A Unified Framework for Resolving Ambiguity in Copy Detection

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

ABSTRACT

Copy detection is an important component of digital rights management and can be implemented using a retrieval-based approach. Under this approach, a query image, suspected to be a copy, is compared against all the images in the owner database. The comparison is done based on a distance metric in feature space. The performance of such a system depends on the mutual separation of the feature representation of the images in the database. In this paper we propose a framework that increases this mutual separation by literally shifting them away from each other. The idea of modifying the features derives its inspiration from the field of watermarking. It is also important to make sure that the semantics of the images do not change after modification. Thus the focus of this paper is on how to modify the images in the database, so that the mutual separation between the images in feature space is above a certain threshold and the distortion induced is minimized. This problem can be formulated as a non-convex optimization problem which is difficult to solve. We propose a restriction of the problem and solve it using second-order cone programming. We present a practical implementation of our framework, named RAM, which uses $AFMT$ as the feature representation. We conduct experiments to test the performance of RAM.

Categories and Subject Descriptors

K.6 [**Management of Computing and Information Systems**]: Miscellaneous; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Algorithms, Security

Keywords

Copy Detection, $AFMT$, Retrieval, Watermarking, non-convex optimization

MM'05, November 6–11, 2005, Singapore.

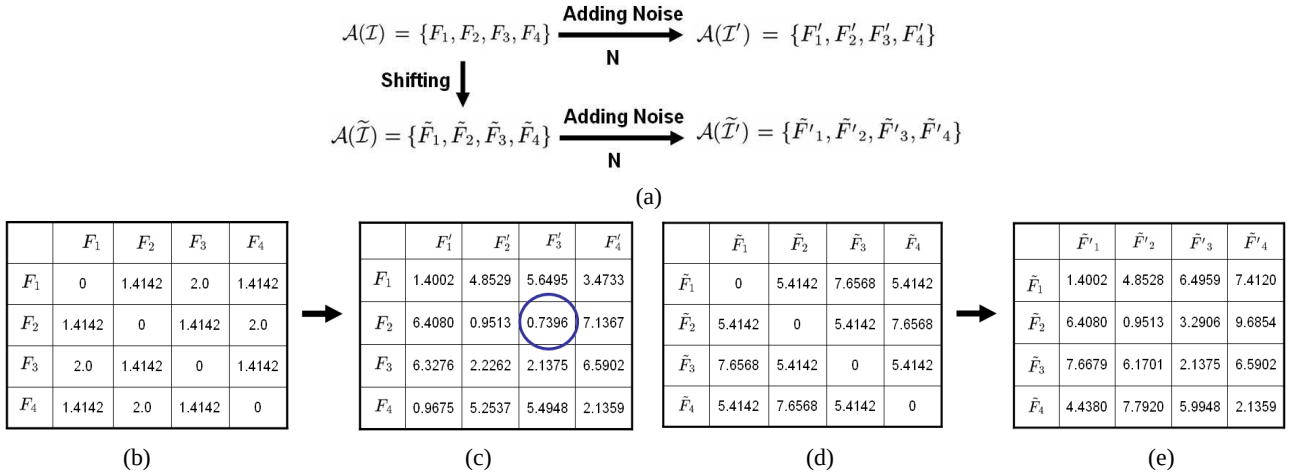
1. INTRODUCTION

Several applications like near-replica detection[15, 13], copy detection[11, 5], sub-image retrieval[13], content based image retrieval[1, 20, 9, 4] etc., use retrieval systems as the underlying framework. Typically, in a retrieval framework, a feature space is chosen and the distance of the features of the query image from the features of the images in the database is measured based on a metric. The image or a set of images near to the query in feature space, is returned as search result. To be robust against permissible manipulations or inevitable noise, an effective retrieval system typically chooses a feature space and metric, such that any two images in the database are well separated from each other. If two images are close to each other, ambiguity in detection might occur, i.e., using a slightly manipulated version of one of them as query may lead to the wrong image.

Finding such a good feature representation and metric is not easy and is an active research area. Instead of refining a known feature representation and distance metric, in this paper, we propose an alternative approach in improving the effectiveness of retrieval systems. The original images are slightly modified to increase their mutual separation in feature space above a threshold, such that, the perceptual difference between the original and the modified image is minimized. The possibility of modifying the original derives its inspiration from the field of watermarking, where the encoder embeds information into the image by modifying the original. The process of modification of the images for the purpose of improving retrieval performance can be seen as a combination of retrieval and watermarking systems. If the feature representation of the images is considered as data points in high dimensional space, our proposed approach can be expressed by the following:

Given a set of multi-dimensional data points, how to minimally shift them so that their mutual separation is above a threshold.

An application that can benefit from a solution to this problem is a copy detection system, where the emphasis is on exact detection as opposed to inexact detection (as is common in CBIR systems). In a typical copy detection scenario, an owner owns a large database of images that is made available to the public for viewing only. The owner may wish to know whether there are illegal copies of his images in the web. He could employ a web-robot, which randomly picks an image from the web, and checks whether this image is a copy of an image in his database. If a copy is found, the owner will decide about what action to carry out. Note that the illegal copy could be a modified, for example a lossy compressed, cropped, rotated, or even a maliciously altered version, modified by an attacker who is aware of the detection mechanism. Furthermore, in a scenario where the owner had sold two copies of the same image in his



	F_1	F_2	F_3	F_4
F_1	0	1.4142	2.0	1.4142
F_2	1.4142	0	1.4142	2.0
F_3	2.0	1.4142	0	1.4142
F_4	1.4142	2.0	1.4142	0

(b)

	F_1'	F_2'	F_3'	F_4'
F_1	1.4002	4.8529	5.6495	3.4733
F_2	6.4080	0.9513	0.7396	7.1367
F_3	6.3276	2.2262	2.1375	6.5902
F_4	0.9675	5.2537	5.4948	2.1359

(c)

