A LEARNING FRAMEWORK FOR ROBUST HASHING OF FACE IMAGES

by

Kamil Şenel

B.S., Electrical and Electronics Engineering, Bogazici University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2010

ACKNOWLEDGEMENTS

I would like to begin by thanking my family, their support and endless love kept me going through hard times. Also a special thanks to my brother Can, who has given me the most interesting ideas during this thesis. I am blessed to have them in my life.

I am indepted to my advisor, Assoc. Prof. Kıvanç Mıhçak. He has been very patient and supportive throughout this process. He is a deeply committed teacher, researcher and advisor.

A very special thanks to Yağmur Denizhan. She has influenced not only my graduate studies, but my whole life. She has given me a strong sense of discipline and integrity, for which I am eternally grateful.

A thanks to Doğan Can, who has been a good friend for a very long time. He has encouraged me to realize my potential. He has always been there for me when I needed and I thank him for that.

ABSTRACT

A LEARNING FRAMEWORK FOR ROBUST HASHING OF FACE IMAGES

Robust image hashing has been actively researched over the last decade with varied applications in image content authentication and identification under distortions. In the existing literature on robust image hashing, hash algorithms are ignorant of the class of images being hashed. There are however significant application domains such as that of face image hashing where a-priori knowledge of the image class as well as permissible distortions can benefit hash algorithm design. In this thesis, we present a two stage cascade of dimensionality reduction constructs for face image hashing. The first stage aims to project the face image to a space where geometric distortions manifest approximately as additive noise. For this purpose, we use the non-negative matrix approximations based hash vector developed by Monga et al. which is known to possess excellent geometric attack robustness. In the second stage, we employ oriented principal component analysis (OPCA) based on estimating signal as well as noise statistics in a learning phase and deriving a projection that mitigates the effect of noise. Experimental results in the form of ROC curves(where available) show that incorporating such a learning phase greatly reduces error probabilities.

ÖZET

YÜZ İMGELERİNİN GÜRBÜZ KIYIMI İÇİN ÖĞRENME YAPISI

Son dönemde gürbüz imge kıyımı yoğun olarak imge içerik uygunluğu ve tanılama alanlarında araştırılmaktadır. Bugüne kadar yapılan arastırmalarda kıyım teknikleri uygulandıkları imgelerden bağımsız olarak geliştirilmiştir. Bununla beraber yüz imge kıyımı gibi önemli uygulama alanlarında imgeler hakkında önsel bilgi çoğunlukla mevcuttur. Bu çalışmada, yüz imge kıyımı için iki aşamali art arda boyut sayısı azaltma tekniği sunulmaktadır. İlk aşama, yüz imgesi üzerindeki geometrik bozulumların yaklaşık olarak eklenebilir gürültü olarak ortaya koyulduğu bir uzaya izdüşümü bulunmasını amaçlamaktadır. Bu amaçla, Monga tarafından geliştirilmiş negatif olmayan matris yaklaşıklamaları tabanlı, geometrik ataklara dayanıklı kıyım vektörü kullanılmıştır. İkinci aşamada, öğrenme fazı ve sonrasında türetilen, gürültünün etkilerini azaltan izdüşüm ile sinyal ve aynı zamanda gürültü istatistikleri kestirimi tabanlı yönlendirilmiş ana bileşen analizi uygulanmaktadır. Alıcı işletim eğrisi şeklinde verilmiş deneysel sonuçlar göstermektedir ki, böyle bir öğrenme fazını dahil etmek hata olasılıklarını oldukça düşürmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

$\mathbf{C}$	Diagonal matrix of $\mathbf{c}$
c_k	Indicator value at time k
e	State noise
$\exp(\cdot)$	Exponential
$\mathbf{g}$	Vector representation of the Gaussian filter
G	Gaussian filter in frequency domain
$\mathbf{G_s}$	Convolution matrix corresponding to $\mathbf{g}$ when convolved with $\mathbf{s}$
$\mathbf{G_h}$	Convolution matrix corresponding to $\mathbf{g}$ when convolved with $\mathbf{h}$
$\mathbf{h}$	Vector representation of h
$\mathbf{H}$	Convolution matrix corresponding to $\mathbf{h}$
$\mathbf{I}$	Identity matrix
KL	Kullback-Leibler distance
$\log(\cdot)$	Natural logarithm
$\mathbf{m}$	Gaussian mean
N	Number of samples
$\mathcal{N}$	Gaussian distribution
$p(\cdot)$	Probability density function
$p(\cdot\|\cdot)$	Conditional probability density function
$\mathbf{r}$	Vector presentation of r
$\mathbf{R}$	The matrix of autoregressive coefficients
$\mathbf{s}$	Vector representation of s
$\mathbf{S}$	Convolution matrix corresponding to $\mathbf{s}$
$s * h$	Convolution of s and h
T	Data length
T	Transition probability
$tr(\cdot)$	Trace
$\mathbf{U}$	State noise matrix

v	Observation noise component
$\mathbf{v}$	Vector representation of v
$\mathbf{w}$	Vector representation of w
x	Latent variable
$x \sim p(x)$	x comes from $p(x)$
$x^{(i)}$	$i^{(th)}$ sample for x
$\hat{x}$	Estimated value for x
$\mathbf{y}$	Vector representation of y
$\mathbf{z}$	Vector representation of z
$\mathcal{Z}$	Normalizing constant
$\mathbf{\Sigma}$	Gaussian covariance
Θ	Parameter set
$\lvert \cdot \rvert$	Determinant
$[\cdot]_{i,j}$	i, j^{th} element of the matrix
$[\cdot]_i$	i^{th} element of the vector
$\approx$	Approximately equal to
$\langle \cdot \rangle_{p(\cdot)}$	Expectation under p
$[\cdot]^\top$	Matrix transpose
$[\cdot]^{-1}$	Matrix inversion
$\propto$	Proportional to
$\equiv$	Definition

GED	Generalized eigenvalue decomposition
ID	Identity
i.i.d	Independently and identically distributed
KL	Kullback-Leibler
MD	Message digest
MDA	Multiple discriminant analysis
MSE	Mean-squared error
NMF	Non-negative matrix factorization
OPCA	Oriented principal component analysis

PCA	Principal component analysis
ROC	Receiver operation characteristics
RSA	Rivest-Shamir-Adleman
SHA	Secure hash application
SNR	Signal-to-noise ratio
SQ	Statistics quantization
SVD	Singular value decomposition
QR	QR-decomposition

1. INTRODUCTION

The goal of a robust signal hashing is to map a signal to a short binary string based on its content. The term "hash" derives from cryptographic hashes which map large digital messages to much smaller strings. Hashing is also referred as "passive content fingerprinting" in some circles especially in audio hashing. Robustness is the property that under small manipulations of the data the hash output stays invariant, and changes significantly only under content changes. Basically a robust signal hashing algorithm should be invariant to incidental modifications and sensitive to malicious modifications. This differentiates robust signal hashing from traditional cryptographic hashes, such as SHA-1 and MD-5 [1], which are extremely sensitive to input data.

A perceptual image hash function would facilitate comparison and searches in large databases in which perceptually identical versions of an image may exist. Furthermore such a need for image descriptors arises for the purpose of integrity verification. Because of the increasing amount of digital media and their easy-to-copy nature, digital data can be modified with ease, there arises the need to verify the content of media to ensure its authenticity. In the literature there are generally two different kind of approaches to this problem namely, digital signature based(hashing) and watermark based[2][3]. A digital signature is a set of features extracted from the media that sufficiently represents the content of the original media. Watermarking, on the other hand, is a media authentication/protection technique that embeds invisible (or inaudible) information into the media. The main difference between a watermark and digital signature is that embedding procedure of the former requires the content to be changed.

There are different robust image hashing techniques, based on image statistics [4][5][6], relations[7],preservation of coarse image representation[8] and low-level image extraction[9][10]. There are many different techniques for audio hashing as well, such as Fourier coefficient based algorithms[11]. Burges et al.[12] proposed an algorithm that chooses noise robust features using distortion discriminant analysis. The common goal

of all hashing algorithms is to find the features that represents perceptual similarity.

Face images form a subclass of images with a smaller variability then all natural images. They can be represented more effectively using class-specific hashing algorithms. We propose an algorithm that is designed for robust hashing of face images that uses 2-stage NMF hashing as a starting point. Using the similarity of face images, it is easier to model the noise on images and resulting hash outputs. Using this fact we use the method of OPCA to choose the projections that maximizes signal-to-noise ratio. Experiments show that this method outperforms existing robust image hashing algorithms developed for this task. Although NMF-NMF hashing is used in this thesis, this method proposed can be extended to other hashing algorithms as well.

The problem focused in thesis is a robust face hashing problem. In biometric hashing, robust face hashing aims to recognize the same person's photos taken under different situations, conditions, different backgrounds and so on. This recognition problem is different and more difficult problem than what we focus on this thesis. We reduce the problem to recognition of the same photo taken and possibly modified version of this photo.

The problem with robust signal hashing comes from the fact that perceptual similarity is not transitive. That is to say perceptual similarity of a pair of signals X and Y and another pair of signals Y and Z does not necessarily imply the perceptual similarity of X and Z. Note that "transitive property" implies that there exist some universal features that are used to decide on perceptual similarity. The "perceptual similarity" term contains an ambiguity itself. For some cases, this ambiguity can be resolved by the requirement of the application that the hashing function is going to be used. But for the most cases it remains as a subjective measure which makes the problem much harder. As a consequence, finding a pre-defined feature set for modeling perceptual similarity is a hard task if possible at all. How do we decide on perceptual similarity or what is the measure for perceptual similarity? These questions are yet to be answered. Trying to model human perception is the task at hand and seems impossible for now. However, by adequate modeling one can find an algorithm that

gives similar outputs for perceptually similar signals.

There are not many learning based hashing techniques in the literature. However, learning based algorithms are frequently used for indexing and searching applications [13][14]. There are also different learning based approaches in object recognition[15][16]. In this thesis, we confine ourselves to a set of images and also a set of attacks. To utilize this property a learning based hashing algorithm is proposed. Although we only consider 2-stage NMF hashing in this thesis, this approach can be applied to any hashing algorithm under similar setups. We will consider 3 different approaches, projecting to random directions, to directions chosen by PCA and OPCA methods. Projecting in the directions chosen by OPCA gives the best results.

There 3 different algorithms considered in this thesis, namely NMF-NMF-SQ, NMF-NMF-PCA and NMF-NMF-OPCA. The first approach NMF-NMF-SQ is represented in [17] and it uses random projection directions to obtain final hash vector. But for a given database and given set of attacks, using random directions for projection can be improved. For this purpose using NMF-NMF-PCA projects the hash vectors to dimensions where signal variance is maximized and NMF-NMF-OPCA projects to dimensions similar to PCA case, with the difference projected dimensions are oriented towards the least favorable dimensions of the noise.

2. PROBLEM STATEMENT, THEORY AND BACKGROUND

In this section, our application scenario is described in detail and the requirement for a robust hashing algorithms is discussed. After that, PCA, OCPA and NMF-based hashing techniques are explained. Also a detailed background is given on non-negative matrix factorization is given as well as some update rules for this factorization. Finally the proposed algorithm is presented.

2.1. Problem Statement

The robust face hashing algorithm is proposed for the following case. Consider a security system where each employee (agent, user ...etc) uses a card to access secured areas. These secured areas may be rooms, schools, work places, any area that needs authentication of the person that want to access. The card contains various information of the person that carries it. The name, an ID (school ID, worker ID, etc.) and different information along with a photo of the user and also an embedded N-length information vector. This information vector may also contain extra bits that represents private key for the user. The function of the private(secret) key will become more apparent in the section where NMF-Based Hashing algorithm is described, section 2.4. At the entrance of secured areas there will be a security check to decide whether the user is authorized to pass or not. In this scenario, a scanner will capture the image printed on the card, then compute its information-vector and compare it with the embedded information-vector. Note that in this application case the same image is considered, not images of the same person taken under different circumstances. To summarize, we want to find an algorithm that compares a previously obtained hash output with the hash output of the current image and decides whether to authorize or not. Also we focus only on image hashing part, text hashing and other possible security algorithms are not our concern.

In this thesis, bold faced letters e.g., $\mathbf{x}, \mathbf{y}$ are used to represent vectors. Upper-

case letters denotes matrices, e.g. W, V. $\mathcal{R}$ and $\mathcal{R}^d$ denote set of real numbers and the d-dimensional real vector space respectively. $\|\mathbf{x}\|$ represents L_2 norm and $< \mathbf{x}, \mathbf{y} >$ denotes inner product for vectors $\mathbf{x}, \mathbf{y} \in \mathcal{R}^d$.

