Watermarking with Retrieval Systems

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

ABSTRACT

We are interested in the method of associating messages to multi-media objects. The association should be done in a way that the message can be readily extracted from a query object, and the extraction is robust even in the presence of noise. This problem is typically addressed by watermarking where the the messages are embedded into the original objects (the multi-media objects are also known as Works in watermarking). A disadvantage of watermarking is that the Works will be distorted during embedding. In some scenarios, we can assume that information (either fully or partially) of the Works database is available. This additional knowledge can be exploited to reduce distortion. For example, instead of watermarking, the original Works and their associated messages can be stored in a database. The retrieval mechanism is then used to obtain the associated message from a query Work. Although this method induces zero distortion, it is computationally intensive and thus not feasible in most practical applications. In this paper, we study the tradeoff between these two extremes. We view the problem of message-Work association as a variant of database retrieval problem. In contrast to the usual retrieval problem, our database entries can be modified for faster retrieval. We propose an algorithm which uses linear storage space (with respect to the size of database) and provides fast retrieval. Our claims are supported by experimental studies.

General Terms

multimedia retrieval, multimedia security

Keywords

Watermarking, nearest-neighbour search, k-mean, image retrieval system

1. INTRODUCTION

A watermarking scheme consists of an *encoder* and *decoder*. The encoder embeds a given message m into a Work I, giving the watermarked $\widetilde{I}$. The embedding is done in such a way that the distortion from I to $\widetilde{I}$ is small, and the message m can be successfully decoded from $\widetilde{I}$ even if $\widetilde{I}$ is corrupted by noise. Watermarking techniques can be applied to various domains, for example images, audio, and 3-d meshes, and they can be employed in many applications. A survey on current watermarking techniques and their applications can be found in [6]. We are particularly interested in applications which use watermarking as an efficient mean for associating messages to Works. The associations are robust against corrupting noise, and the associated message can be readily retrieved from a query Work. The association of m to I is typically achieved by watermarking in the following way: For each association (m, I), the encoder embeds message m into I, giving a watermarked $\widetilde{I}$. Given a query $\widetilde{I}$, the retrieval of the associated m is performed by the decoder, who extracts the message from the watermarked $\widetilde{I}$. Through watermarking, the message can still be successfully extracted even if $\widetilde{I}$ is corrupted by noise. In this paper, each Work is represented as a point in $\mathbf{R}^d$, and the distance measure between Works is the Euclidean 2-norm. The security of the association is not a concern of this paper. Thus, we only consider additive white Gaussian noise as the corrupting factor.

In some scenarios, the encoder and decoder have partial or full access to the set of associations. For example, in the zero knowledge watermark detection[1], and the commercial product Digimarc MediaBridge Reader[2, 7], the decoder can communicate with a server, and thus can receive more information on the watermarked Works. With the additional knowledge of the associations, alternative to the direct usage of watermarking technique should yield higher performance. A seemingly possible but inefficient alternative is a multi-media retrieval system. The retrieval system stores $(I_1, m_1), (I_2, m_2), \ldots$ in a database and take the original Works $I_1, I_2, \ldots$ as indices. 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 and maintaining the high dimensional Works are computationally intensive. In particular, the performance of this method relies heavily on the nearest-neighbour search, which can be stated as follow: given a set of points in d-dimensions, with preprocessing allowed, how quickly can a nearest neighbour of a given query point q be found. Nearest-neighbour search is an important operation in retrieval systems and many algorithms have been proposed, such as R-tree[10], PMR quadtree [11], k-d-trees[3] and their variants [12, 13, 9]. The computing resources required by these algorithm are measured by the size of the index tree and the search time. In most algorithms, the required resources increases rapidly as the dimension of search space increases. This phenomenon is generally referred to as the dimensionality curse and is usually avoided by reducing the dimensionality of the search space. Our message-Work association problem is usually applied to multi-media objects 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. Reducing the dimensionality is not a good option here because higher dimensionality is required for coding more messages.

Another limitation of the retrieval method surfaces 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 computational expensive and introduces ambiguity, it achieves zero distortion. This is in contrast to the watermarking solution, which generates undesire distortion, but achieves fast retrieval and resolves ambiguity. Now, it is an interesting question whether we can combine both techniques and find the right tradeoff for better performance. This is the focus of this paper.

