M.Tech. (Computer Science) Dissertation Series

Uniform Random Sampling from the Distortion less Neighborhood of a Digital Image

A dissertation submitted in partial fulfillment of the requirements for the M.Tech.(Computer Science) degree of the Indian Statistical Institute

By

Nizamuddin Laskar

Under the supervision of

Dr. Palash Sarkar

INDIAN STATISTICAL INSTITUTE
203,Barrackpore Trunk Road
Kolkata-700 108

PROBLEM : UNIFORM RANDOM SAMPLING FROM THE DISTORTIONLESS NEIGHBORHOOD OF A DIGITAL IMAGE.

•***Abstract***: In this paper, we present a new approach for watermark attacking scheme on a single watermarked image using uniform random sampling from the distortion less neighborhood of a digital image. Here, we create a neighborhood of an watermarked image and then we move from given watermarked image (source image) to the target image(destination image) which is as far as possible from the source image. Our goal is to sample uniformly the target image, which is visually as close to the source image from the said neighborhood .

A. Introduction

1. Digital Watermarking

a. Why deal with watermark

b. Watermarking methods

c. Families of watermark attacks

a. Why deal with watermark

- To copy write protection of multimedia digital contents.
- To identify copy write infringer
- To establish ownership

b. Watermarking methods

3 related criteria for watermarking:

Visibility V:

• subjective human evaluation;

• HVS (human visual system)-based computer model;

• PSNR:

Capacity C: given no of bits, typically 64 .. 100.

Robustness R:

• bit error rate;

• binary answer (decision):

- watermark detected;
- watermark not detected.

• ternary answer:

- watermark present & detected,
- watermark present & not detected,
- watermark not present.

c. Families of watermark attacks

Main attack families are :

• geometric → desynchronization, e.g.:

- affine transforms;
- cropping, row/column removal;

- random local distortions;
- mosaicing;
• signal processing → desynchronization, watermark drowning, e.g.:
- lossy compression, (re)quantization, dithering;
- linear, non-linear and adaptive filtering, denoising;
- multiple watermarks, noise addition;
- collage, superimposition;
- **stochastic attacks**;
• specialized, based on knowledge of method:
- **desynchronization attacks**;
- chrominance attack;
- **cryptographic attacks, system-based attacks** (e.g. Oracle, counterfeit original, averaging).

Stirmak: geometric, signal processing.

2. Watermark attacks

a. Why deal with attacks
b. Goals of watermarking attacks
c. General attacks

a. Why deal with attacks

Market is lukewarm towards watermarking technology:
• non-disclosed methods;
• no standard, general purpose benchmarks;
• lack of robustness to attacks.

(Almost) anybody can break a watermark:
• blind use of simple manipulations;
• after study of the methods.

Why work on attacks:
• develop better methods, as with cryptography;
• define better benchmarks.

Pioneering work: Stirmark (benchmarking), Unzign.

b. Goals of watermarking attacks

Notations:
I: source image(watermarked image), size ,N*N
N(I): distortion less neighborhood of I.
I1 • N(I) target image.
I1: **attacked** stego-image.

Main goals of attacks on watermarks:
• preserve image quality:
of I1.

c. General attacks

Goal: general attack on watermark schemes.
The attack:

• takes into account **human visual perception;**
• is **general**- applicable to a wide class of image
Should be used against embedding schemes operating in spatial or transform (DFT, DCT, wavelets) domains.

B.Objectives(goal)

To obtain a generic attack on invisible digital watermark schemes on images. By generic attack, we mean the attack will not consider the details of a particular scheme.
The idea of the attack is as follows:
Let I be the original image and I' is a watermarked image. The image I' is available to the attacker. Let N(I) and N(I') be the set of images in the visually distortion less neighborhood of I and I' respectively. Then we assume that intersections of N(I) and N(I') is very large. Consequently, if we can sample uniformly at random from N(I'), then in effect we are also sampling almost uniformly at random from N(I).Let I'' be the sampled image .Since I'' is sampled almost uniformly at random from N(I) ,we expect the watermarked information introduced into I' will not be present in I''.
Our goal is to remove watermark from the given image without loosing considerable amount of visual quality based on human visual perception and to find the critical values of the parameters of the proposed heuristics.

C. Different attacking schemes

1. ***Heuristic in spatial domain***

Methodology :We have taken a gray scale watermarked digital image and then we replace each pixel intensity value by an another value which is taken as a random value within an interval about the said pixel intensity value. The interval may be fixed or may be different from pixel to pixel .In this way we can create the distortion less neighborhood N(I) of watermarked image(I).If the interval is small the neighborhood will be small else neighborhood will be distorted which we will not expect.
Generally speaking, Human Visual System(HVS) cannot distinguish 2 - 5 % change of pixel intensity value. Taking the advantage of HVS we can apply our method for generic attack.

Case I. In this case we fix the interval length by a constant value.
Algorithm:
(i). Initialize the interval length 2d(k)=constant. where k=(i,j) the position of the pixel value.

(ii).Take a pixel value v(k) from the given the given image & replace it by a random value v'(k) from [v(k)-d(k),v(k)+d(k)].
Do it for all pixel.
(iii). Take the output image.

Results :

Here we get output image which is visually very close to the given image even if the iteration no (large_no) is very large near 60,000, when we choose interval length is less than 10 and if this interval length is more than 20 ,then distortion occurs but does not changes with iteration number..

Case II. In this case we fix the interval length by a constant value and operation is performed on the output image also.

Algorithm:

(i). Initialize the interval length 2d(k)=constant. where k=(i,j) the position of the pixel value and large_no(# of required iteration)
(ii). For each pixel ,find a(k)=v(k)-d(k) and b(k)=v(k)+d(k).
(iii). While (count < large-no) {

- Take a pixel value v(k) from the given the given image & replace it by a random value v'(k) from [a(k),b(k)].
- If(v'(k) < a(k)) set v'(k)=a(k).
- If(v'(k) >ba(k)) set v'(k)=b(k).
- Set v(k)=v(k').
- Count = count+1;

}

(ii).Take a pixel value v(k) from the given the given image & replace it by a random value v'(k) from [v(k)-d(k),v(k)+d(k)].
Do it for all pixel.
(iii). Get the final output image.