	$\tilde{F}_1$	$\tilde{F}_2$	$\tilde{F}_3$	$\tilde{F}_4$
$\tilde{F}_1$	0	5.4142	7.6568	5.4142
$\tilde{F}_2$	5.4142	0	5.4142	7.6568
$\tilde{F}_3$	7.6568	5.4142	0	5.4142
$\tilde{F}_4$	5.4142	7.6568	5.4142	0

(d)

	$\tilde{F}'_1$	$\tilde{F}'_2$	$\tilde{F}'_3$	$\tilde{F}'_4$
$\tilde{F}_1$	1.4002	4.8528	6.4959	7.4120
$\tilde{F}_2$	6.4080	0.9513	3.2906	9.6854
$\tilde{F}_3$	7.6679	6.1701	2.1375	6.5902
$\tilde{F}_4$	4.4380	7.7920	5.9948	2.1359

(e)

Figure 1: Illustration of the ambiguity problem and its solution using the proposed framework. Here $\mathcal{A}(\mathcal{I}) = \{F_1, F_2, F_3, F_4\} = \{(3,2),(4,3),(3,4),(2,3)\}$, is the original feature database, $\mathcal{A}(\widetilde{\mathcal{I}}) = \{\tilde{F}_1, \tilde{F}_2, \tilde{F}_3, \tilde{F}_4\} = \{(3,-0.83),(6.83,3),(3,6.83),(-0.83,3)\}$ is the modified feature database and $N = \{(-1.40,0.12),(0.08,-0.95),(1.64,-1.37),(-2.0,0.75)\}$, is the noise (intensional or un-intensional manipulations) that both $\mathcal{A}(\mathcal{I})$ and $\mathcal{A}(\widetilde{\mathcal{I}})$ encounters to generate $\mathcal{A}(\mathcal{I}') = \{F_1', F_2', F_3', F_4'\}$ and $\mathcal{A}(\widetilde{\mathcal{I}}') = \{\tilde{F}'_1, \tilde{F}'_2, \tilde{F}'_3, \tilde{F}'_4\}$ respectively.

database to two different customers, he may want to identify each copy individually. This is equivalent to having multiple copies (duplicates) of the same image in the database.

In this paper, we refer to the problem of missed detection arising due to lack of separation between the images in feature space, as the *ambiguity problem*. Figure 1 gives an illustration of the ambiguity problem and also demonstrates our main idea. Given a feature database $\mathcal{A}(\mathcal{I})$, we modify it by shifting the features away from each other to generate a database $\mathcal{A}(\widetilde{\mathcal{I}})$. This is depicted in Figure 1(a). Table (b) in Figure 1 gives the pairwise distance between the elements in $\mathcal{A}(\mathcal{I})$. By adding noise N to $\mathcal{A}(\mathcal{I})$, we get $\mathcal{A}(\mathcal{I}')$ (refer Figure 1(a)). Table (b) gives the pairwise distance between $\mathcal{A}(\mathcal{I})$ and $\mathcal{A}(\mathcal{I}')$. The 3rd column in Table (b) shows that F_2 is the nearest to F_3'. Hence adding noise creates ambiguity which will lead to wrong detection when the query is F_3'. On the other hand, Table (c) gives the pairwise distance between elements in $\mathcal{A}(\widetilde{\mathcal{I}})$. Note that the mutual separation between the features has increased. On adding noise N to $\mathcal{A}(\widetilde{\mathcal{I}})$ we get $\mathcal{A}(\widetilde{\mathcal{I}}')$ (refer Figure 1(a)). Table (d) gives the pairwise distance between $\mathcal{A}(\widetilde{\mathcal{I}})$ and $\mathcal{A}(\widetilde{\mathcal{I}}')$. Note that there is no ambiguity problem as the features are still well separated even under noise N.

Outline. In Section 2, a brief review of state-of-the art approaches to implement copy detection systems is presented. This discussion leads to an explanation of the motivation behind our proposed framework. Section 3 presents our framework and formulates it as the solution to a non-convex optimization problem, which is difficult to solve. An approximate algorithm to solve it, is proposed in Section 4. A practical implementation of our framework is presented in Section 5 and the security of our framework is analyzed in Section 6.

2. RELATED WORK AND MOTIVATION

Recently there has been growing interest in copy detection for copyright protection of images [8, 11, 5, 15, 2, 13]. Most of the works highlight the importance of exact detection as opposed to inexact detection (as is common in CBIR systems). In this paper also, we address the issue of accuracy (exact detection). Existing copy detection systems can be classified into two categories or frameworks: retrieval[15, 2, 13, 5, 11] based and watermarking[8] based framework.

In a retrieval framework, the emphasis is on finding a good feature representation and distance metric. Images can be represented either by global or local features. Recent works [14, 13, 16] highlight the efficacy of local features in improving accuracy. In [14, 13] robust scale, rotation invariant descriptors are proposed. Such features however create ambiguity problem when an image has multiple similar regions or when the database consists of images of the same scene taken from different poses. In [16] a solution to this is proposed which augments SIFT [14] descriptors with a global context vector that adds curvilinear shape information for a much larger neighborhood. If the database consists of duplicate copies of the same image this solution would also fail to resolve ambiguity. Some improvements in distance metrics are proposed in [15].

In a watermarking based framework, information about the images identity is embedded into the image by modifying the image. The robustness of the system is inversely related to the degree of modification. So the emphasis here is on finding a proper trade-off between distortion and robustness. Detection speed is faster in a watermarking based framework than in retrieval framework (which suffers from high dimensionality curse). Moreover detection can be done in an off-line setting as the detector does not need to access a database. However, watermarking framework is more vulnerable to attacks and introduces distortions.

A framework combining watermarking and retrieval based frameworks have been studied before[19]. However, the focus was to achieve speedup in nearest neighborhood search by exploiting the freedom to modify the data. An active clustering approach was proposed to cluster data points in a hierarchical manner and generate an index tree.