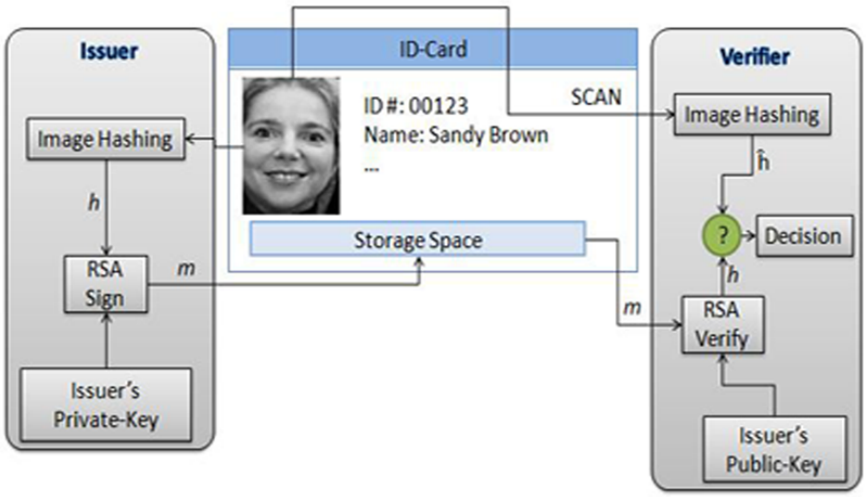

Figure 2.1. Block diagram showing application scenario

Figure 2.1 shows a block diagram of the application scenario. Here $\mathbf{h}$ denotes the original images hash output and $\hat{\mathbf{h}}$ denotes the hash output of the query image. For fairness secret key is not considered while comparing proposed algorithm with other known algorithms. RSA algorithm is a technique that can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers[18]. Also note that text-hashing part is not shown in the 2.1, but it can be added as another algorithm for verification process. But in this thesis, we only focus on image hashing part of this problem.

The information vector will be obtained from the image of the user. For this system to work, for each user's image, there should be a mapping function, that maps images to a vector in an N-dimensional space. Now if denote the input image with I, the function with $H(.)$ and the output vector with $\mathbf{h}$, then $\mathbf{h} = H(I)$ from a set $\{0, 1\}^N$ with 2^N cardinality. The output vector should be equal for images that are perceptually identical and two different images should produce uncorrelated values. Note that perceptually identical images do not necessarily have same digital representation. Since perceptually identical images need to produce same output vector, the function $H(.)$ is a many-to-one mapping. Desired properties for such a function are:

(i) Randomization : For any input image I, the output vector should be uniformly distributed among all possible outputs. That is to say for an N-length output vector:

$$\forall \mathbf{h} \in \{0,1\}^N,\ Pr\{H(I) = \mathbf{h}\} \approx 2^{-N} \tag{1.1}$$

(ii) Pairwise Independence : The output of two different input images, should be independent,

$$\forall \mathbf{h}_1, \mathbf{h}_2 \in \{0,1\}^N, Pr\{H(I_1) = \mathbf{h}_1 | H(I_2) = \mathbf{h}_2\} \approx Pr\{H(I_1) = \mathbf{h}_1\} \approx 2^{-N} \tag{1.2}$$

(iii) Robustness : The output of the function should remain invariant under all possible acceptable disturbances. For input images I and $\hat{I}$,

$$Pr\{d(H(I) = \mathbf{h}, H(\hat{I}) = \hat{\mathbf{h}}) > \tau\} < \theta \tag{1.3}$$

In this thesis, different hash methods will be considered as $H(.)$ function. The output of these hash functions result in information vectors or hash-vectors. Before going into details of hashing functions, consider the possible errors of the described security system. We will be assuming no use of private key unless otherwise stated. The possible error cases are:

(i) User's image is given as an input to the hash function, the output hash-vector does not match the hash-vector embedded on the card when it should match. This error is defined as *miss error*. It occurs when two images of the same user results in different hash-vectors. In application sense, the images considered are classified as not same images. This is due to the fact that there will be some modifications or differences on the original image and these modifications are considered as attacks or disturbances. We can think of this modifications as wear and tear effects. A robust algorithm should map these slightly different or perceptually identical images to the same output hashes, or close enough hashes to the original image's hash. In practical applications these outputs are compared

with a threshold value. Define original image as I, the hash function as $H(.)$, disturbed image image as $\hat{I}$ and threshold as τ then the *miss error* occurs when,

$$d(H(I), H(\hat{I})) > \tau \tag{1.4}$$

where $d(.,.)$ is a distance metric. As threshold value, τ increases this probability decreases.

(ii) A different person's image (different from the image that has its hash-vector embedded on the card) is given as an input to the hash function. When these two different images results in same or close enough hash-vectors, this is considered as an error. This error is defined as *false alarm error*. It occurs when two different images result in the same or close enough output vectors for a given threshold and thus classified as same images. Define 2 perceptually different images as I_i and I_j where $i \neq j$, threshold τ and hash function as $H(.)$ then, *false alarm error* occurs when,

$$d(H(I_i), H(\hat{I}_j)) < \tau \tag{1.5}$$

Note that the value of threshold τ effects the outcome of the algorithm. As τ increases number of miss errors decreases and as τ decreases number of false alarm errors decreases. There is a trade-off with choice of τ. For different application this threshold value can be chosen to maintain a desired probability of miss or false alarm error.

The techniques described later in this chapter are compared according to probability of false alarm and miss curves. This error probabilities are plotted on a ROC (Receiver Operating Characteristics) curve. Throughout this thesis comparisons are made according to ROC curves obtained from different techniques under different modifications. Only exception for using ROC curve is for the cases where ROC curve is not available because two classes are separable for some threshold τ . As can be seen in section 3, for the cases where a secret key usage is viable, we obtain distinct distance groups, which means that there exist threshold value(s) which will result in no errors.

Since it is not possible to give an ROC curve when there is no error instead we will be presenting figures that shows histograms of distances between an image and its attack version versus two distinct images.

2.2. Principal Component Analysis (PCA)

PCA is a useful statistical technique that has found applications in many fields such as face recognition and image compression, and is a commonly used technique for finding patterns in data of high dimensions[19]. PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system in such a way that the greatest variance by any projection of the data comes to lie on the first coordinate called first principal component, the second greatest variance on second second coordinate and so on. PCA is theoretically the optimum transform for given data in least square terms. That is to say, PCA is the optimum linear dimensionality reduction technique with respect to mean squared error(MSE) of the reconstruction for a given set. Given a set of M training vectors $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ of length N, the steps of the PCA is as follows:

(i) Compute empirical mean μ and covariance matrix $\mathcal{C}$ of the training set

$$\mu = \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_i \tag{2.6}$$

$$\mathcal{C} = \frac{1}{M} \sum_{i=1}^{M} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T \tag{2.7}$$

(ii) Compute eigenvalues λ_j's and eigenvectors $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$

(iii) Define a projection matrix E composed of K eigenvectors of $\mathcal{C}$ with highest eigenvalues

$$E = [\mathbf{v}_1, \ldots, \mathbf{v}_K] \tag{2.8}$$

where $\mathbf{v}_1$ corresponds to eigenvector of $\mathcal{C}$ with highest eigenvalue.

(iv) The K-dimensional representation of original N-dimensional vector is given by,

$$\mathbf{y} = E^T(\mathbf{x} - \mu) \tag{2.9}$$

Principal components can be thought in the following way, for a given set of points in Euclidean space, the first principal component(the eigenvector with the largest eigenvalue) corresponds to a line that passes through the mean and minimizes the sum squared error with those points. The second principal component corresponds to the same concept after all correlation with the first principal component has been subtracted out from the points. Each eigenvalue indicates the portion of the variance that is correlated with each eigenvector. Thus, the sum of all the eigenvalues is equal to the sum squared distance of the points with their mean divided by the number of dimensions. PCA essentially rotates the set of points around their mean in order to align with the first few principal components. This moves as much of the variance as possible (using a linear transformation) into the first few dimensions. The values in the remaining dimensions, therefore, tend to be highly correlated and may be dropped with minimal loss of information. PCA is often used in this manner for dimensionality reduction. PCA has the distinction of being the optimal linear transformation for keeping the subspace that has largest variance.

In terms of energy preservation, PCA method gives the directions that maximize the signal variance in a descending order. Basically when we project a vector using directions obtained via PCA, the resulting vector will be projected along maximum energy preserving direction. As a result the output of the PCA algorithm is optimum with respect to mean squared error of the reconstruction. Note that the idea behind hashing is not reconstruction, however PCA is widely used for hashing applications. Also even though PCA has a training phase, this method does not take noise statistics into consideration. Only signal statistics are considered.

PCA is a non-parametric analysis technique and the resulting output is unique and independent of any hypothesis about data distribution. PCA method has its

limitations due to assumptions made in its derivation. These assumptions are:

(i) Assumption on linearity. The data set considered is assumed to be linear combination of certain basis.

(ii) Assumption on statistical importance of mean and covariance(variance). PCA uses the eigenvectors of the covariance matrix and it only finds the independent axes of the data under the Gaussian assumption. For non-Gaussian or multimodal Gaussian data, PCA simply de-correlates the axes. When PCA is used for clustering, its main limitation is that it does not account for class separability since it makes no use of the class label of the feature vector. There is no guarantee that the directions of maximum variance will contain good features for discrimination. This property limits PCA in classification and hashing applications.

(iii) Assumption that large variances have important dynamics. PCA simply performs a coordinate rotation that aligns the transformed axes with the directions of maximum variance. It is only true when the observed data has a high signal-to-noise ratio that the principal components with larger variance correspond to interesting dynamics and lower ones correspond to noise.

2.3. Oriented Principal Component Analysis (OPCA)

Oriented PCA or OPCA is a term introduced by Kung and Diamantaras [20] as a generalization of the widely used PCA. It corresponds to the generalized eigenvalue decomposition (GED) of a pair of covariance matrices in the same way that the PCA corresponds to the eigenvalue decomposition of a single covariance matrix. In hashing Oriented Principal Component Analysis can be used as a linear discriminant analysis technique, where projection directions are chosen in order to maximize signal-to-noise ratio[12]. Given a set of M training vectors $X = \{\mathbf{x}_i, \ldots, \mathbf{x}_M\}$,where *each* $\mathbf{x}_i \in \mathcal{R}^d$ and N distorted versions $\tilde{\mathbf{x}}_i^k,\ k = 1, \ldots, N$, the algorithm is as follows:

(i) Compute difference vectors $\mathbf{z}_i^k$

$$\mathbf{z}_i^k \equiv \tilde{\mathbf{x}}_i^k - \mathbf{x}_i \tag{3.10}$$

(ii) Given μ computed using (2.6), compute the empirical covariance matrix $\mathcal{C}$ of the original training vectors

$$\mathcal{C} \equiv \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T \tag{3.11}$$

(iii) Compute empirical correlation matrix $\mathcal{R}$ of the noise

$$\mathcal{R} \equiv \frac{1}{MN}\sum_{i,\ k}\mathbf{z}_i^k(\mathbf{z}_i^k)^T \tag{3.12}$$

(iv) OPCA directions are defined as the directions $\mathbf{n}_i$'s that maximize generalized Rayleigh quotient

$$q = \frac{\mathbf{n}^T\mathcal{C}\mathbf{n}}{\mathbf{n}^T\mathcal{R}\mathbf{n}} \tag{3.13}$$

(v) Direction vectors can be found by setting $\nabla q = 0$, which will result in a generalized eigenvalue problem

$$\mathcal{C}\mathbf{n} = q\mathcal{R}\mathbf{n} \tag{3.14}$$

The eigenvalue of each solution gives the signal-to-noise ratio obtained when the input is projected along the corresponding eigenvector. For positive semi-definite $\mathcal{C}, \mathcal{R}$, the generalized eigenvalues are positive. By definition of $\mathcal{C}, \mathcal{R}$ it is the case here. $\mathbf{n}_i$'s are, or can be chosen to be, linearly independent q is maximized by choosing $\mathbf{n}$ to be eigenvector corresponding to the maximum eigenvalue.

(vi) Define a projection matrix $E = [\mathbf{n}_i, \ldots, \mathbf{n}_K]$, then K-dimensional representation of original d-dimensional vector is given by

$$\mathbf{y} = E^T(\mathbf{x} - \mu) \tag{3.15}$$

A feature extractor set of $\mathbf{n}_i$'s that minimizes MSE of the reconstruction can be found by taking eigenvectors of $R_1 - R_2$ with largest eigenvalues, where R_1 and R_2 are the

correlation matrices of $\mathbf{x}_i$'s and $\mathbf{z}_i$'s respectively. However a scaling of noise and original vectors with different scaling factors will result in a change of set of $\mathbf{n}_i$'s for this case. Finding directions as described in (3.13) will lead to directions that are invariant under scaling of the noise or original data, although eigenvalues will change.