Results :

Here we get output image which is visually very close to the given image even if the iteration no (large_no) is very large near 60,000, when we choose interval length is less than 10 and if this interval length is more than 20 ,then distortion begins and increases with iteration number..

Case III. In this case the interval length is not fixed.

Algorithm :

(i).Initialize percentage p by a no taken from 2 to 30.
(ii).Find the interval length k=p.v(k) /100, for each pixel.
(iii). For each pixel ,find a(k)=v(k)-d(k) and b(k)=v(k)+d(k).
(iv). Take a pixel value v(k) from the given the given image & replace it by a random value v'(k) from [v(k)-d(k),v(k)+d(k)].
Do it for all pixel.

(v). Take the output image.
Results :

Case IV. In this case we will not fix the interval length and the operation is performed on the output image.
Algorithm :
(i).Initialize percentage p by a no taken from 2 to 30.
(ii).Find the interval length d(k)=p.v(k) /100, for each pixel.
(iii). For each pixel ,find a(k)=v(k)-d(k) and b(k)=v(k)+d(k).
(iv) while (count < large_no) {
for all pixels

- Take a pixel value v(k) from the given the given image & replace it by a random value v'(k) from [a(k),b(k)].
- If(v'(k) < a(k)) set v'(k)=a(k).
- If(v'(k) >b(k)) set v'(k)=b(k).
- Set v(k)=v(k').
- Count = count+1;

}
(v).Get the final output image.

Results :

Here we get output image which is visually very close to the given image even if the iteration no (large_no) is very large near 60,000, when we choose percentage p is less 5% and if p is more than 5 % ,then after some iteration distortion begins.

2. *Heuristic in Fourier Transform domain:*

Methodology :
In this case we first compute the discrete Fourier transform of the given image (I) .Then we find magnitude spectrum & phase angle corresponding to each complex frequency.
Discrete Fourier Transform **(DFT)** of a digital image $I=[f(r,c)]_{MxN}$ is defined by :
$F(u,v)=(1/M.N) \bullet \bullet f(r,c)\exp[(-2\bullet j(ur/M+vc/N)]$,where
$= R+jI$ $0<=r<=M-1,0<=c<=N-1.$
Corresponding to frequency (u,v) magnitude spectrum **(S)** and phase angle **(•)**are **$S=mod(F(u,V)$ and $\bullet = \arctan(I/R)$.**
To reduce computation time to find **DFT** ,we use Fast Fourier Transform **(FFT)** and for this reason we chose a square image of size **NxN** where $N=2^m$,where m is a positive integer.

Case(I). We keep image phase angle and change the all magnitude spectrum value randomly some percentage of each respective magnitude

spectrum value within some specified range and then we transform back in special domain.

Algorithm:

(i).Compute the FFT of the given image.

(ii).Find the interval length d(k)=p.v(k) /100, for each spectrum value[k=(u,v)].

(iii). For each spectrum value ,find a(k)=v(k)-d(k) and b(k)=v(k)+d(k).

(iv) while (count < large_no) {

for all pixels

- Take a spectrum value v(k) from the given the given image & replace it by a random value v'(k) from [a(k),b(k)].
- If(v'(k) < a(k)) set v'(k)=a(k).
- If(v'(k) >b(k)) set v'(k)=b(k).
- Set v(k)=v(k').
- Count = count+1;

}

(v). Compute inverse fourier transform(**IFFT**) of the output transformed image.

(v).Get the final output image.

Results :

Here we get output image which is visually very close to the given image even if the iteration no (large_no) is large near 5,000, when we choose percentage p is less than 7 % and if p is more than 8 % ,then after some iteration distortion begins.

Case(I). We keep image spectrum value and change the all phase angle value randomly some percentage of each respective phase angle value within some specified range and then we transform back in special domain.

Algorithm:

(i).Compute the FFT of the given image.

(ii).Find the interval length d(k)=p.v(k) /100, for each phase angle value[k=(u,v)].

(iii). For each phase angle value ,find a(k)=v(k)-d(k) and b(k)=v(k)+d(k).

(iv) while (count < large_no) {

for all pixels

- Take a phase angle v(k) from the given the given image & replace it by a random value v'(k) from [a(k),b(k)].
- If(v'(k) < a(k)) set v'(k)=a(k).
- If(v'(k) >b(k)) set v'(k)=b(k).

- Set v(k)=v(k').
- Count = count+1;

}

(v). Compute inverse fourier transform(**IFFT**) of the output transformed image.

(v).Get the final output image.

Results :

Here we get output image which is visually very close to the given image even if the iteration no (large_no) is large near 10,000, when we choose percentage p is less than 20 % and if p is more than 30 % ,then after some iteration distortion begins.

Created neighborhood :

Using the above different types of heuristics ,we can create our target neighborhood. Now ,we pick up a sample image from the created distortion less neighborhood N(I) and will go for testing whether the watermark is present in the attacked image or not.

Existence of watermark is measured by a factor called similarity factor (SIM). If SIM (which is discussed later) is less than 60% then it is declared that the watermark is not present in the TESTED IMAGE.

Maximum value of SIM is 100 %. There is a

A correlation among SIM , count (iteration no) and interval length [d(k)] .

Observation: We have seen that after sampling an image from the said neighborhood most of the cases percentage of watermark is reduced but the percentage is more than 70 percent on an average cases.

That's why we are looking for another hueristic to solve our problem.