In light of the above discussion we note that retrieval and watermarking framework both have their advantages and disadvan-

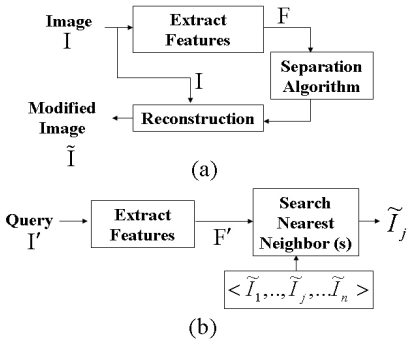


Figure 2: Illustrative block diagram of our proposed framework (a) Preprocessing stage (b) Detection stage.

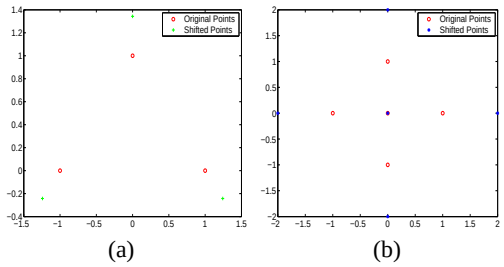


Figure 3: Illustration of the Linear Constraint Restriction Method. All points are in $\mathbf{R}^2$ and consists of (a) 3 points (b) 5 points. The value of δ is taken as 2.

tages. Some interesting observations are that: (1) robustness of retrieval systems is dependent on the robustness of the feature representation. If the same feature representation is employed by a watermarking-based and retrieval-based system, both systems would achieve the same robustness. (2) Although high dimensionality in retrieval systems hinders search speed, we note that high dimensionality means high capacity for information embedding and hence is an advantage from the watermarking perspective. (3) Watermarking based systems are less secure, as the detection routine is fixed and can not be changed after the images are watermarked, Hence, it is not easy to respond to subsequent attacks that target at the fixed watermarking method. These observations motivate us to propose a framework that uses a combination of these two frameworks (refer Section 1) and achieves a tradeoff. Note that we are not trying to improve upon a feature representation or giving an alternative method for watermarking.

3. PROPOSED FRAMEWORK

The overall framework is depicted in Figure 2. It consists of the preprocessing and detection stage. The key component is the separation algorithm, which modifies the feature vectors such that they are well separated and yet minimizes distortion. Section 4.1 will give a detailed discussion of this algorithm. The reconstruction is also an interesting side issue but we will not elaborate it in this paper.

Preprocessing Stage. Given a database of images $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$, we want to preprocess $\mathcal{I}$ to get a modified database $\widetilde{\mathcal{I}} = \{\widetilde{I}_1, \widetilde{I}_2, \ldots, \widetilde{I}_n\}$. For this, first the feature representation of the images are extracted. Let $\mathcal{A}(I)$ be the feature representation of the image I, and $\mathcal{F} = \{\mathcal{A}(I_1), \mathcal{A}(I_2), \ldots, \mathcal{A}(I_n)\} = \{F_1, F_2, \ldots, F_n\}$, denote the set of features corresponding to the image database $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$. Next, these feature vectors are significantly separated from each other using a separation algorithm to generate a modified set of features $\widetilde{\mathcal{F}} = \{\widetilde{F}_1, \widetilde{F}_2, \ldots, \widetilde{F}_n\}$.

Finally, the modified database $\widetilde{\mathcal{I}}$ is reconstructed from $\widetilde{\mathcal{F}}$. In other words, the reconstruction stage takes an image I, its modified feature $\widetilde{F}$ and finds a $\widetilde{I}$ such that $\mathcal{A}(\widetilde{I}) = \widetilde{F}$ so that $\widetilde{I}$ is close to I. The modified database $\widetilde{\mathcal{I}}$ is now ready to be released to the public.

Detection Stage. Given a query image I' we extract its feature representation F' and find the image in the modified database $\widetilde{\mathcal{I}}$ which is closest to it in the feature space. That is, the nearest neighbor in terms of ℓ_2 norm distance metric is returned. Based on the nearest neighbor(s), more elaborate tests can be conducted to determine whether the query is a copy, or we can simply decide whether it is a copy by comparing their distance with a threshold. For performance evaluation, we measure whether the system correctly output the nearest neighbor. In the work, the two main technical issues are: the reconstruction algorithm and the separation algorithm. The focus of this paper is the separation algorithm.

3.1 Reconstruction

The reconstruction algorithm depends on the choice of feature representation. For certain representations, reconstructing the image is straightforward e.g., DCT, DFT coefficients. Besides the ease of reconstruction, the choice of feature representation also depends on the type of noise to handle. Several global features, namely analytical Fourier-Mellin transform ($AFMT$) invariants [10], color histograms [23] etc, and local features, namely SIFT features [14, 12], that are robust to rotation-translation-scaling (RST), illumination variances, affine, and geometric transformations etc, have been proposed. As a proof of concept implementation, to achieve robustness against geometric distortions, we choose $AFMT$ invariants [10], which are robust to RST. One important assumption we make in our analysis is that distortion in the image space can be approximated by a proportional gaussian noise in feature space. The validity of this is experimentally verified in Section 5.1.

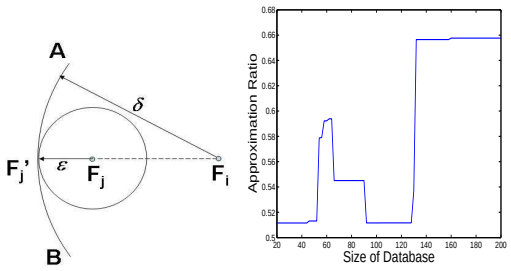
3.2 Separation

Given a set of feature vectors $\mathcal{F} = \{F_1, \ldots, F_i, \ldots, F_n\}$ where each F_i is a vector in the d-dimensional space $\mathbf{R}^d$, and a parameter δ, we want to preprocess $\mathcal{F}$ to get a set of modified feature vectors $\widetilde{\mathcal{F}} = \{\tilde{F}_1, \ldots, \tilde{F}_i, \ldots, \tilde{F}_n\}$, such that (1) the maximum distortion between $\mathcal{F}$ and $\widetilde{\mathcal{F}}$ is minimized, while (2) maintaining a minimum separation of δ between the elements in $\widetilde{\mathcal{F}}$. Specifically,

$$\begin{aligned} &\textbf{minimize} && \epsilon \\ &\textbf{subject to} && \|\tilde{F}_i - \tilde{F}_j\|_2 \geq \delta, \text{ for all } i \neq j \quad (1) \\ &&& \epsilon \geq \|\tilde{F}_i - F_i\|_2, \text{ for all } i \quad (2) \end{aligned}$$