Definition of correlation matrix $\mathcal{R}$ in (3.12) aims to penalize the mean noise signal as well as its variance. In a scenario where we have a noise that has zero variance but non-zero mean, directions are still chosen orthogonal to the mean vector.

OPCA can be related to Multiple Discriminant Analysis(MDA) which is a dimensionality reduction method that uses projection directions that maximize variance between classes and minimize variance within classes. The difference comes from the denominator of the Rayleigh quotient. Here the correlation matrix of the noise vectors are used. The reason for using correlation matrix for noise is that non-zero mean, zero variance noise is also penalized using correlation matrix.

2.4. Non-Negative Matrix Factorization Based Hashing

Hashing via Non-Negative Matrix Factorization is a robust hashing scheme[17], that outperforms existing robust image hashing algorithms. Without going into practical aspects of this algorithm a background and theory of the algorithm as well as a detailed discussion on update rules used is given in this section. Finally the algorithm itself is explained in detail. Detection theoretic analysis of proposed algorithm can be found in Appendix.

2.4.1. Background and Theory of Non-Negative Matrix Factorization

Consider the problem, given a non-negative matrix V, where $V \in R^{m \times n}$, find non-negative matrix factors W and H such that:

$$V \approx WH \tag{4.16}$$

where $W \in R^{m \times r}$, $H \in R^{r \times n}$ and r denotes *rank* of factorization. This factorization reduces storage amount when *rank* r is $< \frac{mn}{m+n}$.

Equation (4.16) can be rewritten column by column as $\mathbf{v} \approx W\mathbf{h}$, where $\mathbf{v}$ and $\mathbf{h}$ are corresponding columns of V and H. That is to say columns of V can be approximated by columns of W using components of $\mathbf{h}$ as weight coefficients. W can be thought as matrix containing basis vectors that are optimized for approximating V. Since r is usually chosen such that $r \ll min(m, n)$, good approximation can only be achieved when basis vectors finds structure that is latent in V.

2.4.2. Update Strategies For Non-Negative Matrix Factorization

To find an approximate factorization, $V \approx WH$ we first need to define cost functions that quantify the quality of approximations. A cost function can be defined by using some distance measure between two non-negative matrices. Euclidean distance is a popular cost function defined as,

$$\|A - B\|^2 = \sum_{ij} (A_{ij} - B_{ij})^2 \tag{4.17}$$

Another commonly used distance measure is,

$$D(A\|B) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right) \tag{4.18}$$

Equation (4.18) is known as Kullback-Leibler(KL) divergence. For the cases where $\sum_{ij} A_{ij} = \sum_{ij} B_{ij} = 1$, it reduces to relative entropy, so that matrices A and B can be regarded as normalized probability distributions.

Theorem 1. *The Euclidean Distance* $\|V - WH\|$ *is non-increasing under the update rules*

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}} \; W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}} \tag{4.19}$$

For the cases where W and H are at a stationary point of distance, Euclidean Distance is invariant.

Theorem 2. *The divergence $D(V\|WH)$ is non-increasing under the update rules*

$$H_{a\mu} \leftarrow H_{a\mu}\frac{\sum_i W_{ia}V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}} \; W_{ia} \leftarrow W_{ia}\frac{\sum_\mu H_{a\mu}V_{i\mu}/(WH)_{i\mu}}{\sum_v H_{av}} \tag{4.20}$$

For the cases where W and H are at a stationary point of distance, KL-Divergence is invariant.

Proofs of these theorems are given in Appendix. For now, it may be noted that the updates are multiplicative and the multiplicative factor is unity when $V = WH$, so perfect reconstruction is necessarily a fixed point in these update rules.

2.4.3. NMF-NMF Hashing

Hash function takes 2 inputs, an Image I and a secret key K and produces a short vector $\overrightarrow{h} = H_K(I)$ as output. The resulting output vector $\overrightarrow{h}$ is from set $\{0,1\}^N$ with 2^N cardinality. The details of the algorithm is as follows:

(i) Given an Image I, pseudo-randomly select p sub-images $A_i \in R^{m\times m}$, $1 \leq i \leq p$

(ii) Obtain a r_1 NMF from each sub-image ($r_1 \ll m$)

$$A_i \approx W_i F_i^T \; where, \; W_i, F_i \in R^{m\times r_1} \tag{4.21}$$

(iii) Pseudo-randomly arrange these matrices to obtain a secondary image J of size $m \times 2pr_1$.

(iv) Re-apply NMF to obtain a rank r_2 representation of J , $r_2 \ll min(m, 2pr_1)$

$$J \approx WH \; where, \; W \in R^{m\times r_2} \; and \; H \in R^{r_2\times 2pr_1} \tag{4.22}$$

(v) NMF hash vector $\mathbf{h}_K^{NMF-NMF}(I)$ of length N is produced by concatenating

columns of W and rows of R.

(vi) As a final step, generate pseudo-random weight vectors $\{\mathbf{t}_i\}_{i=1}^{M}$ $(with\ M \ll N)$, such that *each* $\mathbf{t}_i \in R^N$. The resulting hash vector of length M is given by $\{< \mathbf{h}_K^{NMF-NMF}(I), \mathbf{t}_1 >, \ldots, < \mathbf{h}_K^{NMF-NMF}(I), \mathbf{t}_M >\}$, where $< \mathbf{x}, \mathbf{y} >$ denotes inner product of vectors $\mathbf{x}$ and $\mathbf{y}$.

Design of weight vectors, such that perceptual qualities of the hash are retained, under highly-correlated noise would be a hard task. Here, the property that the noise on NMF-NMF-SQ hash vectors under attacks is i.i.d. gives us advantage[17]. Weight vectors $\mathbf{t}_i$'s are chosen to be i.i.d. Gaussian with zero mean and unit variance.

A secret key K is used for randomizing all of the above steps. Using a different secret key for each image decreases probability of error(both probability of miss and probability of false alarm). This is a result of the algorithm having different random steps. A different key means,different

- choice of sub-images in first factorization step
- different initial point for update algorithm
- secondary image arrangement
- result due to steps (a) and (b) in second NMF part
- different choice of weight vectors in final step.

NMF is distinguished from traditional matrix approximation functions like QR and SVD by its use of non-negativity constraints. These constraints lead to a parts-based representation because they only allow additive not subtractive combinations. This property gives NMF-NMF hashing advantage of capturing local features. Another very useful property of NMF-NMF hashing is that under geometric distortions resulting noise on hash outputs is additive, independent and identically distributed[17]. The importance of this additivity and i.i.d. property of noise allows the usage of a linear method for better performance. This is the main idea behind NMF-NMF-OPCA method which will be explained in detail in next section.

2.5. Proposed-Algorithm Oriented Non-Negative Matrix Factorization(ONMF)

Proposed algorithm is a two step algorithm which merges NMF-based hashing and OPCA. The idea is simply performing a training, based on hash vectors instead of images. The aim is to maximize signal-over-noise ratio for the hash vectors. Steps of the algorithm are as follows:

(i) Obtain hash vectors of training images after second factorization, $\mathbf{h}_K^{NMF-NMF}(I)$'s.

(ii) Obtain hash vectors of attacked training images after second factorization, $\mathbf{h}_K^{NMF-NMF}(\hat{I})$'s.

(iii) Compute difference vectors $\mathbf{z}_i^k$'s using equation (3.10).

(iv) Compute mean μ and covariance matrix $\mathcal{C}$ of training images, using equations (2.6) and (3.11).

(v) Using difference vectors, compute correlation matrix $\mathcal{R}$ using equation (3.12).

(vi) Projection directions are found by solving generalized Rayleigh quotient

$$q = \frac{\mathbf{n}^T \mathcal{C} \mathbf{n}}{\mathbf{n}^T \mathcal{R} \mathbf{n}} \tag{5.23}$$

(vii) Direction vectors can be found by setting $\nabla q = 0$, which will result in a generalized eigenvalue problem. Using the same arguments in the OPCA case, for positive semi-definite $\mathcal{C}$ and $\mathcal{R}$, generalized eigenvalues are positive.

(viii) Define a projection matrix $E = \{\mathbf{n}_i, \ldots, \mathbf{n}_K\}$, using all eigenvectors. Then, projected hash vectors are given by

$$\mathbf{y} = E^T(\mathbf{h}_K^{NMF-NMF}(I) - \mu) \tag{5.24}$$

First of all,note that this algorithm involves a non-linear method, followed by a linear technique and uses both techniques advantages to give better performance than both. NMF-based hashing de-correlates the noise on hash vectors and the noise on hash vectors is additive, independent identically distributed after applying Statistics

Quantization(SQ). Using a linear discrimination method instead of SQ to obtain hash outputs in such a way that signal to noise ratio is maximized, is the proposed change in the algorithm.This approach is not only limited to NMF-based hashing, rather for any image hashing algorithm that contains a suitable stage to apply OPCA. This approach lets us to use NMF-based hashing and OPCA methods advantages at the same time. Distortions including geometric ones are handled by NMF-based hashing part where the noise for modifications becomes additive and independent. Later hash outputs are projected on a space where the noise effects on hash vectors are minimized. Further discussion on NMF-NMF-OPCA method will be given later in this section.

The output of the NMF-NMF-OPCA boils down to "weighted-L2" norm for the case where we use NMF-NMF output as decision statistics. In other words, let $\mathbf{x}_i \in R^n$ represent the output of the NMF-NMF algorithm, the hash vector of image I_i and $E \in R^{n \times n}$ represent the projection matrix. Note that the projection matrix does not reduce the dimensionality. Define $\hat{\mathbf{x}}_i$ as $\hat{\mathbf{x}}_i = \mathbf{x}_i + n$, the hash value of the modified image $\hat{I}_i$. Then the projected outputs are:

$$\mathbf{y} = T\mathbf{x} \tag{5.25}$$

$$\hat{\mathbf{y}} = E(\mathbf{x} + \mathbf{n}) \tag{5.26}$$

$$\hat{\mathbf{y}} = \mathbf{y} + E\mathbf{n} \tag{5.27}$$

Then the distance between the outputs of an image and its modified version boils down to

$$d(\hat{\mathbf{y}}, \mathbf{y}) = \|\hat{\mathbf{y}} - \mathbf{y}\| = \|E\mathbf{n}\| \tag{5.28}$$

$$= \mathbf{n}^T E^T E\mathbf{n} \tag{5.29}$$

Notice that dimensionality reduction using ONMF method can be done in 2 different ways. First of all NMF-based hashing can be used to reduce dimensionality to the required dimension and OPCA part can be used as "weighted-L2" norm as explained

above. Another approach is using OPCA to reduce dimensionality and leaving NMF-based hashing part without any reduction on dimensionality. Also a hybrid method can be used where both method are used to reduce dimensionality at some percentage. The performance for these cases are not explored in this thesis. It remains as an open research as what proportion of reduction should be done in parts such that algorithm gives the best performance. Intuitively non-linear part should be used for most of the dimensionality reduction as NMF-based hashing captures the features that are required for hashing technique and linear method(OPCA) should be used to reduce dimension in such a way that only directions that has low signal to noise ratio is discarded.

Another motivation behind using OPCA method on hash outputs is that we have a prior ideas on the types of images which are going to be used as well as the type of attacks that these original images are going to be subjected to. Thus the aforementioned information enables us to utilize a training based classification technique to further enhance robustness properties. NMF-NMF-OPCA method assumes that a set of attacked images is available as mentioned. This is more convenient than requiring the noise model. We assume that there are 2 types of attacks namely weak attacks and strong attacks; in the former (respectively latter), we assume that there are synchronization marks available (respectively unavailable) on the ID-card. Hence in the weak attack case the distortion on the original image(printed image) consists of a combination of print-scan, compression distortions and potential wear and tear distortions, in the strong attack case the set of attacks in the weak attack case are present as well as potential geometric distortion(mild rotation) attacks present which are due to imperfections that may possibly happen during the scanning case at the receiver side. The specific nature of the attack introduced may potentially have different noise statistics and thus they may potentially change the output of the OPCA stage of the proposed algorithm. In this thesis, for illustration purposes we confined ourselves to a specific set of weak and strong distortions which are further explained in section 3

Usage of a training-based approach is intuitively pleasing and meaningful for the problem setup discussed in Sec. 2.1. This is because of the fact that we confine ourselves to a specific class of images (namely face pictures); furthermore, the class

of attacks in the considered application are more limited than the ones in a typical robust image hashing problem which considers arbitrary modifications that preserve perceptual quality. In particular, we concentrate on face image modifications that may typically arise in the application outlined in Fig. 2.1.

We note that, since it yields a *linear* operator that maximizes SNR, OPCA is expected to enhance the performance in case of additive attacks. As such, if OPCA is applied directly in the image domain (or in a feature domain which is obtained from the input image via applying a linear transform), then one would not expect to experience a performance increase in case of geometric attacks. However, since applying the (non-linear) NMF-NMF operation almost decorrelates the disturbances due to geometric attacks, such attacks produce approximately-additive noise in the NMF-NMF domain (see [17]). Therefore, applying OPCA after NMF-NMF is expected to effectively increase the performance even in case of geometric attacks (justified by our experiments - see Sec. 3).