3. PRINCIPAL COMPONENT ANALYSIS AND SEPARABILITY MEASURE IN TRANSFORM DOMAIN

OUR GOAL:

Suppose, we have an watermarked image,

$I_m = [\, f(r,c)\,]_{NxN} = I_1$ **(say)**

Our object is (target) to find an image

$I_2 = [\, g(r,c)]_{NxN}$ **which is visually same as I_1 having no watermark.**

That is, I_2 should be as far away from I_1 as possible.

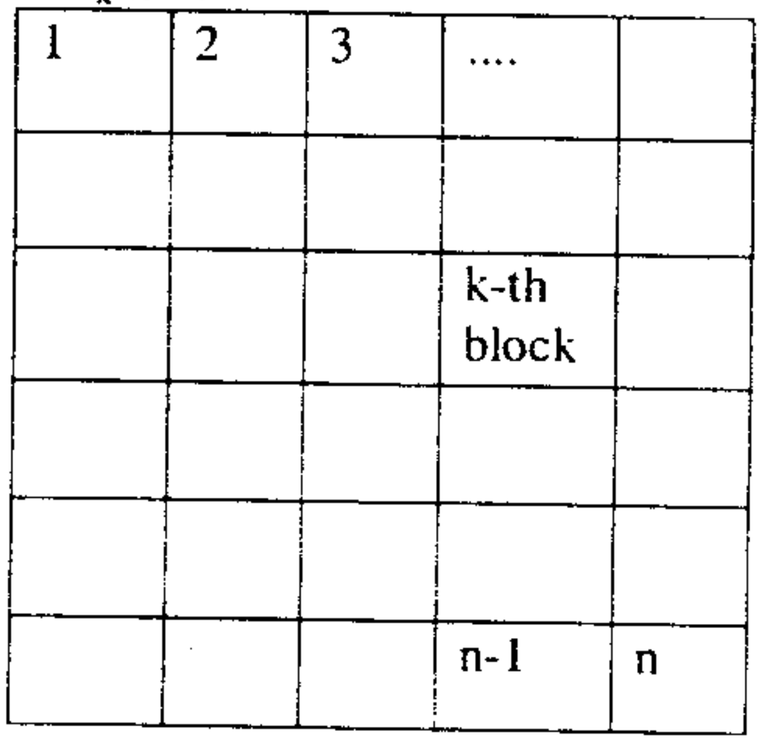
DESCRIPTION :

Multispectral Space , Information Classes and Spectral classes:
In remote sensing image data are modeled by multidimensional normal distributions. The groups or clusters of pixel points are referred to as information classes and they are actual classes of data which a computer will need to be able to recognize.
Information classes are the composed of several sub classes called spectral classes.
Multidimensional data can be represented in order to formulate algorithms for quantitative analysis is to plot them in a pattern space or multispectral vector space.

Feature Reduction and Separability Measure:
For classifiers such as minimum distance procedure is a linear increase of features. Feature selection procedure commonly used to determine the mathematical separability of classes. Feature reduction is performed by checking how separable various spectral classes remain when reduced sets of features are used.

Steps (Algorithmic Steps) :

Step 1: First we divide entire image (I_1) into $n = (N*N)/(m*m)$, number of blocks.



Total # of blocks n = (NxN)/(mxm).

Dimension of each block = mxm

Mean within a block (k-th block) $\mu = (\sum \sum f(r,c)) / M$ where $M=m*m = m^2$

K = (i,j) (say).

$$\mu_k = \mu_{(i,j)} = \left(\sum_{r=i}^{i+m} \sum_{c=j}^{j+m} f(r,c) \right) / (m*m)$$

$= (\Sigma \ \Sigma f(r,c) \ ) / M$

Variance within the K-th block $\sigma_k^2 = \sum_{r=i}^{i+m} \sum_{c=j}^{j+m} [f(r,c) - \mu_k]^2 / M$

Calculate M_k, σ_k^2 $\forall$ k=1,2,.....,n.

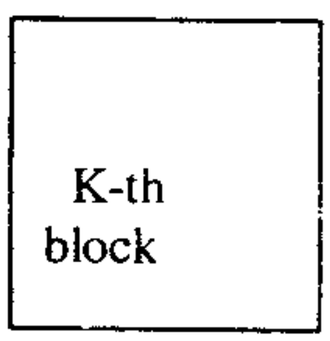
Step 2:

Represent each block by lexicographic ordering of pixels where $M=m^2$.

So,

$\leftarrow \ m \times m \ \rightarrow$

$f(r, c)$



$$\Rightarrow \quad \underset{\sim}{f}_k = \begin{bmatrix} f(i,j) \\ f(i,j+m) \\ f(i+1,j) \\ . \\ . \\ . \\ f(i+m,j) \\ . \\ . \\ f(i+m,j+m) \end{bmatrix} = \begin{bmatrix} f1 \\ f2 \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ f_M \end{bmatrix}$$

where $f_M = f(i+m, j+m)$, $f_1 = f(i, j)$.

$$\Longrightarrow \quad \text{Mean vector of } \mathbf{f_k} \text{ is } \quad \mu_k = \frac{1}{n}\left(\sum_{k=1}^{n} f_k\right) = \begin{bmatrix} \overline{f1} \\ \overline{f2} \\ \cdot \\ \cdot \\ \cdot \\ \overline{f_M} \end{bmatrix}$$

We have represented a set of n two-dimensional discrete signals as a column vectors:

$$\begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_k \\ \cdot \\ \cdot \\ f_n \end{bmatrix}$$