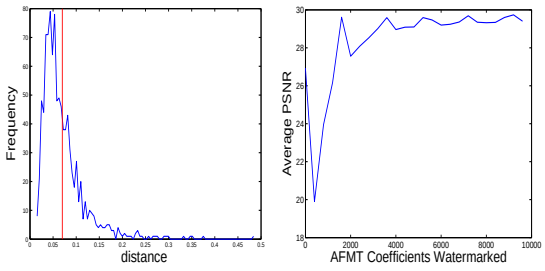
We call ϵ the *maximum distortion*, and δ the *separation*. By minimizing the maximum distortion, modification to each feature will be kept low. The constraint on separation ensures that the modified features are well separated.

Alternative objective function. The usual practice in watermarking literature is to minimize average distortion instead of minimizing the maximum distortion. So an alternative formulation for constraint (2) would be $\epsilon \geq \sum_i (\|\tilde{F}_i - F_i\|_2^2)/n$. Although we only



(a) (b)

Figure 4: (a) Geometric explanation of restriction method. (b) Simulation of the behavior of approximation ratio with change in size of database. The database consists of randomly generated 200 feature vectors of dimension 25.



(a) (b)

Figure 5: (a) Distribution of the amount by which the feature representation $\mathcal{A}(\mathcal{I})$ gets shifted when their corresponding images in $\widetilde{\mathcal{I}}$ are manipulated (rotation, scaling, painting etc). The red line indicates the mean. (b) Change in perceptual distortion with change in blocks of $AFMT$ coefficients watermarked.

consider the maximum distortion, the proposed algorithm and analysis can be adopted for average distortion also.

4. APPROXIMATE ALGORITHM

The optimization problem constraint (1) is non-convex in the sense that the solution space defined by the constraints is non-convex. Such optimization, in general, is very difficult to solve. For example, by replacing the inequality in (2) to equality, it essentially becomes a map labelling problem which is NP-hard[21]. In this section, we propose an efficient approximate algorithm by restricting the constraint. We also give two methods that achieve further speedup.

4.1 Restriction Method

We propose the following restricted formulation,

$$\textbf{minimize} \quad \epsilon$$

$$\textbf{subject to} \quad \frac{(F_i - F_j)^T}{\|F_i - F_j\|_2}(\tilde{F}_i - \tilde{F}_j) \geq \delta, \text{ for all } i \neq j \quad (3)$$

$$\epsilon \geq \|\tilde{F}_i - F_i\|_2, \text{ for all } i$$

Each quadratic constraint in (2) has now been restricted to a linear constraint. The restriction is motivated by the following observation. By the Cauchy-Schwarz's inequality,

$$\|F_i - F_j\|_2 \, \|\tilde{F}_i - \tilde{F}_j\|_2 \geq (F_i - F_j)^T \cdot (\tilde{F}_i - \tilde{F}_j) \quad (4)$$

Putting (4) into (3), we have

$$\frac{(F_i - F_j)^T}{\|F_i - F_j\|_2} \cdot (\tilde{F}_i - \tilde{F}_j) \geq \delta \Rightarrow \|\tilde{F}_i - \tilde{F}_j\|_2 \geq \delta$$

Therefore the solution space of the restricted formulation is a convex subset of the original solution space in (1). The restricted formulation (3) can be cast as a second order cone programming (SOCP) problem and has an efficient solver[22]. This kind of restriction for a hard non-convex constraint as (1) has been suggested in exercise 8.27 of [3]. For a *minimum distance constraint*, $\|\tilde{F}_i - \tilde{F}_j\|_2 \geq \delta$, the restriction is given as, $a_{ij}^T \cdot (\tilde{F}_i - \tilde{F}_j) \geq \delta$, where a_{ij} is any direction with $\|a_{ij}\|_2 = 1$. Note that this is equivalent to a projection of the vector between $\tilde{F}_i$ and $\tilde{F}_j$, onto a_{ij}. In our formulation we chose this direction to be the vector between the original data points. A geometric interpretation for this is given next.

Geometric Interpretation. Figure 4(a) gives a geometric explanation of the restriction. Suppose that a feature vector F_i is fixed and another feature vector F_j is to be shifted to F_j' so that it is δ distance away from F_i. The shifted feature vector F_j' must lie on or outside the arc AB of radius δ with F_i as the center (Figure 4). A point on the arc AB which is closest to the point F_j is the point F_j'. Clearly, the minimum shift from F_j to F_j' is along the direction $(F_j - F_i)$.

Figure 3 (a) and (b) show two examples of the approximate algorithm implemented, for points in $\mathbf{R}^2$. In both examples, the solution for the restricted formulation is indeed the optimal solution of the original problem.

To investigate the accuracy of the approximate algorithm, we perform an experiment where a few hundreds 25-dimensional vectors are randomly chosen from a multi-variate Gaussian distribution where the covariance matrix is the identity. Note that a theoretical lower bound on the optimal maximum distortion (with respect to the original optimization problem) is $\epsilon_l = (\delta - d_{min})$ where d_{min} is the minimum distance between any two vectors in the data set. Figure 4 (b) show the ratio of ϵ_l/ϵ' where ϵ' is the maximum distortion obtained under the restricted formulation. Note that we achieve constant approximation in this experiment.