Motivated by the above observation, we re-formulate the message-Works association problem addressed by watermarking. This new formulation can be viewed as a variant of the classical nearest-neighbour search in high dimensions, but with the additional freedom of modifying (that is, watermarking) the data points. We can also viewed this as a variant of watermarking coding problem, but with addition knowledge of the joint distribution of message-Works (that is, the distribution of message and Works are not independent). We give a solution which is a combination of watermarking techniques and clustering. This algorithm uses linear storage space (with respect to the database size) and facilitates fast retrieval. Our claims is supported by experimental studies.

Outline of this paper. In Section 2, we formulate the association problem and highlight some similarities and differences with existing formulations. The proposed algorithm is described in Section 3. The single-level clustering is given in Section 3.1 and the extension to multi-levels in Section 3.2. In Section 4, we gives experimental results to support our claims, and compares the performance with the theoretical limit of watermarking. Extensions and variations of our formulation are discussed in Section 5.

2. PROBLEM FORMULATION

Given $\mathcal{I} = \langle I_1, I_2, \ldots, I_n \rangle$, 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,

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

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 nose (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 $(1 - 1/d)$.

This formulation can be rephrased to an optimization problem. By fixing the distortion constraint, we want to find an index tree that maximize the robustness σ^2, or vice versa, fixing σ^2 and minimizing the distortion.

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 lookup 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 $\sum_i \|\widetilde{I}_i - I_1\|_2^2 \leq n\epsilon$. 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 on the computational aspect of searching. As we have mentioned in the introduction, the dimensionality curse prevents fast searching. Fortunately, a few differences of our problem from the classical nearest-neighbour search can be exploited. The most notably difference is that, in our problem, the data points can be slightly modified (watermarked) 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.

A more subtle difference arises from the probability requirement on searching. Our formulation does not state the required output when the query is seriously corrupted. In other words, if the query point is not near any data point, it can be ignored. In addition, the query's outcome is probabilistic. Thus, if a query locates about the same distance from $\widetilde{I}_1$, $\widetilde{I}_2$ and $\widetilde{I}_3$, instead of looking into the fine neighbourhood structure, we can just make an arbitrary choice, as long as it conforms to the probabilistic requirement. Furthermore, we are interested in average case performance. We assume the Works are normally distributed. Our algorithm exploits these statistical properties and avoid difficult issues in nearest-neighbour search.

3. CLUSTERING WITH MODIFICATION

In this section, we propose an algorithm based on hierarchical clustering. This algorithm first finds a hyperplane that separates $\mathcal{I}$ into two balance (within a constant factor) clusters. The Works are then watermarked so that none of them is located near the hyperplane. Finally, each 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* (Figure 1).

The hierarchical clustering gives an index tree for searching. The internal nodes of this tree are the separating hyperplane, 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' crosses the hyperplane is extremely small. Thus, robustness is achieved. In Section 3.2, we will quantify how large the buffer zone to be for a required robustness.

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 balance, 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.

We will first describe the single-level clustering (Section 3.1). In Section 3.2, we describe how to perform recursive clustering while ensuring requirements on robustness and distortion are achieved.

3.1 Single level clustering

The single level clustering attempts to solve this sub-problem: Given $\mathcal{I} = \langle I_1, I_2, \ldots, I_n\rangle$, a distortion requirement ϵ_0 and the buffer zone's width τ_0. We want to find a hyperplane (represented by its normal H_0 and a point C_0 on the plane), and a watermarked $\widetilde{\mathcal{I}} = \langle \widetilde{I}_1, \widetilde{I}_2, \ldots, \widetilde{I}_n\rangle$, such that:

1. The distortion is at most ϵ_0, that is $\sum_i \|\widetilde{I}_i - I_i\|_2^2 \le n\epsilon_0$.
2. For any watermarked $\widetilde{I}$, the distance of $\widetilde{I}$ from the hyperplane is at least τ_0 (that is, $|(\widetilde{I} - C_0)\cdot H_0| > \tau_0$).
3. Furthermore, the hyperplane divides the watermarked hosts into two equal (within constant factor) halves. That is, suppose $\mathcal{I}_0$ is the set of watermarked $\widetilde{I}$ where