In the following sections, NMF-NMF-OPCA method will be compared with 2 other methods namely NMF-NMF-PCA and NMF-NMF-SQ. In the case where NMF-NMF is followed by SQ, hash outputs are basically projected randomly in the hash space to achieve desired hash-length. For the PCA case hash outputs are projected to along the directions to maximize variance but noise on hash vectors are not considered in this case.

2.5.1. When to use OPCA?

Usage of OPCA provides an inherent flexibility to the algorithm that may be tailored based on the specific circumstances that occur in utilizing the application in practice. For instance, if the usage of synchronization marks turns out to be effective in practice, then it would be unnecessary to incorporate geometric attacks (such as rotation) in the OPCA training phase; on the other hand, if geometric synchronization turns out to be infeasible in practice, then including such attacks in OPCA training improves the performance.

We note that the proposed NMF-NMF-OPCA algorithm reduces the dimension in two stages, the first (resp. second) of which is achieved via NMF-NMF (resp. OPCA). However, it remains to be a challenging task to "adjust" the amount of dimensionality reduction at each stage so as to achieve a final pre-specified reduced dimension. In practice, we choose the amount of dimensionality reduction at each stage in an experimental manner.

Some further observations are that for the cases where the signal and the noise is i.i.d. distributed in spatial domain, linear discrimination methods will not have any effect. This is due to the fact that there are no directions that favors signal over noise. Because of this fact using OPCA, or in that matter any linear discrimination method for cases where both signal and noise are i.i.d. distributed, will not improve performance. For the NMF-NMF-OPCA case, We need to find the stage where we both have additive noise and correlated noise in the NMF-NMF algorithm. There are 4 possible places where to apply OPCA in this algorithm.

(i) Before NMF-NMF
(ii) After first factorization
(iii) After second factorization
(iv) After NMF-NMF-SQ

Please note that when we refer to NMF-NMF-SQ algorithm it also includes the SQ part at the end of the algorithm where pseudo-random weight vectors are used to reduce dimension to desired length. Before we decide which stage is best to apply OPCA we need to find the behaviors of signal and noise at each stage.

(i) Before the hashing algorithm, we basically have the images and their attacked versions. At that stage noise does not posses additive property whereas it is correlated. Therefore applying OPCA at this stage will not provide any improvement.
(ii) After the first factorization, due to the factorization effect noise can somewhat modeled to have additive property and correlated but especially geometric attacks still does not have additive property. Noise on factorization coefficients are highly-

correlated as well as the signal which results in some improvement on overall performance.

(iii) After the second factorization, noise on hash vectors behaves almost de-correlated, but at this stage regardless of the attack type noise have almost additive property. Although some correlation on both noise and signal is lost (due to shuffling during second image creation), this step is the best spot for OPCA for NMF-NMF algorithm.

(iv) At this stage after applying SQ the noise on the hash vectors have additive property, but the drawback of applying OPCA at this stage is that due to SQ both signal and noise have i.i.d. property which makes OPCA useless.

Note that from the start to the end of NMF-based hashing algorithm, correlation on noise and signal decrease with each step whereas noise behaves more additive with each step. Since we are using a linear discriminant method finding the stage where noise has additive property is more crucial. To summarize, the decision between 4 possible stages to apply OPCA, the first and the last stage is obviously not good choices since in the first noise is on images thus has no additive property and in the last stage noise is almost fully de-correlated. Between the second and the third stage using the fact that additivity is more crucial for the application the third stage seems to be the best choice to apply OPCA.

Finally note that the directions chosen by OPCA method does not have same SNR value. At the decision point having a weighted L2-norm based on the SNR values of the corresponding projection direction, note that SNR values are the eigenvalues in this case, will give an improved performance. This is due to the fact that high SNR directions are more reliable than the other directions.

As can be seen from the figure 2.2 the most suitable places to apply OPCA are after second factorization and before the NMF-NMF algorithm. But since the latter changes the input to the hashing algorithm the final performance is not improved. Using after NMF-NMF as a final step gives much better results. Because at this stage the noise on hash vectors posses additive property and since OPCA is a linear

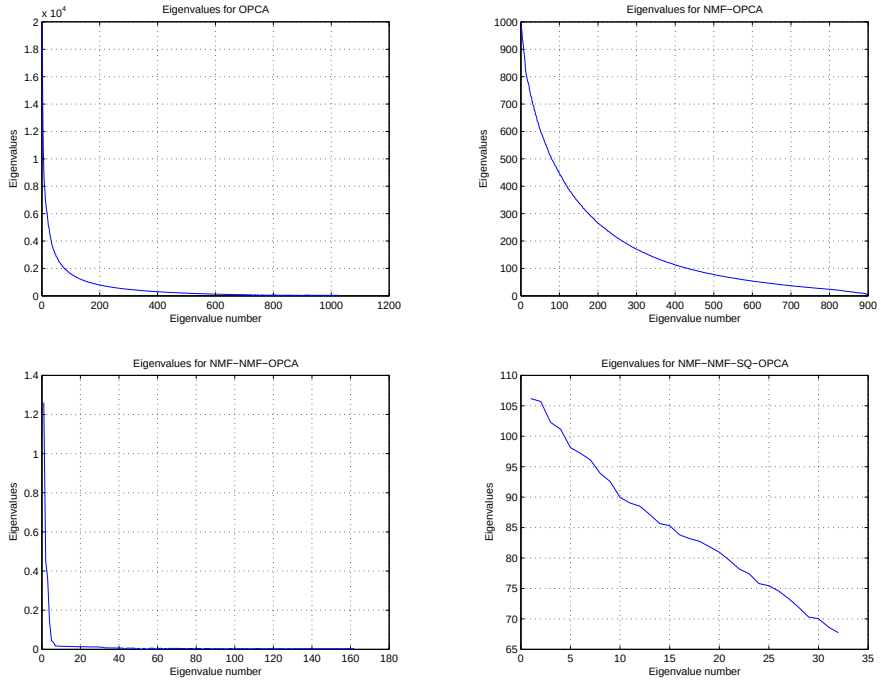

Figure 2.2. Eigenvalue distribution of matrix $R^{-1}C$ obtained from different stages of NMF-NMF algorithm

discrimination method, it is the best place to use OPCA.

3. Experimental Setup and Results

In this chapter, several simulation results will be shown. First of these simulations are conducted in order to evaluate system performance for different hashing techniques. Secondly simulations regarding meshed techniques are shown. A detailed discussion will be given as well as interpretation of result in the preceding sections.

In order to measure robustness, two different attack sets are considered in experiments as well as individual modifications. But for the applications case we are more interested in these attack sets which consists of different modifications applied at the same time. In the first case, using the assumption that images have synchronization marks on them that allows to prevent rotational attacks thus resulting in a small degree of rotation. Also no cropping or scaling attacks are used for this case. And in the second case no synchronization marks are used which results in a strong rotational attack as well as cropping and scaling. In both cases histogram equalization, additive white gaussian noise and print scan attacks are also applied.

3.1. Comparison of Methods

For each of the method described in 2, statistical comparison is performed using a database consisting of 3600 face images. Images are $100x66$ intensity images. The attacks considered are:

(i) Rotation
(ii) Image resizing
(iii) Cropping (used with rotation)
(iv) Random contrast enhancement
(v) Print-Scan (PS)
(vi) Additive white Gaussian Noise (AWGN).

Figure 3.1. Original Image, Rotated, Resized, Cropped, Contrast enhanced, PS attacked, AWGN respectively.

Figure 3.1 shows a sample of original image and attacked images. Let I be an image and Q be an attack from the attack set A such as, $Q \subset A$. Experimental setup is as follows:

(i) Apply resize (32 by 32) to all images

(ii) Obtain length-32 hash vectors $\mathbf{h}_m(I_i)$, i.e. $\mathbf{h}_{PCA}(I_i)$ denotes the hash vector of image i obtained by using PCA method. These hash vectors are labeled as original hash vectors. For the techniques where NMF based hashing is used, a different keys are not used unless otherwise stated. This is due to the fact that using a secret key for each image will not result in a fair comparison between hashing methods. PCA and OPCA only takes image as an input for hashing function. One can argue that although the aforementioned methods only takes image as input, they also have training phases which gives them an edge. But it should also be noted that these methods are linear techniques whereas NMF based hashing is non-linear.

(iii) Select an attack Q from attack set A, and apply on image I,such that $\hat{I} = Q(I)$

(iv) Obtain hash vectors using hashing function $H(.)$.

(v) Randomly select 3 images, I_i, $\hat{I}_i$ and $\hat{I}_j$, where $i \neq j$. Note that this step is not limited to number of images used. Although we select 3 images, the chosen images in this part are called random pairs. This is because we select 2 different images actually, third image chosen is an attacked version of one of the previously chosen images.

(vi) Obtain $d(H(I_i), H(\hat{I}_i))$ and $d(H(I_i), H(\hat{I}_j))$ of the selected images in the previous step. Here $d(.,.)$ denotes the metric used to measure the distance between hash

vectors. In this experiment Euclidean distance is used for measuring distances between hash vectors.

(vii) Find empirical probability of miss and false alarm curves. That is, for a fixed τ:

$$P_M(\tau) = \frac{number\ of\ d(I_i, \hat{I}_i) > \tau}{total\ number\ of\ iterations} \quad (1.1)$$

$$P_F(\tau) = \frac{number\ of\ d(I_i, \hat{I}_j) < \tau}{total\ number\ of\ iterations} \quad (1.2)$$

Probability of miss and false alarm curves are obtained by calculating P_M and P_F for $\tau = min((d(I_i, \hat{I}_i), d(I_i, \hat{I}_j)), \ldots, max(d(I_i, \hat{I}_i), d(I_i, \hat{I}_j))$.

10.000 iteration are done for each attack. For NMF-NMF-OPCA, PCA and OPCA methods 1500 images are used as training images, remaining 2100 are used for testing. So that the testing results are not biased. For the NMF-NMF-SQ case all the images are used for testing as NMF-NMF-SQ algorithm does not have a training phase. For the NMF-NMF and NMF-NMF-OPCA cases key space is spanned by using different case for at each iteration. Note that this is not equal to using secret key for each image.

Figures 3.2, 3.3 and 3.4 shows the performance of different hashing techniques under individual attack, AWGN, print-scan, rotation, histogram equalization, combination of these attacks respectively. It can be seen that for each individual case NMF-NMF-OPCA gives better performance. These results clearly show that PCA and OPCA are not suited for this task when used for hashing as a stand alone technique.

Figure 3.5 shows that PCA and OPCA can be used as a stand alone technique where hash-length constraints are less strict, in that case PCA and OPCA gives comparable results to NMF-NMF-SQ with hash-length 32.

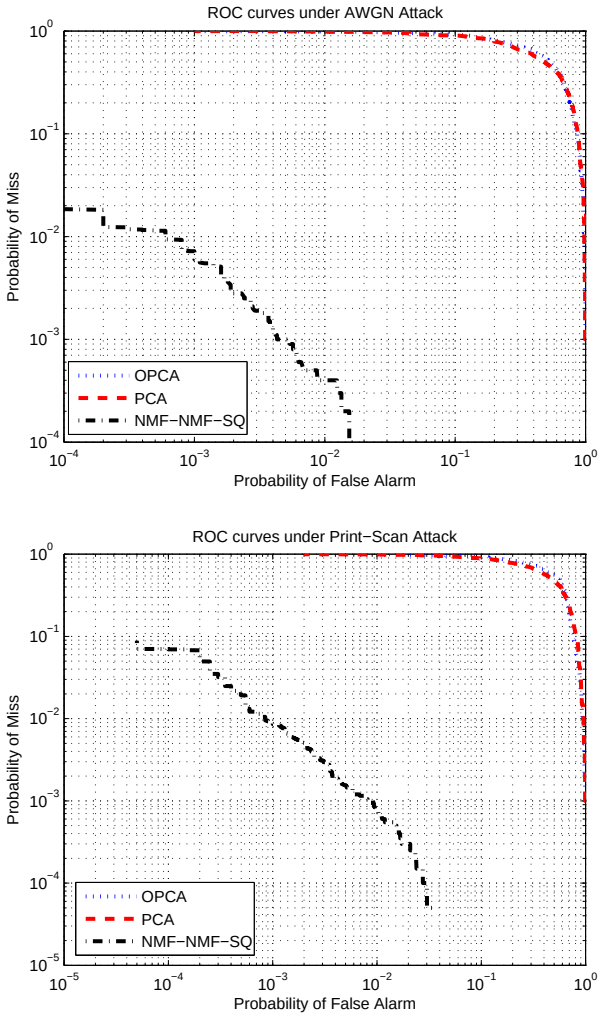

Figure 3.2. Simulation results for PCA, OPCA and NMF-NMF-SQ under AWGN, Print-Scan respectively.