4.2 Improving Scalability

If the data set consists of n vectors, each in $\mathbf{R}^d$, then the number of constraints in the above formulation is in $\Theta(n^2)$, and the number of variables is dn. For a database of images where n and d is large, the number of constraints and variables are too large for existing SOCP solvers. For example, in our experiment, the number of images is $n = 23000$ and $d = 400$. Hence we have to significantly reduce the size of the input.

Constraint Pruning. Many constraints are redundant as the feature are already far apart from each other. Note that for a particular feature vector we only need to consider its interaction with all feature vectors which are within a ball of radius $\delta + 2\epsilon_u$ around it, where ϵ_u is an upper bound of the maximum distortion ϵ. In our application, a reasonable upper bound of ϵ is δ. This is because distortion of δ will give unacceptable perceptual distortion from the original image. Hence, we can only consider feature vector pairs which are within a radius of 3δ.

Although this method significantly prunes the number of constraints, it is still not good enough for very large databases.

Dividing into subproblems. Another way of improving scalability would be to partition the feature $\mathcal{F}$ into well-separated subsets, so that we can independently modify the features in each subset. Given 2 subsets C_1 and C_2 of $\mathcal{F}$, define the distance between them as

$$d(C_1, C_2) = \min_{F_1 \in C_1, F_2 \in C_2} \{\|F_1 - F_2\|_2\}.$$

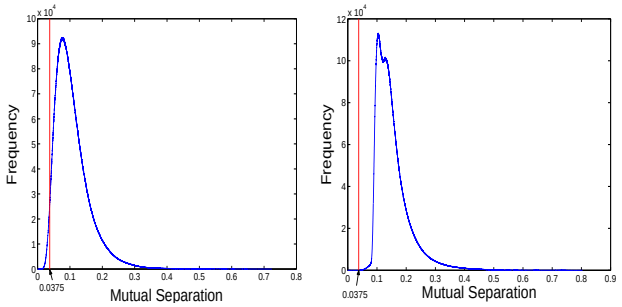
If $d(C_1, C_2) > \delta + 2\epsilon_u$ where ϵ_u is an upper bound of ϵ, then there is no interaction between C_1 and C_2. Hence, we can perform optimization on C_1 and C_2 independently and yet the solution is still same as if we consider them together. Such partitioning can be easily found by scanning the pairwise distances among the features. We can also view each feature as a vertex in a graph, wherein, there is an edge between two features F_1 and F_2 if and only if $\|F_1 - F_2\|_2 < \delta + 2\epsilon_u$. Then the partition corresponds to the different connected components in the graph.

In all our experiments, the combination of the above two methods is sufficient in reducing and dividing the optimization problem into manageable sub-problems. Note that pruning away constraints and dividing into sub-problems does not affect the optimality of the solution. That is, no approximation is being applied. Nevertheless, in cases where the above two methods fail to achieve manageable sub-problems, we can apply clustering algorithm on the feature set in that sub-problem. Features that lie on the boundary of the clusters are shifted so that the cluster are $\delta + 2\epsilon_u$ apart. This is the technique applied in [19]. However, unlike the above two methods, this is an approximation and the speedup will affect the solution.

5. IMPLEMENTATION

We have developed a proof of concept system, RAM (**R**esolving **A**mbiguity by **M**odification). The system follows the framework illustrated in Figure 2. We conducted experiments on a database of colored images from two data sets, namely, (1) COIL-100 data set[17] (7200 images of average size 128×128) and (2) Corel Image database[6] (15949 images of average size 384×256 or 256×384). The Corel database consists of natural images, few of which are duplicates. Out of 15949 images there are 65 duplicates. The COIL-100 database consists of images of 100 objects taken from different poses. As noted by Ke et. al. [13] features robust to pose changes would not fair well for near-replica detection. So it is interesting to test how the ability to modify the features helps in near-replica detection of images of the same object at different poses. This is one of the primary motivations in choosing the COIL-100 database.

Feature Representation. For invariance to color modification we extract the Y-component of the YUV representation of the images and obtain the $AFMT$ invariants of these representations. The fast algorithm as in [7] is employed to compute a two dimensional Fourier transform on the log-polar transformed image of the Y-component. Coefficients 1001 to 1400 of the $AFMT$ invariant vector is taken as the feature representation to form a set of feature vectors, $\mathcal{A}(\mathcal{I}) = \{F_1, F_2, \ldots, F_n\}$ and they correspond to the mid-frequency components. This choice was experimentally verified. We took 30 images and modified blocks of 400 coefficients by adding a random sequence to them, starting with the first coefficient and then shifting it as a sliding window from the 1^{st} to the 10000^{th} coefficient. Figure 5(b) illustrates the perceptual measure (average $PSNR$) after reconstruction. For coefficients after the 1000^{th} coefficient, the $PSNR$ between the original and reconstructed images remain almost constant. Therefore we take the 1001^{th} to 1400^{th} coefficients.



(a) (b)

Figure 6: (a) Histogram of the mutual separation between elements in $\mathcal{A}(\mathcal{I})$. (b) Histogram of the mutual separation between elements in $\mathcal{A}(\widetilde{\mathcal{I}})$. (For Corel database)

5.1 Estimating the Parameter δ

We model the manipulations on the images in the spatial domain (namely geometric transformations, cropping, painting, adding Gaussian noise, JPEG compression, brightness change, contrast change etc.) by additive Gaussian white noise in the feature domain. To verify this assumption, we experimentally estimate the distribution of $\|\mathcal{A}(I) - \mathcal{A}(I')\|_2$ where I is a randomly chosen image from the database, and I' is obtained from I by a combination of a rotation of 10^o, cropping by removing 70%, scaling down by 4 times, painting of 4, and Gaussian noise of strength 3. Such an estimated distribution is shown in Figure 5. Note that the distribution emulates a χ^2 distribution. This strongly supports the fact that the noise in the feature domain can be modelled as a Gaussian distribution. The variance of the noise due to various manipulations is used to estimate an appropriate value for δ, which makes the separation robust to manipulations.

In Figure 5 the variance of the distribution suggests the minimum mutual separation of the features so as to be robust against manipulations. Hence, we choose $\delta = 0.0375$.