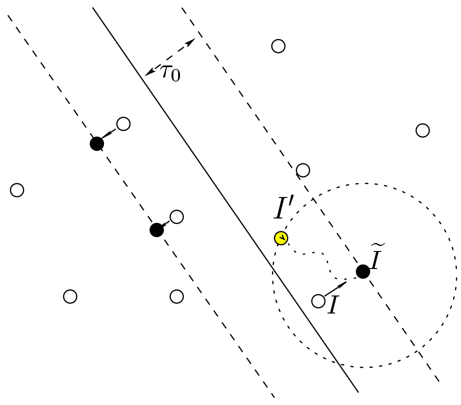

Figure 1: Each circle represents a Works. Each filled circle represents the corresponding watermarked Work, if it is different from the original. The region between the two dotted lines is the buffer zone, and its width is τ_0. The point I is an original Work, $\widetilde{I}$ is the watermarked Work and I' is a corrupted query.

$(\widetilde{I} - C_0)\cdot H_0 > 0$, then
$$\frac{1}{4} < |\mathcal{I}_0|/|\mathcal{I}| < \frac{3}{4}.$$

Figure 1 illustrates the result of a single level clustering in 2-dimensional space. This problem can be rephrased as an optimization problem by fixing the buffer zone's width τ_0 and minimizing the distortion, or vice versa.

Here is a simple heuristic based on the 2-mean algorithm.

1. Compute the 2 means, m_0 and m_1 using the well-known iterative method [8]. Let $\widehat{H} = m_0 - m_1$ and $\widehat{C} = (m_0 + m_1)/2$.
2. Partition $\mathcal{I}$ into two clusters $\mathcal{I}_0$ and $\mathcal{I}_1$, where $\mathcal{I}_0$ contains all the Works in $\mathcal{I}$ that is nearer to m_0, and $\mathcal{I}_I$ contains the remaining. Specifically, if $(I - \widehat{C})\cdot\widehat{H} > 0$, then I is in $\mathcal{I}_0$.

   Next, find a "good" hyperplane that separates $\mathcal{I}_0$ and $\mathcal{I}_1$. Ideally, we want to find the hyperplane with the maximum distance from its nearest Work. That is, we want to find the support vectors for the two clusters. Support vector machine is an established technique, and the support vectors can be efficiently found using quadratic programming [4]. In our current implementation, instead of finding the optimal hyperplane, we use a simple approximation. The details of this simple iterative method is omitted here. Let H_0 and C_0 be the normal and a point on this hyperplane respectively.
3. For all I in the buffer zone, watermark them by shifting them along the direction H_0 and away from C_0. They are shifted until they reach the surface of the buffer zone. Specifically, if $(I - C_0)\cdot H_0 \ge 0$, then the watermarked $\widetilde{I}$ is
$$\widetilde{I} = I + \max(0, \tau_0 - (I - C_0)\cdot H_0)H_0, \qquad (2)$$

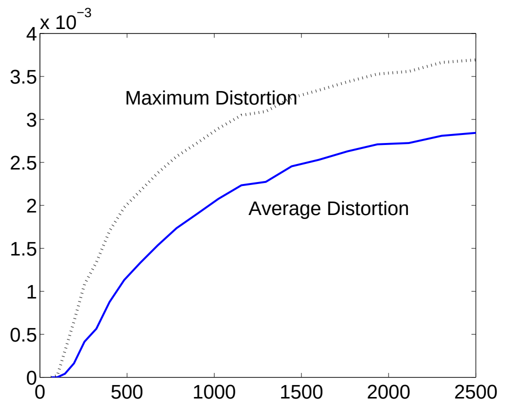

Figure 2: Performance of the single level clustering as the number of Works increases. The dimension $d = 64^2$ and the width of buffer zone is $\tau_0 = 5/\sqrt{d}$. The upper graph gives the largest distortion among the n Works. The lower graph gives the average distortion.