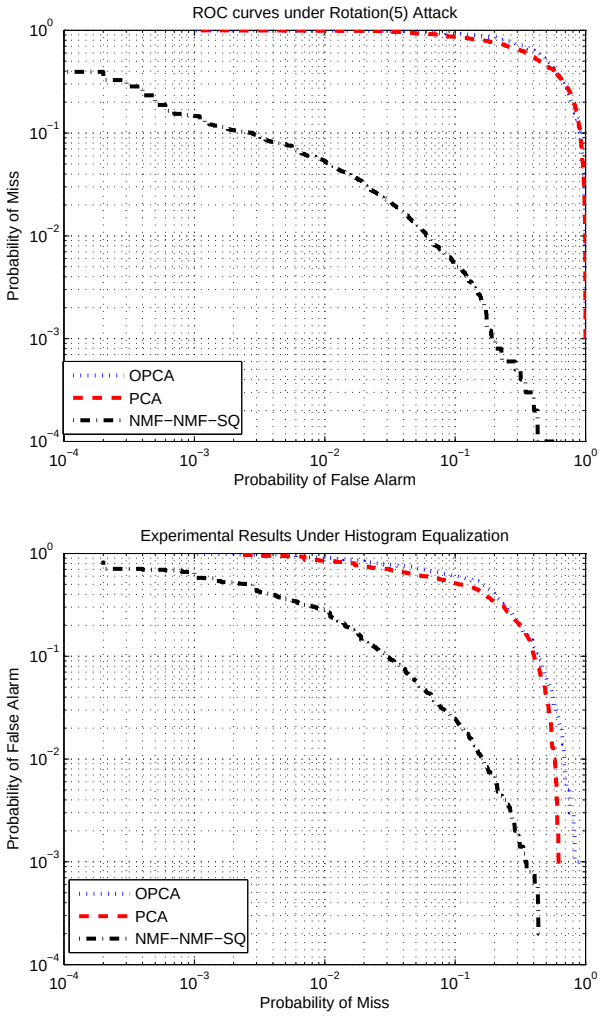

Figure 3.3. Simulation results for PCA, OPCA and NMF-NMF-SQ under Rotation by 5, Histogram Equalization attacks respectively.

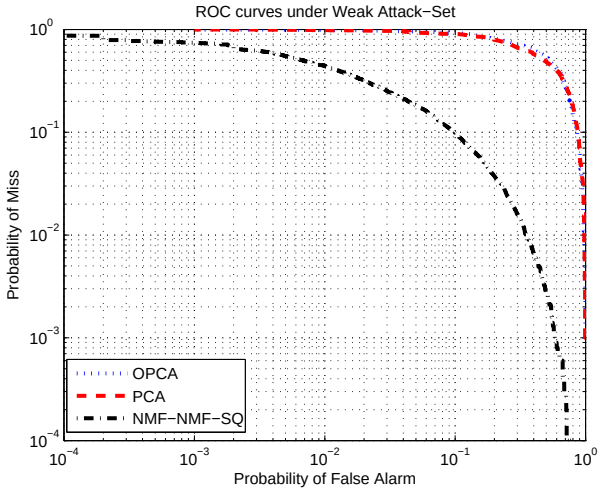

Figure 3.4. Simulation results for PCA, OPCA and NMF-NMF-SQ under Weak Attack-Set.

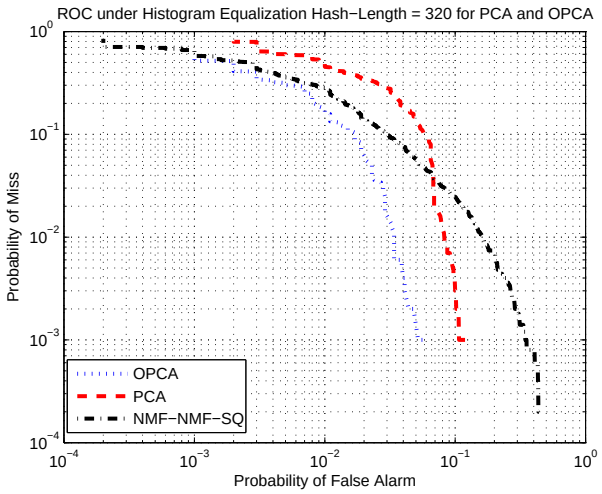

Figure 3.5. Simulation results for PCA, OPCA and NMF-NMF-SQ under Histogram Equalization Attack Hash-Length is 320 for PCA and OPCA, 32 for NMF-NMF-SQ.

3.2. Learning Based Hashing

In this section, we will consider the case where PCA and OPCA methods are used with NMF-NMF algorithm to produce the hash outputs. V. Monga and K. Mıhçak used SQ after NMF-NMF algorithm to achieve desired hash length. This is basically projecting output of NMF-NMF algorithm to a random sub-space. Instead of using a random projection, We will use the directions obtained from OPCA and PCA which chooses directions that maximize SNR and variance of hash outputs respectively. For experiments 3500 images are used, 1500 images are used for training and remaining 2000 images are used for testing. Among testing images 5000 random pairs are chosen to evaluate performances. The results are shown in Figure 3.6. For the cases where AWGN and Print-Scan is applied, ROC curve for NMF-NMF-OPCA is not available, the distances are distinct. For these cases probability of miss and false alarm versus threshold graph is given in 3.7. As can be seen from these figures, NMF-NMF-OPCA gives better results in all cases as expected. Combined attack case is more interesting for us since, it is the most realistic scenario. Combined attack case results are shown in Figure 3.8.

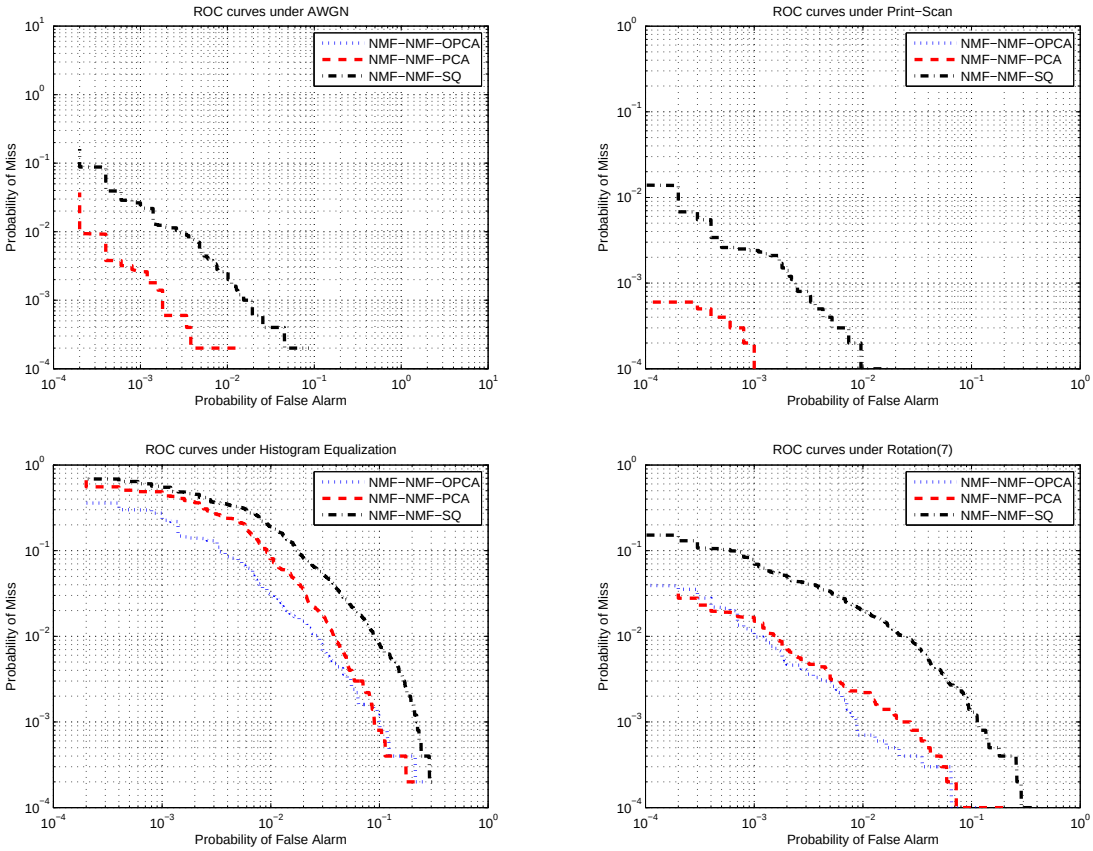

Figure 3.6. Simulation results for NMF-NMF-PCA, NMF-NMF-OPCA and NMF-NMF-SQ under AWGN, Print-Scan, Histogram Equalization and Rotation(7) respectively.

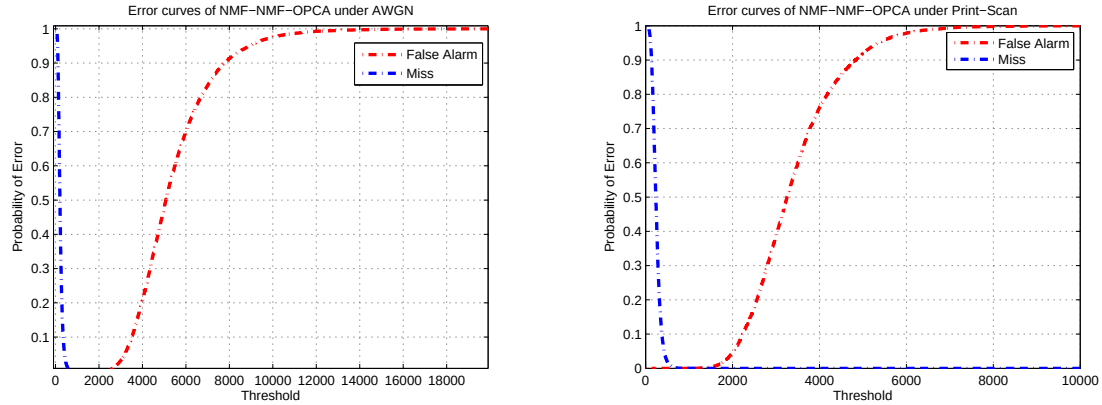

Figure 3.7. Simulation results for NMF-NMF-OPCA under AWGN, Print-Scan respectively.

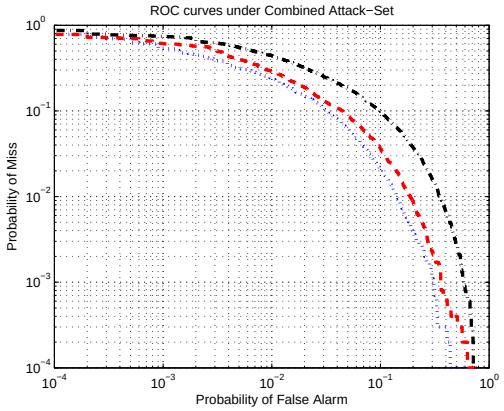

Figure 3.8. Simulation results for NMF-NMF-PCA, NMF-NMF-OPCA and NMF-NMF-SQ under Combined attacks namely, AWGN, Print-Scan, Resize, Crop, Histogram Equalization and Rotation respectively.

3.3. Secret Key Setup

For the setup where a different key is used for each image, system performance increases drastically as expected. Details of advantages of using secret key is given in section 2.4.3. To evaluate system performance where secret key is used, 32-bit NMF-NMF output hashes are used, 1000 images are used as a training set and 3000 images are used as a testing set. 5000 image pairs are used to evaluate system performance. Further results can be found in [17].

3.4. Rectangle Size and Number of Rectangles Analysis

In NMF-NMF-SQ based hashing a number of rectangles are chosen from the input image and these rectangles are used to create second image. It is very hard to make a analytic analysis to chose the optimum number of rectangles(NR and size of these rectangles(RS). Note that total pixels chosen from the image is equal to $NR \times RS^2$, for the cases where squares are chosen and these chosen pixels can be overlapping.