5.2 Performance of our Framework

Preprocessing. For the Corel database, the minimum and maximum separation between any two feature vectors in $\mathcal{A}(\mathcal{I})$ is 0 and 0.72483. For $\delta = 0.0375$, after preprocessing using the separation algorithm in Section 4, the minimum and maximum separation between the feature vectors in $\mathcal{A}(\widetilde{\mathcal{I}})$ is 0.0375 and 1.3338 respectively, with maximum distortion $\epsilon = 0.001$. For the COIL database, the minimum and maximum separation between any two feature representations in $\mathcal{A}(\mathcal{I})$ is 0.0007355 and 0.33369 before separation and 0.0375 and 1.2433 after separation with maximum distortion $\epsilon = 0.01855$. Figure 6 shows the distribution of the mutual separation matrix before and after preprocessing of the database $\mathcal{I}$. Note that, after preprocessing all the feature vectors are at least δ separated. An illustration of the reconstruction process is given in Figure 7. The availability of the original image in the inverse log-polar transformation stage (refer Figure 2(a)) helps to get an accurate reconstruction. Note that the original and the modified image are perceptually similar.

Detection. To test the detection performance of RAM, we randomly pick 211 images in $\widetilde{\mathcal{I}}$ and manipulate each image by rotating (45^o), cropping (removing 70% about center), scaling (down x4), adding Gaussian noise (strength 3), changing contrast (x2), chang-

(a) (b) (c) (d) (e)

Figure 7: Reconstruction Process: (a) Original Image I. (b) Log-polar transform of I. (c) Reconstructed log-polar image of preprocessed $AFMT$ invariants $\mathcal{A}(\tilde{I})$. (d) Reconstructed preprocessed image $\tilde{I}$. (e) Difference between the luminance components of I and $\tilde{I}$.

ing brightness (150%), painting (x2), shearing (15% about x axis) and JPEG compressing (quality 20) them to generate 211 query images $\mathcal{I}' = \{I'_1, \ldots, I'_{211}\}$ for each category of manipulation, i.e, a total of 1899 images. The manipulations are performed using ImageMagick. Our manipulations are similar to the manipulations in [13, 15]. Next, for every query, we search for their original in $\widetilde{\mathcal{I}}$. Unlike [13, 15] that searches for the manipulated copies using the original as query, we search for the original in $\widetilde{\mathcal{I}}$ using the manipulated query and return the nearest neighbors. The results of the query are presented in the fifth column (titled "Preprocessed") of Table 1.

5.3 Comparison with Existing Framework

For fair comparison of RAM with a retrieval framework that does not do preprocessing, we consider a retrieval system that uses the 1001^{th} to 1400^{th} coefficients of the $AFMT$ invariants as the feature representation. We take 211 images from $\mathcal{I}$ and manipulate each image using the image transforms described in Section 5.2 to generate 211 query images for each category of manipulation, i.e., a total of 1899 images. The manipulation are again performed using ImageMagick. Using each manipulated image as query, we search for its original in $\mathcal{I}$ and return its $k = 10$, $k = 5$ and $k = 1$ nearest neighbors. A retrieval is considered correct if the correct copy is one of the k-nearest neighbors of the query. Columns 2, 3 and 4 of Table 1 give the detection accuracy obtained by the retrieval system by searching in the original database $\mathcal{I}$. Compared to the accuracy obtained using RAM (indicated in column 5 of Table 1), note that for a retrieval framework we do not achieve 100% detection accuracy even for $k = 10$. In our proposed framework for most cases the nearest neighbor (i.e., $k = 1$) is the query.

Figure 9 gives examples of the different kind of queries we have considered in our experiments. For all these queries we get correct results. Since we use a global feature representation, the performance of our system under manipulations like excessive cropping (say 90%) is less effective than a state-of-the-art retrieval system (for example [13]). However, our goal here is not to compare with a state-of-the-art system but to demonstrate the efficacy of our framework by giving a proof of concept implementation.

Figure 8 depicts detection results for a query into the COIL database using RAM. The purpose of this test is to analyze how the problems due to lack of uniqueness and robustness is solved by RAM. Table 2 gives a comparison of our system with a SIFT based retrieval system and an $AFMT$ based retrieval system without preprocessing. We find the nearest neighbor in all three cases, i.e., k = 1. We used the SIFT feature extraction and matching implementation made publicly available by David Lowe [14]. For the SIFT based system implementation, the image with the maximum number of "keypoint" matches is taken as the nearest neighbor. Lack

	Nearest Neighbor Accuracy			
	Without Preprocessing(%)			Preprocessed(%)
	k = 10	k = 5	k = 1	k = 1
Rotation (45^o)	100	100	95.26	100
Scaling (down x4)	98.57	98.57	91.46	100
Cropping (remove 70%)	2.84	0.47	0	71.4
Gaussian Noise (3)	96.20	94.3	84.83	100
Paint (2)	95.73	95.73	88.15	100
JPEG Compression (20)	100	100	95.26	100
Contrast (x2)	53.08	47.39	31.75	100
Brightness (150%)	77.72	70.61	60.95	100
Shear (x15)	73.45	60.67	32.7	100

Table 1: Comparison of RAM with AFMT based retrieval systems without preprocessing (RF).