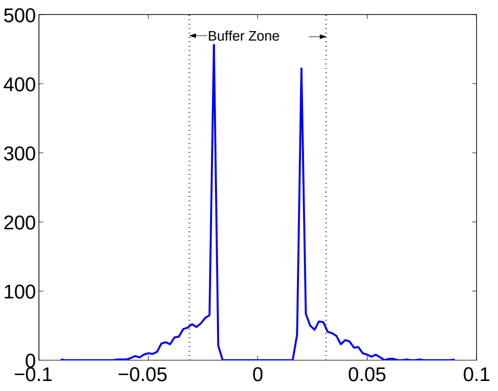

Figure 3: Histogram of the distances of original Works from the hyperplane. The dimension $d = 64^2$, $n = 2048$. In-between the two vertical lines is the buffer zone with width $\tau_0 = 2/\sqrt{d}$.

otherwise

$$\widetilde{I} = I - \max(0, \tau_0 + (I - C_0) \cdot H_0)H_0,$$

Now, we want to find the relationship between τ_0 and the required robustness σ^2. Consider I' in Figure 1. The point $I' = \widetilde{I} + z$ is corrupted by noise z. Error occurred during searching if the noise vector z, after projected onto the one-dimensional normal H_0, is more than τ_0 (or $-\tau_0$ depending on which side $\widetilde{I}$ is in). Because the noise is AWGN with variance σ^2, the distribution of the one-dimensional projected noise is also normally distributed but with variance σ^2/d. Since the probability of deviation from the standard deviation $\sqrt{\sigma^2/d}$ is small, we choose τ_0 to be

$$\tau_0 = A_d\sqrt{\sigma^2/d}, \quad (3)$$

where A_d is a slow-growing function, for example $\log d$. In our experimental studies (Section 4), instead of a slow-growing function, we choose A_d to be the constant 3. This gives the probability of error about 0.0015.

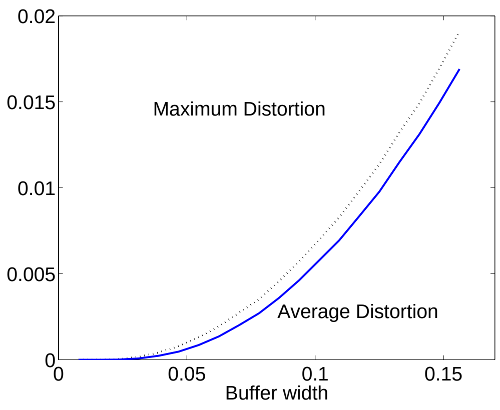

Figure 4: Distortion versus number the buffer zone's width τ_0. The number of Works is fixed at $n = 2048$ and the dimension is $d = 64^2$.

3.2 Extension to Multi-level

Extending the single level clustering to multi-level without special care might violate the robustness requirement. Recall that step 3 in Section 3.1 moves Works out of the buffer zone. There are chances that the newly watermarked Works re-enter the buffer zone created in previous levels. Geometrically, the buffer zone is an union of slabs (each clustering contributes one slab), and the non-buffer zone is divided into disjoint polyhedras. The task of watermarking is to move original I out of the buffer zone and to the nearest point in non-buffer zone, which is on the surface of a polyhedra. For simplicity in implementation, instead of finding the nearest point on the polyhedras, we iterate step 3 to ensure buffer zones in all levels are empty. This iteration might not give the nearest point. However, its converges fast and gives good approximation, probably because the hyperplanes are nearly orthogonal.

In the optimization version (fixing distortion and maximizing robustness), the allowable amount of distortion is valuable resources and has to be allocated to different levels. The allocation should be fair so that every level of clustering achieves same robustness. Let $\epsilon_0, \epsilon_1, \ldots \epsilon_k$ be the distortion allocated to the k levels. Assuming that the hyperplanes at different levels are orthogonal, then the overall distortion is $\sum_i^k \epsilon_i = \epsilon$. We shall see in next section and Figure 2 how distortion increases as n increases. The allocation of ϵ should be $\epsilon_i = \epsilon B/\mathit{Dist}(2^{-i}n)$, where $\mathit{Dist}(\cdot)$ is the average distortion as a function of n estimated from the empirical data, and B is a normalizing factor such that $\sum_i B/\mathit{Dist}(2^{-i}n) = 1$.

4. 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 are natural images, resized to 64 by 64 gray-scaled pixels (Figure 7).

Random Works. In these experiments, Works are generated from a Gaussian source, more specifically, it is a multi-

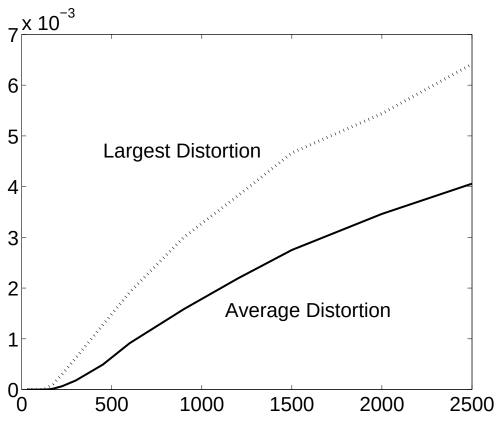

Figure 5: Distortion versus size of database.

variate Gaussian random random variable $I = (x_1, x_2, \ldots, x_d)$ where each x_i is normally distributed with variance $1/d$.