Results of simulations are shown in figure 3.11. Random number and sequence of attacks are chosen for this simulation. Same key is used for all images and for all NR(Number of Rectangles), RS(Rectangle Size) pairs. Considered parameters are

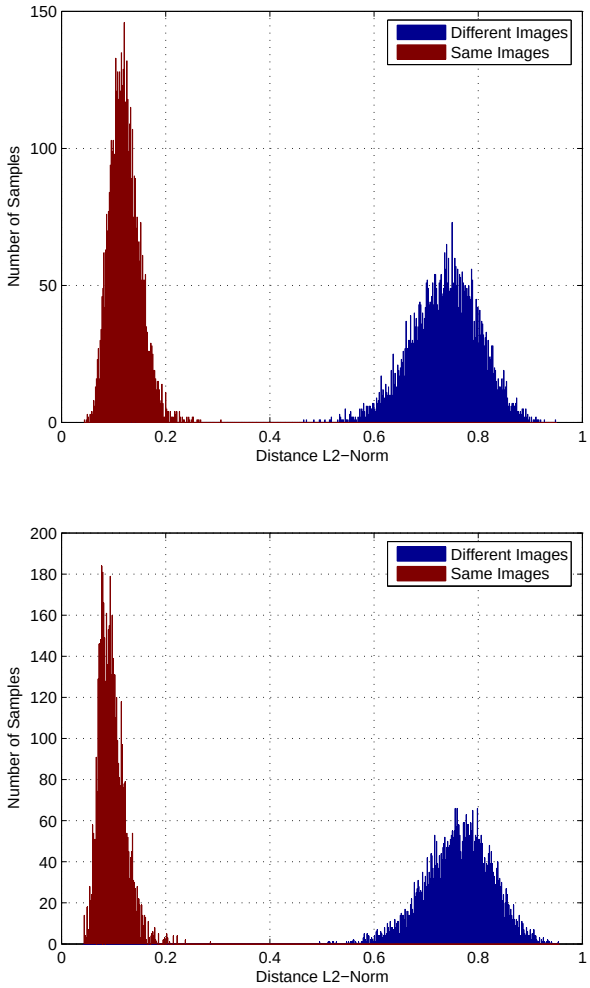

Figure 3.9. Simulation results for NMF-NMF-SQ and for NMF-NMF-OPCA using secret key under attacks Rotation by 7, Print-Scan, Histogram Equalization, AWGN, Scaling, Cropping,

$5, 10, \ldots, 45, 50$ for NR and $2, 4, \ldots, 14, 16$ for RS. Results of the simulation shows for $NR = 45$ and $RS = 12$ the algorithm gives best results. The results are inconclusive, in other words there is no correlation between number of pixels chosen and performance. The optimum parameters may change depending on database, attack set, hash length...etc. But this process can be thought of a training phase for a given database and attack set. For some applications attacks or modifications on images are known, in these cases finding the (NR, RS) pair that gives the best results, will increase the performance of the hash algorithm. Such a preprocessing may be viable in application scenario described in section 2.1 where images comes from a constrained set and a database is available.

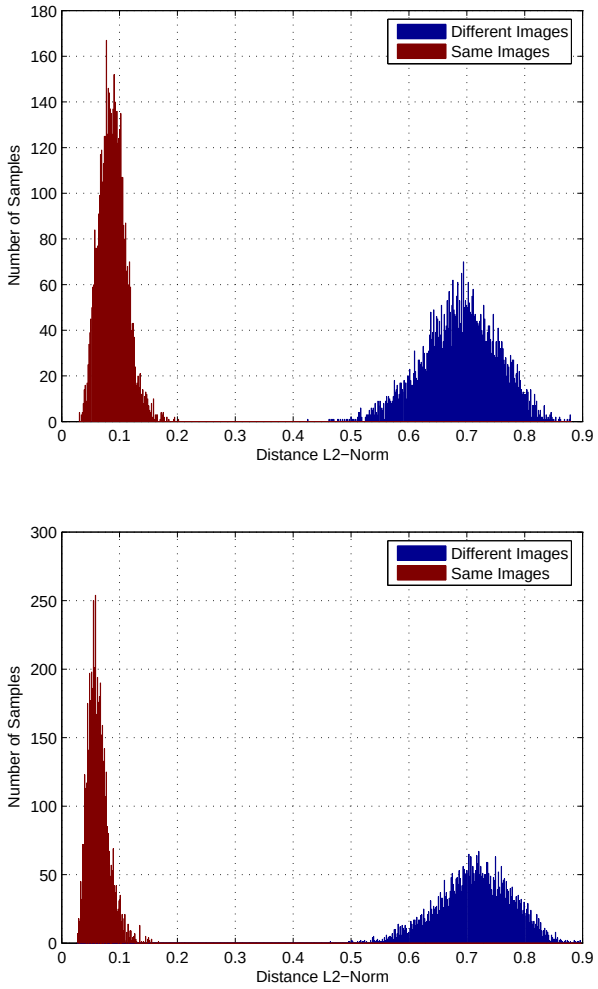

Figure 3.10. Simulation results for NMF-NMF-SQ and for NMF-NMF-OPCA using secret key under attacks Rotation by 2, Print-Scan, Histogram Equalization, AWGN, Scaling, Cropping

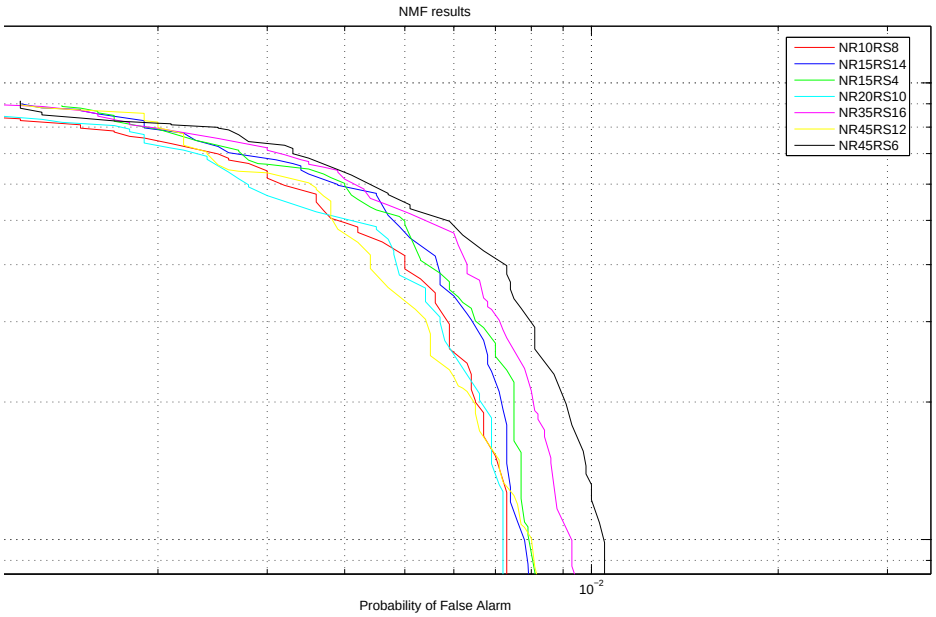

Figure 3.11. NMF-NMF-SQ results under different rectangle sizes and number of rectangles.

4. CONCLUSION AND FUTURE WORK

In this thesis, we show that for a constrained set of images and a constrained set of attacks, using learning based approaches improves performance. In the literature, hash functions are ignorant of the images being hashed. But for many applications this is not the case, a-priori knowledge on images are available. We utilized this information by using OPCA and PCA technique. Although all the simulations in this thesis is done using NMF-NMF hashing algorithm, this approach can be extended to any hashing algorithm that will be used under such circumstances. Furthermore detection estimation analysis of the proposed algorithm is given and theoretical ROC curves are computed.

Future work includes, testing proposed algorithm on a different image database and under different attacks. Also using a different detector other than energy detector may give better results. Applying OPCA to another hashing function other than NMF-NMF is also part of future work.

During this thesis, I have learned many things but the most important one for me is to learn how to conduct a research. I get many bad results and have my share of disappointment, but in the end the joy of finding an algorithm that works better than any other known methods overcame all. I also learned that although you have to learn to trust your instincts, you should also be prepared that they may not be right at all times.

Finally this thesis includes a different look at the hashing problem. It is pleasing intuitively as well as theoretically and outperforms the best hashing algorithm up to date. Using every information available to you during this problem is very crucial which I tried to utilize to my best at this algorithm. Although this proposed algorithm is far from being optimal, it is a step in the right direction.

APPENDIX A: DETECTION ESTIMATION ANALYSIS OF NMF-NMF-OPCA

Here, we will represent the detection analytic analysis of the NMF-NMF-OPCA method. The setup is as follows:

(i) $\mathbf{y}$: is the hash of the received image.
(ii) $\mathbf{x}$: is the hash of the reference image.
(iii) $\mathbf{z}$: is the hash of another image different than reference image.
(iv) $\mathbf{n}$: is the noise on hash vectors.

Assumptions:

(i) Every image and noise vectors are N-dimensional such that $\mathbf{x}, \mathbf{z}, \mathbf{n} \in R^N$ and they are independent.
(ii) Hash of the images comes from a zero-mean colored gaussian distribution as $\mathbf{x}, \mathbf{z} \sim N(0, \Sigma_s)$.
(iii) Noise on the hash vectors of the images comes from a zero-mean colored gaussian distribution $\mathbf{n} \sim N(0, \Sigma_n)$.

These assumptions comes from observation of hash outputs. Throughout this part we will use image and image hash interchangeably, because every image will have its corresponding hash vector coming from the same hash function. Also note that since We are comparing the received image with reference image from the database, $\mathbf{x}$ namely the reference is known.The hypothesis are as follows:

(i) H_0: The received image $\mathbf{y}$ is an attacked version of the reference image $\mathbf{x}$ such that $\mathbf{y} = \mathbf{x} + \mathbf{n}$.
(ii) H_1: The received image $\mathbf{y}$ is a different image from the reference image, $\mathbf{y} = \mathbf{z} + \mathbf{n}$.

$$H_0 \;:\; \mathbf{y} = \mathbf{x} + \mathbf{n} \tag{A.1}$$

$$H_1 \;:\; \mathbf{y} = \mathbf{z} + \mathbf{n} \tag{A.2}$$

Defining 2 variables $\mathbf{w}, \mathbf{v}$ as $\mathbf{w} = \mathbf{y} - \mathbf{x}$ and $\mathbf{v} = \mathbf{z} - \mathbf{x}$, the hypothesis becomes

$$H_0 \;:\; \mathbf{w} = \mathbf{n} \tag{A.3}$$

$$H_1 \;:\; \mathbf{w} = \mathbf{v} + \mathbf{n} \tag{A.4}$$

The PDF of $\mathbf{w}$ under hypothesis are as follows:

$$f(w|H_0, x) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma_n|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{n}^T\Sigma_n^{-1}\mathbf{n}} \tag{A.5}$$

$$f(w|H_1, x) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma_n + 2\Sigma_s|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{z}-\mathbf{x}+\mathbf{n})^T(\Sigma_n+2\Sigma_s)^{-1}(\mathbf{z}-\mathbf{x}+\mathbf{n})} \tag{A.6}$$

Using an energy detector,

$$\|\mathbf{w}\|^2 \underset{H_0}{\overset{H_1}{\gtrless}} \tau \tag{A.7}$$

we can define probability of false alarm and miss for a given τ and $\mathbf{x}$.

$$P_F(\mathbf{x}, \tau) = Pr[\|z - x + n\|^2 < \tau | H_1] \tag{A.8}$$

$$P_M(\mathbf{x}, \tau) = Pr[\|n\|^2 > \tau | H_0] \tag{A.9}$$

where P_F denotes probability of false alarm, deciding that 2 different images are same images and P_M denotes probability of miss, deciding that 2 perceptually identical

images are different images. Note that in both hypothesis $w \sim N(0, \Sigma_w)$ where,

$$\Sigma_w|H_0 = \Sigma_n \tag{A.10}$$

$$\Sigma_w|H_1 = \Sigma_n + 2\Sigma_s \tag{A.11}$$

Furthermore in both hypothesis Σ_w is symmetrical positive definite matrices by construction. Defining δ_M as:

$$\delta_M(\mathbf{n}, \tau) = \begin{cases} 1 & : \mathbf{n}^T\mathbf{n} > \tau \\ 0 & : \mathbf{n}^T\mathbf{n} < \tau \end{cases}$$

and δ_F as:

$$\delta_F(\mathbf{n}, \mathbf{z}, \mathbf{x}, \tau) = \begin{cases} 1 & : (\mathbf{z}-\mathbf{x}+\mathbf{n})^T(\mathbf{z}-\mathbf{x}+\mathbf{n}) < \tau \\ 0 & : (\mathbf{z}-\mathbf{x}+\mathbf{n})^T(\mathbf{z}-\mathbf{x}+\mathbf{n}) > \tau \end{cases}$$

