The average distortion generated is 8.5×10^{-4} and the maximum distortion among the images is 0.010. Figure 5 shows two of the watermarked images. The top row is the image with the maximum distortion.

Figure 6 shows selected nodes of the tree at the 1^{st}, 4^{th} and 8^{th} level. These nodes are visited while searching for the top-right image in Figure 3. That is, the query image is first compared with (a) and finally compared with (c).

5.1 Comparison with watermarking

It is interesting to compare the performance of our algorithm with methods based solely on watermarking. For the purpose of comparison, we consider watermarking schemes which fall into the framework of Gaussian channel with side information. Costa [4] showed that, the maximum achievable rate with distortion ϵ and robustness σ^2 is

$$C = \frac{d}{2}\log \quad 1+\frac{\epsilon}{\sigma^2} \quad . \tag{2}$$

That is, the maximum number of messages that can be embedded is 2^C. If we employ solely watermarking to solve the identification problem, with the constraint on distortion

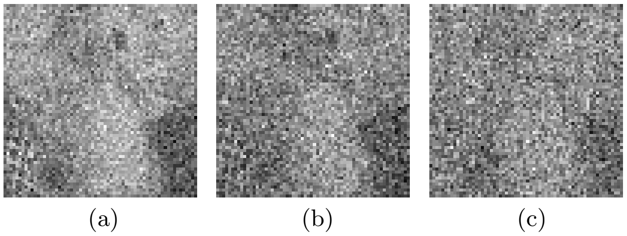

Figure 4: Three corrupted queries. The noise variance is (a) 1, (b) 2 and (c) 4 respectively. The uncorrupted image is shown in the top-right corner of Figure 3. The proposed algorithm correctly retrieves the index for (a) and (b), but not (c).

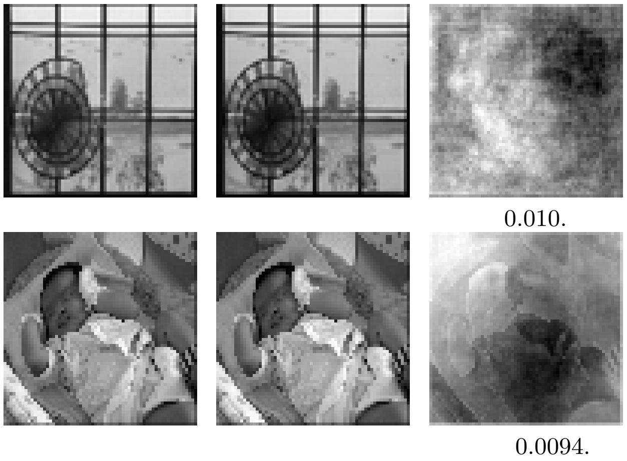

Figure 5: Images in *left* column are the original, *middle* are the respective watermarked image and *right* are the differences. The differences are enhanced (by scaling the intensity) for better printing quality. The values below the images are the distortion.

and robustness, the size of the database is bounded above by $(1+\epsilon/\sigma^2)^{d/2}$. From the experimental data in Section 5, with robustness $\sigma^2 = 2$, dimension $d = 64^2$ and distortion $\epsilon = 0.0035$, our method can have 2048 Works. In contrast, the theoretical maximum number achievable by watermarking is $(1+0.0035/2)^{d/2} < 36$.

6. FUTURE WORKS

The main aim of this paper is to demonstrate that we can combine watermarking techniques and retrieval systems. Currently, we are looking into a number of practical issues, for example, (1) choosing the suitable image presentation and features, (2) considering other noises like affine transformation, and (3) allowing insertion of new Works into the database. Many interesting issues remain open.

7. CONCLUSION

In this paper, we observe that the identification problem can be addressed by a combination of watermarking methods and retrieval systems. This combination can be viewed as watermarking with the detector having access to the Work database. We can also view it as a variant of retrieval problem where the data points can be slightly distorted. To prove this concept we give an algorithm which is a combination of watermarking techniques and clustering algorithm. This improves performance both from a retrieval and watermarking perspective. Firstly, from a multimedia retrieval perspective, by introducing small distortions we can search faster, achieving logarithmic retrieval complexity. Secondly, from a watermarking perspective with some searching ability we can significantly reduce distortion and thus improve watermarking performance.

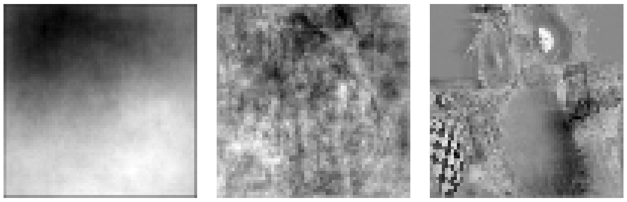

Figure 6: The normal H of the hyperplanes computed at the 1^{st}, 4^{th}, 8^{th} level are depicted as images from left to right respectively. These normals can be viewed as the "watermarks".

8. REFERENCES

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

[2] A.Kalker, J.Haitsma, and J. Oostveen. Issues with digital watermarking and perceptual hashing. In *SPIE Conf. on Multimedia Sys. & Appl.*, August 2001.

[3] A.M. Alattar. Briding printed media and the internet via digimarc's watermarking techniology. *Multimedia and security Workshop, ACM MM*, 2000.

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

[5] J. Haitsma, J.C. Oostveen, and A. Kalker. Robust audio hashing for content identification. In *Content based multimedia Indexing (CBMI), Brescia Italy*, 2001.