Figure 2, 3 and 4 illustrate the performance of the single level clustering. Figure 2 gives the average distortion as the number of Works increases. When the number of Works increases, the computed 2-means in Section 3.1 step 1 are closer to the overall average. Thus, the distortion should increase.

Figure 3 shows the distribution of the distance of the original Works from the hyperplane. Note that these are the distances before watermarking. Observe that the center region is empty. This is because the hyperplane is derived from the support vectors. Thus, the slab enclosed by the support vectors is empty, even before watermarking. Recall that we do not find the optimal support vectors. The two peaks in the histogram are side-effects of our approximation algorithm. The two vertical lines in the figure indicate the buffer zone with $\tau_0 = 2/\sqrt{d}$. Works fall between these two lines have to be watermarked. Figure 4 shows how the width τ_0 affects the distortion. Observe from the histogram that the Works are concentrated around 0.025 and -0.025. Thus, for large τ_0, the distortion is approximately the square of the distance of τ_0 from 0.025. This observation is confirmed in Figure 4, where the distortion is roughly proportional to $(\tau_0 - 0.025)^2$.

Figure 5 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 0.0027. This i;s much smaller than the energy of the Works (which is 1). It is also significantly smaller than the noise variance $\sigma^2 = 2$. Figure 6 illustrates how distortion decreases as the dimensionality increases. The size $n = 512$ and width $\tau = 3\sqrt{2/d}$. Interestingly, performance improves as dimensionality increases. This is in contrast to general searching problems in high dimensional space.

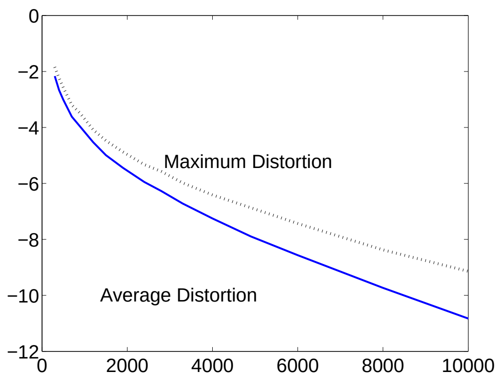

Figure 6: Distortion versus dimension. Logarithmic scale is used for the y-axis. The number of Works $n = 512$ and width $\tau = 3\sqrt{2/d}$.

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. Although images are typically much larger than 64 × 64 pixels, for watermarking purpose, their dimension is usually reduced to remove redundancies and coherence among the pixels. Because image representations are not a key issue here, we simply take the down-sampled images as the domain to work in. Figure 7 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.

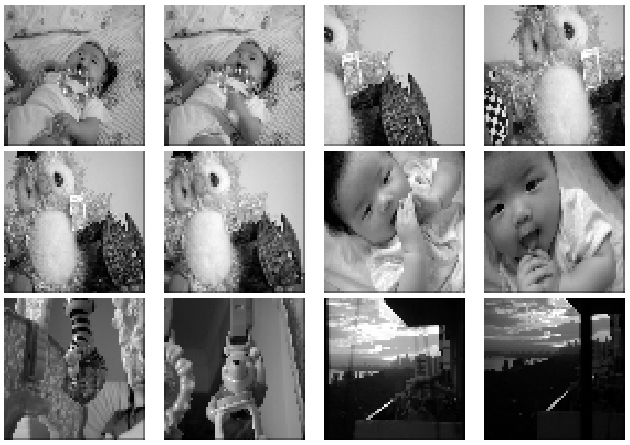

Figure 7: Twelve sample images from the database.

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 8 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. In our implementation, the queries are normalized to unit energy before searching.

The average distortion generated is 8.5×10^{-4} and the maximum distortion among the images is 0.010. Figure 9 shows three watermarked images. The top row is the image with maximum distortion. Comparing to the random Works (see Figure 5), the average distortion for the images database is much lower (0.0027 for random Works) but the maximum distortion is higher (0.0052 for random Works). We do not know why the performance are different. Probably this is because natural images tend to form clusters and thus reducing the average distortion. On the other hand, a minority of the images might get too close, and require larger distortion for separation. This small cluster of images increases the maximum distortion.