To find the theoretical probability of miss and false alarm, we need to compute:

$$P_M(\tau) = \int_{\mathbf{n}} \delta_M(\mathbf{n}, \tau) f_{\mathbf{n}}(\mathbf{n}) d\mathbf{n} \tag{A.12}$$

$$= E_{f(\mathbf{n})}[\delta_M(\mathbf{n}, \tau)] \tag{A.13}$$

$$P_F(\tau) = \int_{\mathbf{x}}\int_{\mathbf{z}}\int_{\mathbf{n}} \delta_F(\mathbf{n}, \mathbf{z}, \mathbf{x}, \tau) f_{\mathbf{x}}(\mathbf{x}) f_{\mathbf{z}}(\mathbf{z}) f_{\mathbf{n}}(\mathbf{n}) d\mathbf{x} d\mathbf{z} d\mathbf{n} \tag{A.14}$$

$$= E_{f(\mathbf{x}),f(\mathbf{z}),f(\mathbf{n})}[\delta_F(\mathbf{x}, \mathbf{z}, \mathbf{n}, \tau)] \tag{A.15}$$

Note that both hash of images and noise on hashes are colored gaussian. So given $\mathbf{x}, \mathbf{z}, \mathbf{n}$ are gaussian, also $\mathbf{z} - \mathbf{x} + \mathbf{n}$ is a gaussian. The problem reduces to finding

squares of colored gaussian variables, namely:

$$H_0 \quad : \quad \mathbf{n}^T\mathbf{n} = \sum_i^N n_i^2 \tag{A.16}$$

$$H_1 \quad : \quad \mathbf{m}^T\mathbf{m} = \sum_i^N m_i^2 \tag{A.17}$$

where $\mathbf{m} = \mathbf{z} - \mathbf{x} + \mathbf{n}$. The resulting pdf in both cases are known as generalized chi-squared distribution which does not have a closed from expression. It can only be computed using numerical methods. We used Monte-Carlo simulations to compute the theoretical probability of miss and false alarm. The results are as follows:

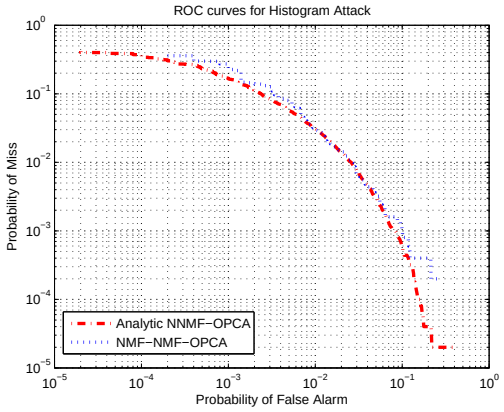

Figure A.1. NMF-NMF-OPCA and analytic ROC curve under Histogram Equalization Attack

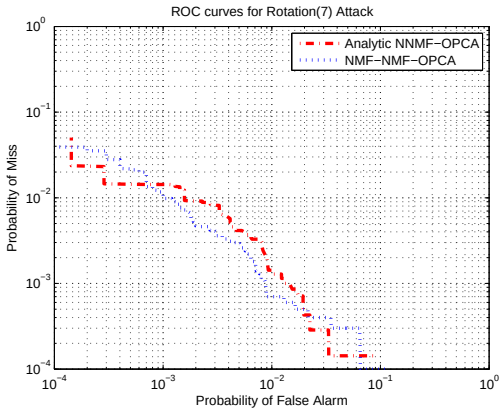

Figure A.2. NMF-NMF-OPCA and analytic ROC curve under rotation by 7 Attack

The analytical results are consistent with empirical results. More iterations are required for the cases where empirical error probabilities are small. The number of

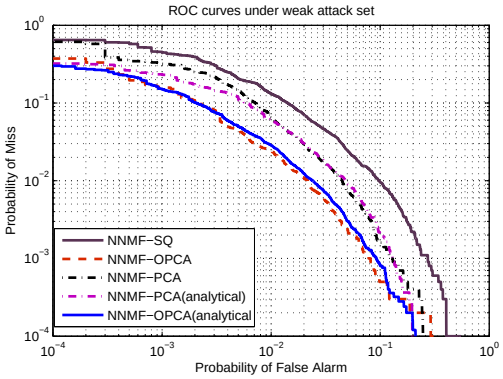

Figure A.3. NMF-NMF-OPCA and analytic ROC curve under weak attack set

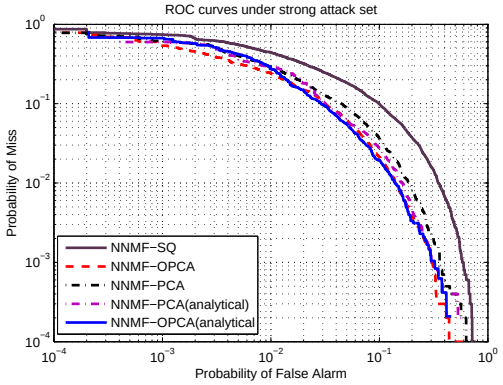

Figure A.4. NMF-NMF-OPCA and analytic ROC curve under strong attack set

iterations required changes for attack types as the error probabilities changes for each attack type.

APPENDIX B: UPDATE RULES

Theorem 1. *The Euclidean Distance* $\|V - WH\|$ *is non-increasing under the update rules*

$$H_{a\mu} \leftarrow H_{a\mu}\frac{(W^TV)_{a\mu}}{(W^TWH)_{a\mu}}\ W_{ia} \leftarrow W_{ia}\frac{(VH^T)_{ia}}{(WHH^T)_{ia}} \tag{B.1}$$

For the cases where W and H are at a stationary point of distance, Euclidean Distance is invariant.

Theorem 2. *The divergence* $D(V\|WH)$ *is non-increasing under the update rules*

$$H_{a\mu} \leftarrow H_{a\mu}\frac{\sum_i W_{ia}V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}}\ W_{ia} \leftarrow W_{ia}\frac{\sum_\mu H_{a\mu}V_{i\mu}/(WH)_{i\mu}}{\sum_v H_{av}} \tag{B.2}$$

For the cases where W and H are at a stationary point of distance, KL-Divergence is invariant.

Before we start to prove the theorems described above, consider two alternative formulations of NMF as optimization problems:

Problem 1. *Minimize* $\|V - WH\|^2$ *with respect to W and H, subject to the constraints* $W, H \geq 0$.

Problem 2. *Minimize* $D(V\|WH)$ *with respect to W and H, subject to the constraints* $W, H \geq 0$.

Since the cost functions defined above are convex in W only or H only, it is unrealistic to expect an algorithm to solve **Problem 1** and **2** in sense of finding a global minima.But, there are numerical optimization techniques that can be applied to find local minima. It is useful to compare update rules defined in **Theorem 1** and **2** with those arising from gradient descent. In particular, a very simple additive update

rule for H that reduces the squared distance can be written as,

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}[(W^T V)_{a\mu} - (W^T W H)_{a\mu}] \tag{B.3}$$

For small $\eta_{a\mu}$ this update rule reduces to conventional gradient descent. Sufficiently small and positive $\eta_{a\mu}$ should reduce the cost function $\|V - WH\|^2$. If we set

$$\eta_{a\mu} = \frac{H_{a\mu}}{(W^T W H)_{a\mu}}, \tag{B.4}$$

update rule given in (B.3) turns into the same update rule given in **Theorem 1**.

For KL-divergence, gradient descent is,

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}\Big[\sum_i W_{ia}\frac{V_{i\mu}}{(WH)_{i\mu}} - \sum_i W_{ia}\Big] \tag{B.5}$$

Again, a sufficiently small and positive choice for $\eta_{a\mu}$ should reduce the KL-divergence $D(V\|WH)$. If we set

$$\eta_{a\mu} = \frac{h_{a\mu}}{\sum_i W_{ia}}, \tag{B.6}$$

in (B.5), we obtain update rule defined in **Theorem 2**.

Definition 1. *$G(h, h')$ is an auxiliary function for $F(h)$,if it satisfies*

$$G(h, h') \geq F(h)\ , \quad G(h, h) = F(h) \tag{B.7}$$

This auxiliary function definition is useful for the following lemma defined as,

Lemma 1. *if G is an auxiliary function for $F(h)$, then $F(h)$ is non-increasing under the update rule given as,*

$$h^{t+1} = \underset{h}{\operatorname{argmin}}\, G(h, h^t) \tag{B.8}$$

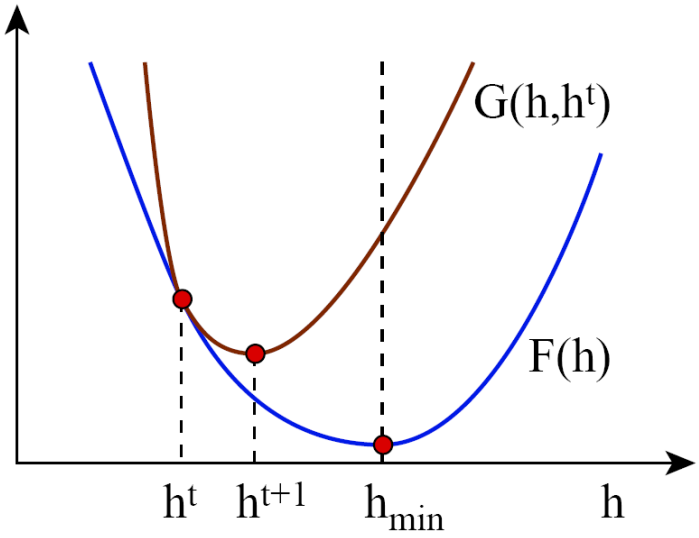

Figure B.1. Minimizing the auxiliary function defined as $G(h, h^t) \geq F(h)$ implies that $F(h^{t+1}) \leq F(h^t)$ for $h^{t+1} = \text{argmin}_h G(h, h^t)$

.

Proof. For an auxiliary function $G(h, h')$ for $F(h)$

$$F(h^{t+1}) \leq G(h^{t+1}, h^t) \tag{B.9}$$

by definition and

$$G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t) \tag{B.10}$$

Finally combining two equations we get

$$F(h^{t+1}) \leq G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t) \tag{B.11}$$

□

Note that $F(h^{t+1}) = F(h^t)$ only if h^t is local minimum of auxiliary function $G(h, h^t)$. If the derivatives of F exist and continuous in a small neighborhood of h^t, this also implies that the derivatives $\nabla F(h^t) = 0$. So using equation (B.8), a sequence

of estimates that converge to local minimum

$$h_{min} = \underset{h}{\operatorname{argmin}} F(h)$$

can be obtained.

$$F(h_{min}) \leq \ldots F(h^{t+1}) \leq F(h^t) \ldots \leq F(h_1) \leq F(H_0) \tag{B.12}$$

Lemma 2. *If $K(h^t)$ is the diagonal matrix*

$$K_{ab}(h^t) = \delta_{ab}(W^T W h^t)_a / h^t_a \tag{B.13}$$

then

$$G(h, h^t) = F(h^t) + (h - h^t)\nabla F(h^t) + \frac{1}{2}(h - h^t)^T K(h^t)(h - h^t) \tag{B.14}$$

is an auxiliary function for

$$F(h) = \frac{1}{2}\sum_i (v_i - \sum_a W_{ia} h_a)^2 \tag{B.15}$$

Proof. It is straight-forward to show that $G(h, h) = F(h)$. To show $G(h, h^t) \geq F(h)$, we compare

$$F(h) = F(h^t) + (h - h^t)\nabla F(h^t) + \frac{1}{2}(h - h^t)^T (W^T W)(h - h^t) \tag{B.16}$$

with (B.14). $G(h, h^t) \geq F(h)$ is equivalent to

$$0 \leq (h - h^t)^T [K(h^t) - W^T W](h - h^t) \tag{B.17}$$

We need prove positive semi-definiteness of $K - W^T W$, first we define M_{ab} as

$$M_{ab}(h^t) = ha^t(K(h^t) - W^T W)_{ab} h_b^t \tag{B.18}$$

which is basically re-scaling of $K - W^T W$. M is positive semi-definite

$$0 \leq v^T M v \ , \qquad \forall v$$

we can show positive semi-definiteness by,

$$\begin{aligned} v^T M v &= \sum_{ab} v_a M_{ab} v_b \\ &= \sum_{ab} h_a^t (W^T W)_{ab} h_b^t v_a^2 - v_a h_a^t (W^T W)_{ab} h_b^t v_b \\ &= \sum_{ab} (W^T W)_{ab} h_a^t h_b^t \Big[\frac{1}{2} v_a^2 + \frac{1}{2} v_b^2 - v_a v_b \Big] \\ &= \frac{1}{2} \sum_{ab} (W^T W)_{ab} h_a^t h_b^t (v_a - v_b)^2 \\ &\geq 0 \end{aligned}$$

□

Now we can prove **Theorem 1** by using (B.14) in (B.8), then the update rule becomes:

$$h^{t+1} = h^t - K(h^t)^{-1} \nabla F(h^t) \tag{B.19}$$

Under this update rule F is non-increasing as shown in Lemma 1. If we explicitly write the components of this equation, we obtain

$$h_a^{t+1} = h_a^t \frac{(W^T v)_a}{(W^T W h^t)_a} \tag{B.20}$$

F can also be shown to be non-increasing under the update rule defined for W, by switching roles of W and H in Lemma 1 and 2 and following the same steps.