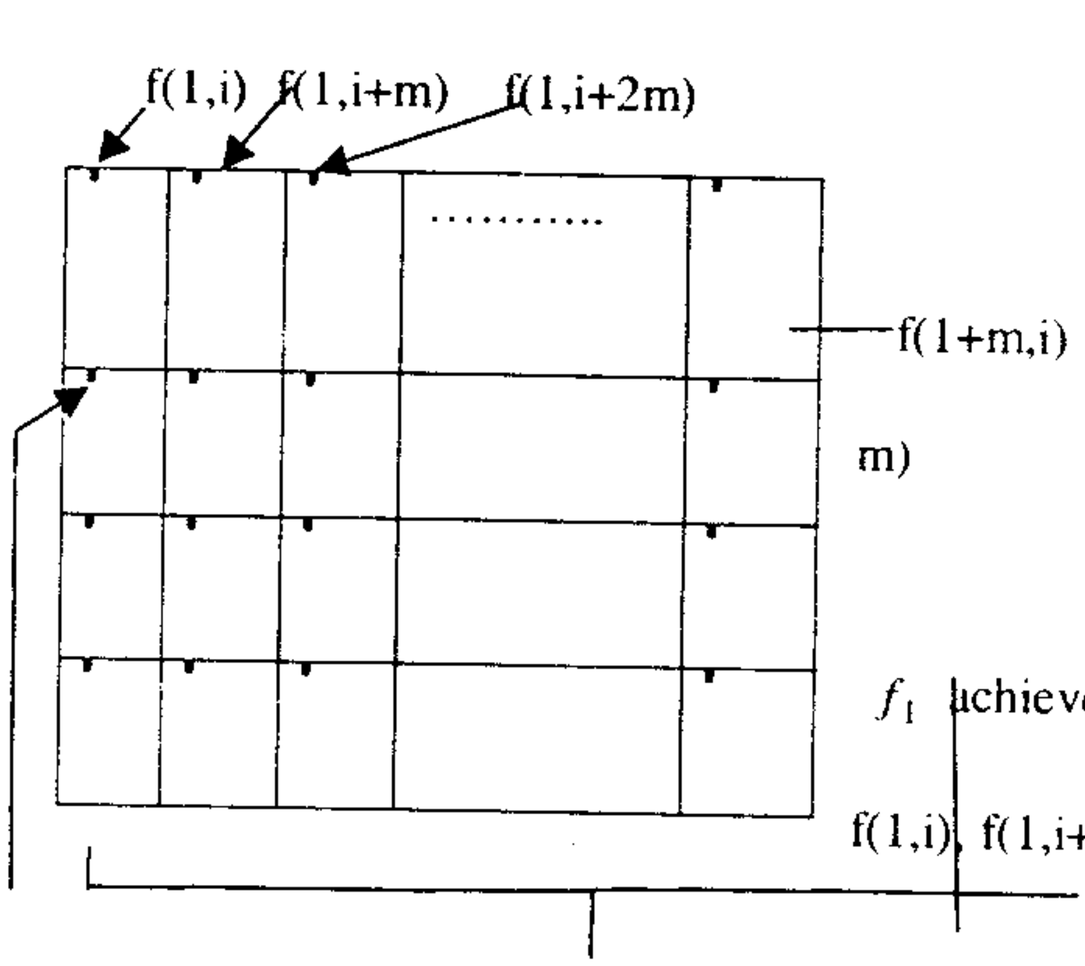
each having M elements whose mean vector is μ_k .

Variance Covariance matrix of above samples :

$$\sum_f = \mathrm{E}\left\{\left(f_k - \mu_k\right)\left(f_k - \mu_k\right)^T\right\}$$

$$= \sum_1$$

It is clear that --



$n = \frac{N_x N}{m_x m}$ no. of pixels.

←

$$\overline{f_i} = \frac{f(1,i) + f(1,i+m) + \ldots + f(1,i+2m) + \ldots + \ldots + f(N-m, N-m)}{n}$$

Similarly, we can find $\overline{f_2}$, $\overline{f_3}$, ..., $\overline{f_M}$

Hence, $\mu_k = \begin{bmatrix} \overline{f}_1 \\ \overline{f}_2 \\ . \\ . \\ \overline{f_M} \end{bmatrix}$ $\Rightarrow$ can be determined

$$\therefore \sum_1 = \mathrm{E}\left\{ \left(f_k - \mu_k \right) \left(f_k - \mu_k \right)^T \right\}$$

$$= \mathrm{E} \begin{bmatrix} (f_1 - \bar{f}_1) \\ (f_2 - \bar{f}_2) \\ . \\ . \\ (f_M - \bar{f}_M) \end{bmatrix} \left[(f_1 - \overline{f_1})(f_2 - \overline{f_2}) \ldots (f_M - \overline{f_M}) \right]$$

$$= \begin{bmatrix} E\{(f_1 - \overline{f_1})(f_1 - \overline{f_1})\} \ldots\ldots\ldots\ldots E\{(f_1 - \overline{f_1})(f_M - \overline{f_M})\} \\ E\{(f_1 - \overline{f_1})(f_2 - \overline{f_2})\} \ldots\ldots\ldots\ldots E\{(f_2 - \overline{f_2})(f_M - \overline{f_M})\} \\ . \\ . \\ . \\ E\{(f_1 - \overline{f_1})(f_M - \overline{f_M})\} \ldots\ldots\ldots\ldots E\{(f_M - \overline{f_M})(f_M - \overline{f_M})\} \end{bmatrix}$$

Our target is to search an image whose variance covariance matrix $\sum_2\left(\hat{\sum}_2\right)$ and mean vector M_2 are so that Bhattyacharya distance between two classes $\begin{Bmatrix} N_x\left(M_1, \hat{\sum}_1\right) \\ N_x\left(M_2, \hat{\sum}_2\right) \end{Bmatrix}$ becomes maximum.

$$d = \frac{1}{8}(M_2 - M_1)^T \left(\frac{\hat{\sum}_1 + \hat{\sum}_2}{2}\right)^{-1} (M_2 - M_1) + \frac{1}{2}\ln\left\{\frac{\left|\hat{\sum}_1 + \hat{\sum}_2\right|}{\sqrt{\left|\hat{\sum}_1\right|\left|\hat{\sum}_2\right|}}\right\}$$

$$= \mu\left(\frac{1}{2}\right) \qquad \text{----------(1)}$$

$$= \mu_1 + \mu_2$$

$$\approx \frac{1}{8}(M_2 - M_1)^T \left(\frac{\sum_1 + \sum_2}{2}\right)^{-1} (M_2 - M_1) + \frac{1}{2}\ln\left\{\frac{\left|\frac{\sum_1 + \sum_2}{2}\right|}{\left|\sum_1\right|\left|\sum_2\right|}\right\}$$