	RF(%)	SIFT(%)	RAM(%)
Rotation (45^o)	50	48.33	100
Scaling(down x4)	75	36.67	98.3
Cropping (remove 70%)	0	30	68.33
Gaussian Noise (3)	33.33	40	80
Painting (2)	76.67	38.33	90
JPEG Compression (20)	96.67	70	100
Contrast (x2)	16.67	81.67	98.3
Brightness (150%)	51.67	80	100
Shear (x15)	6.67	70	100
Original	100	90	100

Table 2: Performance comparison between (a) $AFMT$ based retrieval system without preprocessing (RF), (b) SIFT based systems and (c) RAM on the COIL database (for k = 1). Total 360 queries were used.

of ability to resolve ambiguity in images of the same object taken from different poses by SIFT descriptor is indicated by the search results when we use the original as the query. $AFMT$ descriptor being a global descriptor has better discrimination ability. $AFMT$ features without preprocessing perform very poorly under cropping of 70%. Using RAM, we clearly improve upon this. SIFT features do not seem to perform well under rotation, scaling and JPEG compression, for k=1. A possible explanation for this observation is that for COIL images, the number of keypoint's is less. This is mainly because the amount of texture in these images is less. This is also one of the known problems with local descriptors. Added to that, since many of the images in the database are of the same object taken from different pose, the descriptors are very close to each other and hence are not robust under such operations. From this we can conclude that, for a copy detection system aiming at finding the nearest neighbor (k=1), this result is not good enough. We also note that the COIL images are very much sensitive to Gaussian noise. Overall we note that RAM performs significantly better

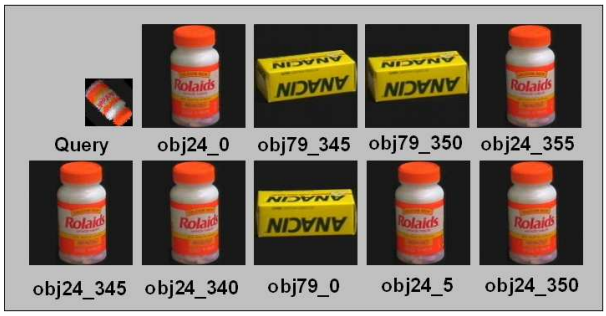


Figure 8: Result of search in COIL database: Nearest neighbors to the query arranged in decreasing order from left to right. Query is a rotated (130^o), cropped (8%), scaled down (2 times), and paint (strength 2) copy of the image obj24_0

than existing systems on a database of images of same objects taken from different poses. This is true for both cases, when the query is the original or is a manipulated copy.

Our proposed idea of selectively modifying some of the features is also an advantage over a watermarking based framework, where every image needs to be embedded with a message to identify it uniquely. In our framework, the natural separation of the images in feature space helps us to perturb only those features which are close enough and are liable to create ambiguity problem. This can be seen as a method of "watermarking with knowledge of image database", which has been shown [18] to improve the watermarking performance measures compared to a system that does not use knowledge of the database during the watermarking process.

6. AMBIGUITY ATTACKS

In this section we analyze attacks that try to create ambiguity by tampering the feature representation. Firstly we would like to highlight that the notion of perceptual similarity is a subjective measure and there is no good existing measure for it. Nevertheless, any manipulation of the image that distorts the original semantics is likely to induce distortion in the feature domain. So with this assumption, the notion of security can be measured in terms of analyzing how much $\mathcal{A}(\tilde{I})$ needs to be shifted in feature space to get $\mathcal{A}(\tilde{I}')$, so that $\mathcal{A}(\tilde{I}')$ is closer to the feature representation of another image in $\widetilde{\mathcal{I}}$.

The ability of an attacker to create ambiguities is dependent on his knowledge of the database itself. If the attacker has just one image from the database, he can add a random perturbation to its feature representation and try to create ambiguity. We assume that the attacker has full knowledge of the the database $\widetilde{\mathcal{I}}$. Hence, given any $\tilde{I}$, the attacker is able to induce minimum distortion so that the distorted image will cause ambiguity.

For our database, the average distance of an image in $\mathcal{A}(\widetilde{\mathcal{I}})$ to its nearest neighbor is 0.0726 and the distance between the closest pair is 0.0375. Thus, if an attacker has knowledge of the whole database $\widetilde{\mathcal{I}}$, given a randomly chosen image from $\widetilde{\mathcal{I}}$, he can create ambiguity by moving it towards its nearest neighbor, and the expected distortion is (0.0726/2). In the best case for the attacker, when the chosen image happens to be closest to its nearest neighbor, the attacker just has to distort the image by 0.0375/2. Figure 10(a),(b) illustrate the distortion required on a randomly chosen image, and 10(c), (b) illustrate the best case for the attacker. Note that the distortion is perceptually noticeable.

Figure 11 illustrates the nearest neighbor distance distribution for the full database, before and after preprocessing. This supports the fact that, for an attacker to create ambiguity by perturbing the feature representation of the images, it is much more easy when the images are not preprocessed. Hence RAM is more secure to malicious attacks than a scheme that does not preprocess the database.

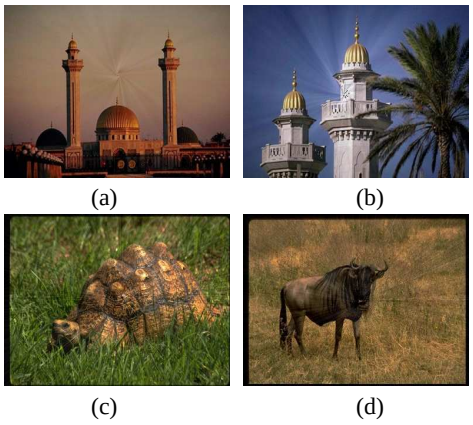


Figure 10: Reconstructed Images: (a)-(b) A randomly chosen image and its nearest neighbor, shifted towards each other by an amount half the distance between them. (c)-(d) The closest pairs in a database shifted towards each other by an amount half the distance between them.

7. DISCUSSIONS AND FUTURE WORK

We present a unified framework that resolves ambiguity by modifying the features which is applicable to any modifiable feature representation. We also present a proof of concept implementation, RAM, that uses Analytical Fourier-Mellin ($AFMT$) invariants as features. Experiments and comparison with existing frameworks show promising results. Our framework does not attempt to present a new feature representation to resolve ambiguity. It is to be applied to existing feature representations to further reduce ambiguity. Hence, it complements existing methods.