For KL-divergence, we start by defining an auxiliary function,

Lemma 3.

$$G(h, h^t) = \sum_i (v_i \log v_i - v_i) + \sum_{ia} W_{ia} h_a - \sum_{ia} v_i \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \Big(\log W_{ia} h_a - \log \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \Big) \tag{B.21}$$

is an auxiliary function for

$$F(h) = \sum_i v_i \log \left(\frac{v_i}{\sum_a W_{ia} h_a} \right) - v_i + \sum_a W_{ia} h_a \tag{B.22}$$

Proof. It is straightforward to show that $G(h, h) = F(h)$. We can write

$$-\log \sum_a W_{ia} h_a \leq -\sum_a a_a \log \frac{W_{ia} h_a}{a_a} \tag{B.23}$$

using convexity of the log function. This inequality holds for all non-negative a_a that sum to unity. If we set

$$a_a = \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \tag{B.24}$$

we get,

$$-\log \sum_a W_{ia} h_a \leq -\sum_a \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \Big(\log W_{ia} h_a - \log \frac{W_{ia} h_a^t}{\sum_b W_{ib} h_b^t} \Big) \tag{B.25}$$

This inequality proves that $G(h, h^t) \geq F(h)$. □

We can now prove **Theorem 2** using auxiliary function defined in (B.21).

Proof. The minimum of $G(h, h^t)$ with respect to h is obtained by setting derivative

with respect to h to zero.

$$\frac{dG(h,h^t)}{dh_a} = -\sum_i v_i \frac{W_{ia}h_a^t}{\sum_b W_{ib}h_b^t}\frac{1}{h_a} + \sum_i W_{ia} = 0 \tag{B.26}$$

Using this result equation (B.8) becomes

$$h_a^{t+1} = \frac{h_a^t}{\sum_b W_{kb}} \sum_i \frac{v_i}{\sum_b W_{ib}h_b^t} W_{ia} \tag{B.27}$$

F is non-increasing under this update rule, as G is an auxiliary function. Rewriting this equation in matrix form results in equation (B.2), update rule defined in **Theorem 2**. By reversing roles of W and H update rule for W can be shown to be non-increasing as well. □

APPENDIX C: DETECTION THEORETIC ANALYSIS OF NMF-NMF-SQ HASHING

For given an image I and a key K, let $h_K(I)$ denote the output of a hash algorithm. A robust hash should have the property that when input(image I in this case) is perturbed by an attack A, the hash value does not changed much. That is to say $\|h_K(I) - h_K(A(I)) < \tau$, with high probability. Also, when two distinct images I and I' are compared then, $\|h_K(I) - h_K(I') > \tau$ with high probability. Note that, $\|.\|$ denotes a meaningful notion of distance on difference of hash vectors. Probability of miss and false alarm is defined as,

$$Probability\ of\ miss\ :\ P_M(\tau) = Pr(\|h_K(I) - h_K(A(I))\| > \tau) \tag{C.1}$$

$$Probability\ of\ false\ alarm:\ P_F(\tau) = Pr(\|h_K(I) - h_K(I')\| < \tau) \tag{C.2}$$

To evaluate statistical performance, we consider the problem of finding the optimum threshold τ for NMF based hashing algorithm, in order to minimize total error, miss and false alarm, defined in (C.1) and (C.2), respectively.

For a query image $\hat{I}$, we define two hypothesis as H_0 is the hypothesis that $\hat{I}$ is an attacked version of image I, i.e. $\hat{I} = A(I)$ and H_1 is the hypothesis that $\hat{I}$ is a virtually different image than I. We can define these hypothesis as,

$$H_0 :\ \mathbf{y} \quad = \quad \mathbf{x} + \mathbf{n}_A \tag{C.3}$$

$$H_1 :\ \mathbf{y} \quad = \quad \mathbf{z} \tag{C.4}$$

where, $\mathbf{x}$ denotes the hash vector of image I, $\mathbf{y}$ denotes the hash vector of query image $\hat{I}$, $\mathbf{z}$ denotes the hash vector of an image virtually distinct from image I and $\mathbf{n}_A$

is the noise vector resulting from attack A.If we rewrite (C.3) and (C.4) in terms of difference of hash vectors, $\mathbf{v} = \mathbf{y}$ - $\mathbf{x}$, we obtain alternative hypothesis.

$$H_0 : \mathbf{v} = \mathbf{n}_A \tag{C.5}$$

$$H_1 : \mathbf{v} = \mathbf{z} - \mathbf{x} \tag{C.6}$$

Recall from NMF-NMF hashing final step,

$$\mathbf{v}(i)/H_0 =< \mathbf{n}, \mathbf{t}_i >= \sum_{j=1}^{N} \mathbf{n}(j)\mathbf{t}_i(j),\ i = 1, 2, \dots\ , M. \tag{C.7}$$

where $\mathbf{n}$ represents the noise on NMF-NMF hash vector, and N is the length of the hash vector. As mentioned in Section 2.4.3, $\{\mathbf{n}(j)\}_{j=1}^{N}$ are i.i.d. Also $\mathbf{t}_i(j)$ are i.i.d. by definition. Using Central Limit theorem for large enough N, $\{\mathbf{v}(i)\}_{i=1}^{M}$ are i.i.d. Gaussian. We can define the probability density function $f(\mathbf{v}/H_0)$ as,

$$f(\mathbf{v}/H_0) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{v_i^2}{2\sigma_0^2}} \tag{C.8}$$

where σ_0^2 is the variance of each of the $\mathbf{v}(i)'s/H_0$. Also note that, $\sigma_0^2 = N\sigma_{0n}^2$, where σ_{0n}^2 denotes the variance of each component of $\mathbf{n}(j)$ of the i.i.d. noise vector $\mathbf{n}$. If we interpret $\mathbf{z} - \mathbf{y}$ as the noise under different images, which is verified experimentally, we can write $f(\mathbf{v}/H_1)as$

$$f(\mathbf{v}/H_1) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{v_i^2}{2\sigma_1^2}} \tag{C.9}$$

where σ_1 is the variance of the each component of the noise vector $\mathbf{z}$ - $\mathbf{x}$. Both $\sigma_0,\ \sigma_1$ can be estimated by across a large number of images and secret keys. Under equal

priors, we can write decision rule as,

$$H_0 : \ if \ f(\mathbf{v}/H_0) \geq f(\mathbf{v}/H_1) \tag{C.10}$$

$$H_1 : \ if \ f(\mathbf{v}/H_0) < f(\mathbf{v}/H_1) \tag{C.11}$$

Using equations (C.8), (C.9), (C.10) and (C.11) we can obtain the decision rule,

$$H_0 \ : \ if \ \|v\|_2 \leq \sqrt{(2M)\left(\log\frac{\sigma_1}{\sigma_0}\right)\left(\frac{\sigma_0^2\sigma_1^2}{\sigma_1^2-\sigma_0^2}\right)}, \ H_1 \ otherwise \tag{C.12}$$

Note that left-side of equation (C.12) is optimum threshold for an attack A whose magnitude is quantified by noise variance σ_0.

With these results in hand, we can now find probability of miss and false alarm. For any $\tau > 0$, the probability of miss $P_M(\tau)$ and false alarm $P_F(\tau)$ can be written as,

$$P_M(\tau) = Pr(\|v\|_2/H_0 > \tau) = Pr(\|v\|_2^2/H_0 > \tau^2) \tag{C.13}$$

$$P_F(\tau) = Pr(\|v\|_2/H_1 > \tau) = Pr(\|v\|_2^2/H_1 < \tau^2) \tag{C.14}$$

Let $\mathbf{v}_0(i) = \mathbf{v}(i)/H_0$ and $\mathbf{v}_1(i) = \mathbf{v}(i)/H_1$. Define, $Y = \|\mathbf{v}_0\|_2^2 = \sum_{i=1}^{M} \mathbf{v}_0(i)^2$. Then, we can rewrite probability of miss as,

$$P_M(\tau) = Pr(Y > \tau^2) = Pr(X > \frac{\tau^2}{\sigma_0^2}) \tag{C.15}$$

where $X = \sum_{i=1}^{M} \hat{\mathbf{v}}_0(i)^2$ and $\hat{\mathbf{v}}_0(i) = \frac{\mathbf{v}_0(i)}{\sigma_0}$. Note that $\{\hat{\mathbf{v}}_0(i)\}_{i=1}^{M}$ are i.i.d. Gaussian with zero mean and unit variance. So X is the sum of the squares of i.i.d. random variables $\sim N(0,1)$, i.e. X is Chi-squared distributed. We can write the density of X as,

$$f_X(x) = \frac{x^{(\frac{M}{2}-1)}}{2^{\frac{M}{2}}\Gamma(\frac{M}{2})}e^{-\frac{x}{2}}, \ x \geq 0 \tag{C.16}$$

where, $\Gamma(a)$ is Gamma function given by,

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1}\ dt \tag{C.17}$$

Then,

$$P_M(\tau) = Pr(X > \frac{\tau^2}{\sigma_0^2}) \;=\; 1 - Pr(X < \frac{\tau^2}{\sigma_0^2}) \tag{C.18}$$

$$\;=\; 1 - \int_0^{\tau^2/\sigma_0^2} \frac{x^{(\frac{M}{2}-1)}}{2^{\frac{M}{2}}\Gamma(\frac{M}{2})} e^{-\frac{x}{2}}\ dx \tag{C.19}$$

Following similar steps, $P_F(\tau)$ is derived as,

$$P_F(\tau) = \int_0^{\tau^2/\sigma_1^2} \frac{x^{(\frac{M}{2}-1)}}{2^{\frac{M}{2}}\Gamma(\frac{M}{2})} e^{-\frac{x}{2}}\ dx \tag{C.20}$$

By looking at (C.19) and (C.20), we can conclude that for a fixed τ, $P_F(\tau)$ is decreasing function with respect to hash length M, whereas $P_M(\tau)$ is increasing.

REFERENCES

1. Menezes, A., V. Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", *CRC Press*, 1998.

2. Cox, I. J., J. Kilian, and F. T. Leighton, "Secure spread spectrum watermarking for multimedia", *IEEE Trans. on Image Processing*, 1996.

3. Xie, L. and G. R. Arce, "A class of authentication digital watermarks for secure multimedia communication", *IEEE Trans. on Image Processing*, 2001.

4. Schneider, M. and S. F. Chang, "A robust content based digital signature for image authentication", *Proc. IEEE Conf. on Image Processing*, 1996.

5. Kailasanathan, C. and R. S. Naini, "Image authentication surviving acceptable modifcations using statistical measures and k-mean segmentation", *Proc. IEEE-EURASIP Work. Nonlinear Sig. and Image*, 2001.

6. Kozat, S. S., R. Venkatesan, and M. K. Mihcak, "Robust Hashing via Matrix Invariances", *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2004.

7. Lu, C. S. and H. Y. M. Liao, "Structural digital signature for image authentication", *IEEE Transactions on Multimedia*, 2003.

8. Fridrich, J. and M. Goljan, "Robust hash functions for digital watermarking", *Proc. IEEE International Conf. on Information Technology: Coding and Computing*, 2000.

9. Monga, V. and B. L. Evans, "Robust perceptual image hashing using feature points", *Proc. IEEE Conf. on Image Processing*, 2004.

10. Dittman, J., A. Steinmetz, and R. Steinmetz, "Content based digital signature for

motion picture authentication and content-fragile watermarking", *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, 1999.

11. Haitsma, J. and T. Kalker, "A highly robust audio fingerprinting system", *Proc. ISMIR*, 2002.

12. Burges, C. J. C., J. C. Platt, and S. Jana, "Extracting noise-robust features from audio data", *Acoustics, Speech, and Signal Processing, Proc. ICASSP*, 2002.

13. Squire, D. M., W. Mullera, H. Mullera, and T. Puna, "Content-based query of image databases", *Pattern Recognition Letters*, 2000.

14. Ke, Y., R. Sukthankar, and L. Huston, "An efficient parts-based near-duplicate and sub-image retrieval system", *IMC*, 2004.

15. Beis, J. S. and D. G. Lowe, "Learning Indexing Functions for 3-D Model-Based Object Recognition", *Proceedings CVPR*, 1994.

16. Belongie, S., J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

17. Monga, V. and K. Mıhçak, "Robust and secure image hashing via non-negative matrix factorization", *IEEE Trans. on Info. Forensics and Security*, 2002.

18. Rivest, R., A. Shamir, and L. Adleman., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, 1978.

19. Jolliffe, I. T., *Principal Component Analysis*, Springer, 2002.

20. Diamantaras, K. I. and S. Y. Kung, *Neural Networks and Principal Component Analysis: Theory and Applications*, John Wiley, NY, 1996.