Since $\hat{\sum}_1 \approx \sum_1$ and $\hat{\sum}_2 \approx \sum_2$ when n = no of sample is very large.

$$\mu(s) = \frac{s(1-s)}{2}(M_2 - M_1)^T \left[s\sum_1 + s(1-s)\sum_2\right]^{-1} + \frac{1}{2}\ln\left\{\frac{\left|s\sum_1 + s(1-s)\sum_2\right|}{\left|\sum_1\right|^s \left|\sum_2\right|^{1-s}}\right\} \qquad \text{----------(2)}$$

This $\mu(s)$ is called Chernoff distance.

When s=1/2 . Chernoff distance Becomes Bhattyacharya distance .

Bhattyacharya distance between two normal classes $N_x(0, I)$ and $N_x(0,\wedge)$, (here, $\sum_1 = I$ and $\sum_2 = \wedge$).

$$d = \mu_1 + \mu_2$$

$$= \mu_2 \qquad \text{since } \mu_1 = 0$$

$$= \frac{1}{2}\ln\left\{\frac{\left|\frac{\sum_1 + \sum_2}{2}\right|}{\sqrt{|\sum_1||\sum_2|}}\right\}$$

$$= \frac{1}{2}\ln\left\{\frac{\left|\frac{I + \wedge}{2}\right|}{\sqrt{|I||\wedge|}}\right\} \quad , \qquad \text{where } \wedge = \begin{bmatrix} \lambda_1 0\ 0\ ....0 \\ 0\ \lambda_2 0\ ....0 \\ . \\ . \\ 0\ 0\ 0\ ...\lambda_n \end{bmatrix}, \text{the eigen value matrix of } \bullet$$

$$= \frac{1}{2}\ln\left\{\frac{\left|\frac{1+\lambda_1}{2}\right|\left|\frac{1+\lambda_2}{2}\right|\cdots\left|\frac{1+\lambda_n}{2}\right|}{\sqrt{1.\lambda_1.\lambda_2...\lambda_n}}\right\}$$

$$= \frac{1}{2}.\sum_{i=1}^{n}\ln\left(\frac{1+\lambda_i}{2\sqrt{\lambda_i}}\right) \qquad \text{---------- (3)}$$

Orthogonal Transformation :

$\phi^T \phi = I.$

$Y = \phi^T X.$

$\phi^T \sum_x \phi = \wedge$

$$= \begin{bmatrix} \lambda_1 0\ 0\ ....0 \\ 0\ \lambda_2 0\ ....0 \\ . \\ . \\ 0\ 0\ 0\ ....\lambda_n \end{bmatrix}$$

$$\sum_1 = \begin{bmatrix} \sigma_{11}^2\ \sigma_{12}^2\ \sigma_{13}^2 .....\sigma_{1n}^2 \\ \sigma_{21}^2\ \sigma_{22}^2\ \sigma_{23}^2 .....\sigma_{2n}^2 \\ \sigma_{11}^2\ \sigma_{12}^2\ \sigma_{13}^2 .....\sigma_{1n}^2 \\ . \\ . \\ \sigma_{n1}^2\ \sigma_{n2}^2\ \sigma_{n3}^2 .....\sigma_{nn}^2 \end{bmatrix}$$

Where Variance and covariance are as follows

$$\sigma_{ii}^2 = E\left\{\left(f_i - \overline{f_i}\right)\left(f_i - \overline{f_i}\right)\right\}$$

$$\sigma_{ij} = E\left\{\left(f_i - \overline{f_i}\right)\left(f_j - \overline{f_j}\right)\right\} = \sigma_{ji}$$

Similarly $\sum_2$ is of the form of that of $\sum_1$.

If $\hat{\sum}_1$ is the biased estimate of $\sum_1$.

Then

$$\hat{\sum}_i = \frac{1}{n-1} \mathrm{E}\left\{ \left(f_k - \mu_k \right) \left(f_k - \mu_k \right)^T \right\}$$

$$= \frac{n}{n-1} \cdot \frac{1}{n} \mathrm{E}\left\{ \left(f_k - \mu_k \right) \left(f_k - \mu_k \right)^T \right\}$$

$$\hat{\sum}_i = \frac{n}{n-1} \cdot \sum_i$$

If n is large then

$$\hat{\sum}_i \approx \sum_i \text{ ----------} > (4)$$

Let us assume the samples $f_1, f_2, f_3, \ldots\ldots\ldots\ldots f_n$ follows normal distribution

$$N_x\left(M_1, \hat{\sum}_1 \right) = \frac{1}{(\sqrt{2\Pi})^n \left| \hat{\sum}_1 \right|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} d^2(X) \right\}$$

$$d^2(X) = (X - M1)^T \hat{\sum}_1^{-1} (x - M_1)$$

$$= Tr\left\{ \hat{\sum}_1^{-1} (x - M_1)(x - M_1)^T \right\} \text{ ----------} > (5)$$

This $d^2(X)$ is called distance function.