Figure 10 shows selected nodes of the tree at the first, third, 6th and 8th level. These nodes are visited while searching for the top-right image in Figure 7. That is, the query image is first compared with ((a), (d)), and finally compared with ((c), (f)).

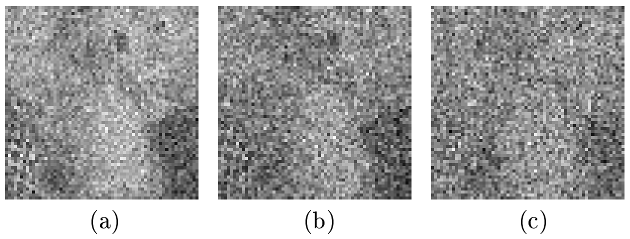

Figure 8: Three corrupted queries. The noise variance is (a) 1, (b) 2 and (c) 4 respectively. The uncorrupted image is shown in the top-right corner of Figure 7. The proposed algorithm correctly retrieves the index for (a) and (b), but not (c).

4.1 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 [5] shows that, the maximum achievable rate with distortion ϵ and robustness σ^2 is

$$C = \frac{d}{2} \log \left(1 + \frac{\epsilon}{\sigma^2} \right) . \qquad (4)$$

That is, the maximum number of messages that can be embedded is 2^C. If we employ solely watermarking to solve the message-Work association problem, with the constraint on distortion and robustness, the size of the database is bounded above by $(1 + \epsilon/\sigma^2)^{d/2}$. From the experimental data in Section 4, with robustness $\sigma^2 = 2$, dimension $d = 64^2$ and distortion 0.0027, our method can have 2048 Works. In contrast, the theoretical maximum number achievable by watermarking is $(1 + 0.0027/2)^{d/2} < 16$.

Note that the capacity in (4) does not depend on the energy of the Works. Intuitively, (4) is achieved by first quantizing the Works space into cells and uses different code books for each cell. To embed message m to a Work I, the cell I is first determined and m is then embedded using the corresponding code-book.

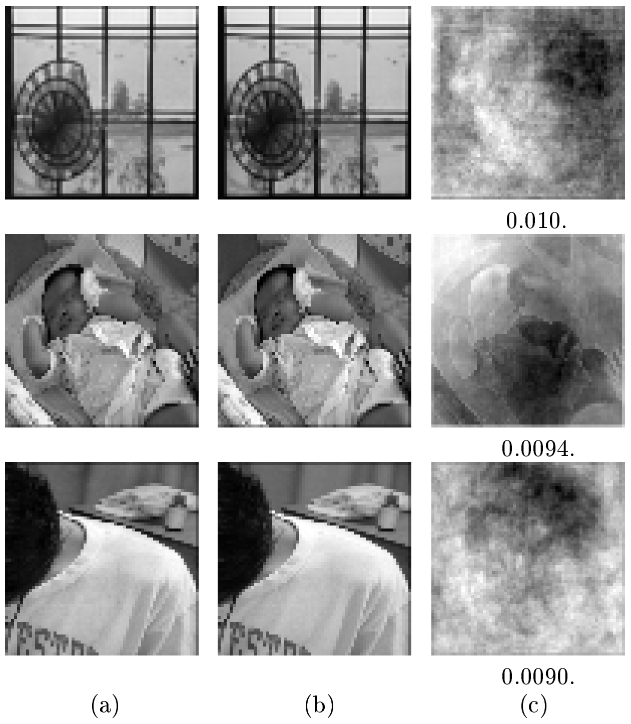

Figure 9: Images in column (a) are the original, (b) are the respective watermarked image and (c) are the difference (watermarked minus original). The images in (c) are enhanced (by scaling the intensity) for better printing quality. The values below the images are the distortion (that is, energy of (c)).

In contrast, our algorithms depends on the energy of the Works. The experiments conducted use Works and images of unit energy. The performance will increase as the energy increases (to increase the energy by a factor of A is equivalent to reduce the noise variance and distortion by the same factor). We can viewed our method as a quantization of the Work space (as in the coding), where each cell reveals partial information of the associated message m. If there is only one Work in the cell, then there is no ambiguity. If there is more than one, distortion is required. Thus, potentially, the achievable rate could be

$$C' = \frac{d}{2} \log \left(1 + \frac{S + \epsilon}{\sigma^2} \right) . \qquad (5)$$

where S is the energy of the Works.