Unfortunately our framework inherits some of the limitations from watermarking and retrieval systems. (1) We need an explicit feature representation. For certain feature representations it is not clear how the reconstruction can be achieved, for example, if the feature is derived from line and shape information in the images. (2) We must have access to the database during detection. (3) It is only possible in situations where modification of the database is allowed. On the other hand, unifying both retrieval and watermark-

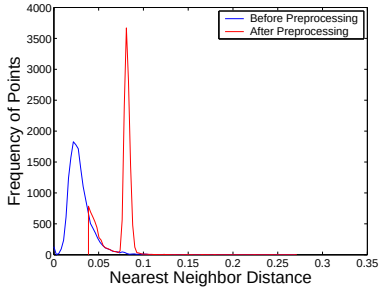


Figure 11: Comparison of nearest neighbor distance distribution before and after preprocessing.

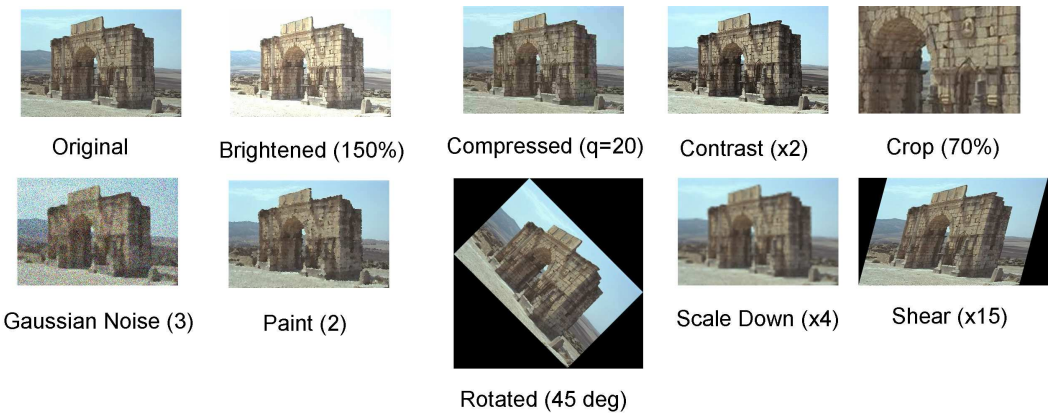


Figure 9: Examples of queries into the database. For all of them we can detect the correct original (for k=1).

ing frameworks enhances performance: (1) It further separates the images and thus reduces the chances of ambiguity. (2) It is arguably more secure. (3) It introduces less distortions compared to a watermarking based approach. In view of the pro's and con's in existing frameworks, our framework presents an alternative that complements currents methods.

The proposed framework is designed for a static database setting. For an on-line setting, the proposed framework can be extended by adding constraints to the original optimization formulation. The added constraints retain the present separation between the data points and separate the added data in relation to it. Some studies on the effect on performance in the on-line setting can be found in [18].

Acknowledgements

Sujoy Roy is partially supported by the Singapore Millennium Foundation (SMF). He would like to thank Dr. Natarajan, Department of Mathematics, National University of Singapore, for the helpful discussions on non-convex optimization problems.

8. REFERENCES

[1] A.W.M.Smeulders, M.Worring, S.Santini, A.Gupta and R.Jain. Content based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22(12), pp: 1349–1380, 2000.

[2] S. Berrani, L. Amsaleg and P. Gros. Robust Content Based Image Searches for Copyright Protection. *ACM Workshop on Multimedia Databases*, 2003.

[3] S. Boyd and L. Vandenberghe. Convex Optimization. *Cambridge University Press*, 2004.

[4] M. L. Cascia, and E. Ardizzone. JACOB: Just a content-based query system for video databases. *IEEE Int. Conference on Acoustics, Speech and Signal Processing*, May, 1996.

[5] E. Chang, J. Wang, C. Li and G. Wiederhold. RIME: A Replicated Image Detector for the World-Wide Web. *SPIE Vol. 3527*, pp. 68–67, 1998.

[6] Corel. `http://www.corel.com`.

[7] S. Derrode and F. Ghorbel. Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description. *CVIU*, Vol. 83(1), July, 2001.

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

[9] M. Flickner et. al. Query by Image and Video Content: the QBIC system. *IEEE Computer*, pp. 23–32, September, 1995.

[10] F. Ghorbel. A Complete Invariant Description for Gray Level Images by the Harmonic Analysis Approach. *Pattern recognition letters*, Vol. 15, pp 1043-1051, October, 1994.

[11] A. Hampapur and R. Bolle. Comparison of distance measures for Video copy detection. *ICME*, 2001.

[12] Y. Ke and R. Suthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. *IEEE CVPR*, 2004.

[13] Y. Ke, R. Suthankar and L. Huston. Efficient Near Duplicate Detection and Sub Image Retrieval. *ACM Multimedia Conference*, 2004.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Vol. 60(2), pp. 91–110, 2004.

[15] Y. Meng, E. Chang and B. Li. Enhancing DPF for Near-replica Image Recognition. *IEEE CVPR*, 2003.

[16] E. Mortensen, H. Deng and L. Shapiro. A SIFT Descriptor with Global Context. *IEEE CVPR*, 2005.

[17] S. A. Nene, S. K. Nayar and H. Murase. Columbia Object Image Library (COIL-100). *Technical Report CUCS-006-96*, February, 1996.

[18] S. Roy and E-C. Chang. Watermarking with knowledge of image database. *ICIP*, 2003.

[19] S. Roy and E-C. Chang. Watermarking with Retrieval Systems. *ACM Multimedia Systems Journal*, Vol. 9(5), pp. 433–440, 2004.

[20] J. R. Smith and S. F. Chang. VisualSEEK: a fully automated content-based image query system. *ACM Multimedia Conference*, November, 1996.

[21] T. Strijk and A. Wolff. Labeling points with circles. *International Journal of Computational Geometry and Applications*, Vol. 11(2), pp. 181–195, 2001.

[22] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization Methods and Software. *Special issue on Interior Point Methods 11-12*, pp. 625–653, 1999.

[23] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, Vol. 7(1), pp. 11–13, 1991.