$\therefore \sum_1 \phi = \phi \wedge$ and $\phi^T \phi = I$ ----------> (6)

Then $\sum_1$ and $\sum_2$ are transformed to

$$A^T \sum_1 A = \left(\phi \wedge^{-\frac{1}{2}} \right)^T \sum_1 \left(\phi \wedge^{-\frac{1}{2}} \right)$$

$$= \wedge^{-\frac{1}{2}} \left(\phi^T \sum_1 \phi \right) \wedge^{-\frac{1}{2}}$$

$$= \wedge^{-\frac{1}{2}} (\wedge) \wedge^{-\frac{1}{2}}$$

$$= I \text{ ----------} > (8)$$

and

$$A^T \sum_2 A = \left(\phi \wedge^{-\frac{1}{2}} \right)^T \sum_2 \left(\phi \wedge^{-\frac{1}{2}} \right)$$

$$= \wedge^{-\frac{1}{2}} \left(\phi^T \sum_2 \phi \right) \wedge^{-\frac{1}{2}}$$

$$= \text{K (say)} \text{ ----------} > (9)$$

Now ,

$K^T = A^T \sum_2 {}^T A = A^T \sum_2 A$

Hence K is symmetric.

(ii) Secondly, we can apply an orthogonal transformation (magnitude of eigen vector is unity)

$Z = \psi^T Y$

If ψ and θ are eigen vector and eigen value matrices of K then

$K\psi = \psi\theta$ and $\psi^T \psi = I$ ----------> (9)

Then I and K are transformed to

$\psi^T I \psi = \psi^T \psi = I$, as $\psi^T \psi = \text{I}$ and

$$\psi^T K \psi = \theta = \begin{bmatrix} \mu_1 0 \ldots\ldots 0 \\ 0 \mu_2 \ldots\ldots 0 \\ . \\ . \\ 0 \ldots\ldots\ldots \mu_n \end{bmatrix} \text{ -------} > (10)$$

is the eigen value matrix of K, where

$K = \psi\theta\psi^T$.

* Bhattyacharya distance between two classes $N_X(\mu_1, \sum_1)$ and $N_X(\mu_2, \sum_2)$ is invariant under the transformation when simultaneous diagonalization of two matrices $\sum_1$ and $\sum_2$ (both are symmetric matrix) are required.

This distance d=μ (1/2)

$$=\frac{1}{8}(M_2 - M_1)^T \left(\frac{\sum_1 + \sum_2}{2} \right)^{-1} (M_2 - M_1) + \frac{1}{2} \ln \left\{ \frac{\left| \frac{\sum_1 + \sum_2}{2} \right|}{|\sum_1 \| \sum_2|} \right\} == \mu_1 + \mu_2$$

$$\sum_Y = E\{(Y - \mu_Y)(Y - \mu_Y)^T\}$$
$$= E\{(\phi^T X - \phi^T \mu_X)(\phi^T X - \phi^T \mu_X)^T\}$$
$$= \phi^T E\{(X - \mu_X)(X - \mu_X)^T\}\phi$$
$$= \phi^T \sum_X \phi$$
$$= \wedge$$
$$= \begin{bmatrix} \lambda_1 & 0 & 0 & .....0 \\ 0 & \lambda_2 & 0 & .....0 \\ ..... & ..... & ..... & ..... \\ 0 & 0 & ..... & .....\lambda_n \end{bmatrix}$$

$$\wedge^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 & 0 & .....0 \\ 0 & \frac{1}{\lambda_2} & 0 & .....0 \\ ..... & & & \\ 0 & 0 & ..... & \frac{1}{\lambda_n} \end{bmatrix}$$

$$\left(\wedge^{-1}\right)^{\frac{1}{2}} = \wedge^{-\frac{1}{2}}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & 0 & .....0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} & 0 & .....0 \\ ..... & & & \\ 0 & 0 & ..... & \frac{1}{\sqrt{\lambda_n}} \end{bmatrix}$$

Whitening Transformation:

Let us consider the transformation $Y = A^T X$ where A=• • $^{-1/2}$

$$\therefore \sum_Y = A^T \sum_X A$$

$$= \left(\phi \wedge^{-\frac{1}{2}}\right)^T \sum_X \left(\phi \wedge^{-\frac{1}{2}}\right)$$

$$= \wedge^{-\frac{1}{2}} \left(\phi^T \sum_X \phi\right) \wedge^{-\frac{1}{2}} \quad \textit{where} \quad \phi \quad \textit{is} \quad \textit{defined} \quad \textit{above}$$

$$= \wedge^{-\frac{1}{2}} \wedge \quad \wedge^{-\frac{1}{2}}$$

$$= I$$

1) **Whitening transformations are not orthogonal transformation because**

$A^T A$ *where* $A = \phi \wedge^{-\frac{1}{2}}$, $\phi^T \phi = I$, and

$$\wedge = \begin{bmatrix} \lambda_1 & 0 & 0 & .....0 \\ 0 & \lambda_2 & 0 & .....0 \\ ..... & ..... & ..... & ..... \\ 0 & 0 & ..... & .....\lambda_n \end{bmatrix} \neq I$$

$$= \wedge^{-\frac{1}{2}} \phi^T \phi \wedge^{-\frac{1}{2}}$$
$$= \wedge^{-1}$$
$$\neq I$$

2) After whitening transformation the variance matrix is invariant under any orthogonal transformations

$\psi^T I \psi = \psi^T \psi = I$ *where* $A^T \sum A = I$

$$A = \phi \wedge^{-\frac{1}{2}}$$

<u>Simultaneous Diagonalisation:</u>

We can diagonalize two symmetric matrices $\sum_1 and \sum_2$ simultaneously by a linear transformation.

(i) First, we whiten $\sum_1$ by $A = \phi \wedge^{-\frac{1}{2}}$

$Y = A^T X$, *Where* $\phi^T \phi = I$

ϕ is Eigen vector matrix of $\sum_1$

$\wedge$ is Eigen value matrix of $\sum_1$

Then $\sum_1$ becomes $A^T \sum_1 A = I$ .and $\sum_2$ becomes $A^T \sum_2 A$ =K (say)

(ii) Now , we can diagonalize $\sum_2$ by an orthonormal transformation

$\bullet^T K \bullet = \bullet$, where $\bullet^T \bullet = I$.

Hence we can write $|K - \bullet_i| = 0$ ----------(II)

is the eigen equation for K , $\bullet_i$'s are eigen value of K and $\bullet$ *is the* eigen value matrix of K.