5. VARIATIONS AND FUTURE WORKS

Toward this end, the database remains unchanged throughout the encoding and query process. It will be interesting to study the *dynamic* setting. In this setting, the database $\mathcal{I}$ starts from one Work I_1. New Works can be added into $\mathcal{I}$, but once added, cannot be removed. Furthermore, the corresponding watermarked Works must be computed before the arrival of new Works. The watermarked Work, once computed, can not be modified. The dynamic setting is

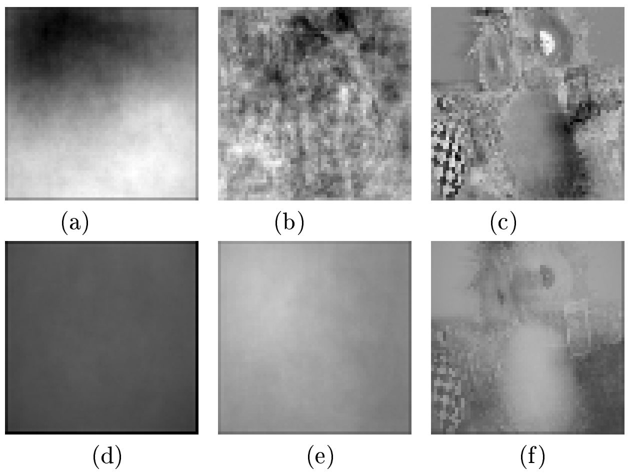

Figure 10: The normal H of the hyperplanes computed at the first, fourth, 8th level are depicted as image (a), (b) and (c). Image (d), (e) and (f) are the corresponding point C. These hyperplanes are visited while searching the query shown in Figure 9(a).

motivated by application where a stream of images are to be watermarked by a watermarking service provider before releasing to the public domain. The watermarking service provider does not know in advance the images to be watermarked, and the watermarked images, once released to the public domain, can not be recalled for modification.

Another possible research direction is to study how the size of the index tree affects watermarking performance. Most watermarking formulations (for example, watermarking with side information) assume that the encoder and decoder know the distribution of the Works, but not the actual Works. In our formulation, the encoder and decoder has access to the index tree and thus has full information of the actual database. In applications where the decoding is to be performed in the client-side, the index tree has to be sent over the network. This is practical only if the description of database is small. Thus, it is useful to know how to obtain a compact description of the database, and how to trade-off its size with other watermarking performance measures.

6. CONCLUSION

In this paper, we introduce a variant of retrieval problem where the data points can be slightly distorted. This problem is motivated by the observation that, the message-Work association typically addressed by watermarking, can also be treated as a searching problem. We give an algorithm which is a combination of watermarking techniques and clustering algorithm. This simple algorithm demonstrates that with small distortion, we can search fast, even in very high dimension. From the watermarking perspective, this algorithm demonstrates that with some searching ability, we can significantly reduce distortion and thus improve watermarking performance.

7. REFERENCES

[1] André Adelsbach and Ahmad-Reza Sadeghi. Zero-knowledge watermark detection. *4th International Workshop on Information Hiding*, LNCS 2137:273–288, 2001.

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

[3] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.

[4] Vladimir Cherkassky and Filip Mulier. *Learning from Data: Concepts, Theory, and Methods.* Wiley, New York, 1998.

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

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

[7] Digimarc. `http://www.digimarc.com/`.

[8] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* Wiley-Interscience, 2001.

[9] J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm fro finding best matches in logarithmic expected time. *ACM Trans. on Math Software (TOMS)*, (3), 1977.

[10] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–57, June 1984.

[11] Gisli R. Hjaltason and Hanan Samet. Ranking in spatial database. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–57, 1984.

[12] R. Kurniawati, J.S. Jin, and J.A. Shepherd. An efficient nearest-neighbour search while varying euclidean metrics. *ACM Multimedia*, pages 411–418, 1998.

[13] David A. White and Ramesh Jain. Similarity indexing with the ss-tree. In *Proc. 12th IEEE International Conference on Data Engineering*, Feburary 1996.