Hence resultant transformation for simultaneous diagonalization is

$$A = \phi \wedge^{-\frac{1}{2}} \psi$$

Where $\mu_1 = \frac{1}{8}(M_2 - M_1)^T \left(\frac{\sum_1 + \sum_2}{2} \right)^{-1} (M_2 - M_1)$

and $\mu_2 = \frac{1}{2} \log \left\{ \frac{\left| \frac{\sum_1 + \sum_2}{2} \right|}{\sqrt{|\sum_1| + |\sum_2|}} \right\}$

Let the transformation is

$A = \phi \wedge^{-\frac{1}{2}} \psi$ **(The notations have their usual meaning)**

X-space	Y-space	Transformed space
M_1, $\sum_1$	M_1, $\sum_1$	
M_2, $\sum_2$	M_2, $\sum_2 = \theta$	$A^{-1} = \psi^{-1} \wedge^{\frac{1}{2}} \phi^{-1}$ $= \psi^T \wedge^{\frac{1}{2}} \phi^T$
M_1, μ_2	M_1, μ_2	
μ	μ	

We have ,when A **is a transformation matrix like** $A = \phi \wedge {}^{2}\psi$

$$A^T \sum_1 A = I$$

$$A^T \sum_2 A = \theta$$

$$A^T \left(\sum_1 + \sum_2 \right) A = I + \theta$$

$$\Rightarrow \quad A^T \left(\frac{\sum_1 + \sum_2}{2} \right) A = \frac{I + \theta}{2} \qquad \text{----------} > \quad (1)$$

$$\Rightarrow \quad \frac{\sum_1 + \sum_2}{2} = \left(A^T \right)^T \left(\frac{I + \theta}{2} \right) A^{-1}$$

$$\Rightarrow \quad \left(\frac{\sum_1 + \sum_2}{2} \right)^{-1} = A \left(\frac{I + \theta}{2} \right)^{-1} A^T$$

Hence,

$$\mu_1 = \frac{1}{8} (M_2 - M_1)^T \left(\frac{\sum_1 + \sum_2}{2} \right)^{-1} (M_2 - M_1)$$

$$= \frac{1}{8} (M_2 - M_1)^T A \left(\frac{I + \theta}{2} \right)^{-1} A^T (M_2 - M_1)$$

$$= \frac{1}{8} \left(A^T M_2 - A^T M_1 \right)^T \left(\frac{I + \theta}{2} \right)^{-1} \left(A^T M_2 - A^T M_1 \right)$$

$$= \frac{1}{8} \left(M_2^{'} - M_1^{'} \right)^T \left(\frac{I + \theta}{2} \right)^{-1} \left(M_2^{'} - M_1^{'} \right)$$

$$= \mu_1^{'} \qquad \left(transformed \quad dis\tan ce \quad for \quad first \quad part \right)$$

$$\mu_2 = \frac{1}{2}\log\left\{\frac{\left|\frac{\sum_1+\sum_2}{2}\right|}{\sqrt{|\sum_1||\sum_2|}}\right\}$$

$$\mu_2 = \frac{1}{2}\log\left\{\frac{\left|\frac{I+\theta}{2}\right|}{\sqrt{|I||\theta|}}\right\}$$

$$= \frac{1}{2}\log\left\{\frac{\left|\frac{I+\theta}{2}\right|}{\sqrt{|\theta|}}\right\}$$

Now from (1) $A^T\left(\frac{\sum_1+\sum_2}{2}\right)A=\frac{I+\theta}{2}$ yields

$$\left|A^T\left(\frac{\sum_1+\sum_2}{2}\right)A\right|=\left|\frac{I+\theta}{2}\right|$$

$$\Rightarrow \quad |A^T||A|\left|\frac{\sum_1+\sum_2}{2}\right|=\left|\frac{I+\theta}{2}\right| \qquad \text{------>} \quad (2)$$

Now, since

$$A=\phi\wedge^{-\frac{1}{2}}\psi$$

$$\Rightarrow \quad A^T=\psi^T\wedge^{-\frac{1}{2}}\phi^T$$

and $\quad |A|=|\phi||\wedge^{-\frac{1}{2}}||\psi|$

$$\Rightarrow \quad |A^T|=|\psi^T||\wedge^{-\frac{1}{2}}||\phi^T|$$

$$\therefore \quad |A^T||A|=|\phi^T||\phi||\wedge^{-1}||\psi^T||\psi|$$

Now, $\phi^T\phi = I \quad \Rightarrow \quad |\phi^T||\phi| = 1$

And, $\psi^T\psi = I \quad \Rightarrow \quad |\psi^T||\Psi| = 1$

$\therefore \quad |A^T||A| = |\wedge^{-1}| = \dfrac{1}{|\wedge|} = |\wedge|^{-1}$

Hence (2) yields

$$|\wedge|^{-1}\left|\frac{\sum_1 + \sum_2}{2}\right| = \left|\frac{I+\theta}{2}\right|$$

$$\Rightarrow \quad \left|\frac{\sum_1 + \sum_2}{2}\right| = \left|\frac{I+\theta}{2}\right||\wedge|$$

Also, $A^T\sum_1 A = I$ yields

$$|A^T|\left|\sum_1\right||A| = 1$$

$$\Rightarrow \quad |\wedge|^{-1}\left|\sum_1\right| = 1$$

$$\Rightarrow \quad \left|\sum_1\right| = |\wedge|$$

and $A^T\sum_2 A = \theta$ helps to derive the following –

$$|A^T||A|\left|\sum_2\right| = |\theta|$$

$$\Rightarrow \quad |\wedge|^{-1}\left|\sum_2\right| = |\theta|$$

$$\Rightarrow \quad \left|\sum_2\right| = |\wedge||\theta|$$

$$\therefore \quad \mu_2 = \frac{1}{2}\log\left\{\frac{\left|\frac{I+\theta}{2}\right||\wedge|}{\sqrt{|\wedge||\wedge||\theta|}}\right\}$$

$$= \frac{1}{2}\log\left\{\frac{\left|\frac{I+\theta}{2}\right|}{|\theta|}\right\}$$

$$= \mu_2$$

$$\therefore \quad \mu = \mu_1 + \mu_2$$

$$= \mu_1 + \mu_2 \qquad (Distance \quad is \quad invariant)$$

Hence, Bhattacharyya distance $d = \mu_1 + \mu_2 = \mu_1' + \mu_2'$ ***,which is invariant after simultaneous diagonalization.***

Then we can diagonalize two symmetric matrices as
$A^T \sum_1 A = I$ and $A^T \sum_2 A = \theta$

Where $A = \phi \wedge^{-\frac{1}{2}} \bullet$ and the notations have their usual meanings.

and, ϕ and A are the Eigen value and Eigen vector matrices of $(\sum_1{}^{-1} \sum_2)$,
Where $\left(\sum_1{}^{-1} \sum_2\right) A = A\theta$

Bhattacharyya distance,

$$d = \mu\left(\tfrac{1}{2}\right) = \mu_1 + \mu_2 = \mu_1' + \mu_2'$$

$$= \frac{1}{8}\left(M_2' - M_1'\right)^T \left(\frac{I+\theta}{2}\right)^{-1} \left(M_2' - M_1'\right) + \frac{1}{2}\ln\left(\frac{\left|\frac{I+\theta}{2}\right|}{\sqrt{|\theta|}}\right)$$

where $\frac{I+\theta}{2} = \begin{bmatrix} \frac{1+\lambda_1}{2} & 0 & 0.....0 \\ 0 & \frac{1+\lambda_2}{2} & 0.....0 \\ ..... & & \\ 0 & 0 & ....\frac{1+\lambda_i}{2}...0 \\ ..... & & \\ 0 & 0 & .....\frac{1+\lambda_n}{2} \end{bmatrix}$

$M_1 = \begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \\ . \\ . \\ . \\ \bar{f}_M \end{bmatrix}$, mean vector in spatial domain corresponding to $\bullet_1$

$M'_1 = \begin{bmatrix} \bar{f}'_1 \\ \bar{f}'_2 \\ . \\ . \\ . \\ \bar{f}'_M \end{bmatrix}$, mean vector in transformed domain corresponding to $\bullet_1$

$$M_2 = \begin{bmatrix} \bar{g}_1 \\ \bar{g}_2 \\ \cdot \\ \cdot \\ \cdot \\ \bar{g}_M \end{bmatrix}$$, mean vector in spatial domain corresponding to $\bullet_1$.

$$M'_2 = \begin{bmatrix} \bar{g}'_1 \\ \bar{g}'_2 \\ \cdot \\ \cdot \\ \cdot \\ \bar{g}'_M \end{bmatrix}$$, mean vector in transformed domain corresponding to $\bullet_2$

$$\text{So, } d = \frac{1}{8}\sum_{i=1}^{n} \frac{\left(\bar{g}_i - \bar{f}_i\right)\left(\bar{g}_i - \bar{f}_i\right)}{\frac{1+\lambda_i}{2}} + \frac{1}{2}\ln \frac{\prod_{i=1}^{n}\left(\frac{1+\lambda_i}{2}\right)}{\sqrt{\prod_{i=1}^{n}\lambda_i}}$$

$$= \frac{2}{8}\sum_{i=1}^{n} \frac{\left(\bar{g}_i - \bar{f}_i\right)^2}{(1+\lambda_i)} + \frac{1}{2}\sum_{i=1}^{n} \ln \frac{(1+\lambda_i)}{2\sqrt{\lambda_i}}$$

$$= \frac{1}{4}\sum_{i=1}^{n}\left[\frac{\left(\bar{g}_i - \bar{f}_i\right)^2}{(1+\lambda_i)} + 2\ln\frac{(1+\lambda_i)}{2\sqrt{\lambda_i}}\right]$$

We want to maximize d. For any fixed value of λ_i, d will be maximum when each $\left(\bar{g}_i - \bar{h}_i\right)$ is maximum.

Lemma : We can diagonalize two symmetric matrices Σ_1 & Σ_2 as

$$A^T\Sigma_1 A = I \ \& \ A^T\Sigma_2 A = \theta$$

Where $A = \phi\Lambda^{-1/2} \Rightarrow$ the notations have their usual meanings. θ and A are the eigen value and eigen vector matrices of $\Sigma_1^{-1}\Sigma_2$ i.e

$$(\Sigma_1^{-1}\Sigma_2)A = A\theta$$

Proof : From equation (II) θ is the eigen value matrix of K

$$|K - \mu I| = 0 \ where \ \theta = \begin{pmatrix} \mu_1 & 0 & 0 & . & . & . & 0 \\ 0 & \mu_2 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & 0 & \mu_n \end{pmatrix}$$

$\Rightarrow |A^T\Sigma_2 A - \mu_i A^T\Sigma_1 A| = 0$
$\Rightarrow |A^T(\Sigma_2 - \mu_i\Sigma_1)A| = 0 : \ K = A^T\Sigma_2 A$
$\Rightarrow |A^T||(\Sigma_2 - \mu_i\Sigma_1)||A| = 0 : \ I = A^T\Sigma_1 A$
$\Rightarrow |\Sigma_2 - \mu_i\Sigma_1| = 0$
$\Rightarrow |\Sigma_1(\Sigma_1^{-1}\Sigma_2 - \mu_i I)| = 0$
$\Rightarrow |\Sigma_1||\Sigma_1^{-1}\Sigma_2 - \mu_i I| = 0$
$\Rightarrow |\Sigma_1^{-1}\Sigma_2 - \mu_i I| = 0, \ when |\Sigma_1 \neq 0|$
Hence μ_i are the egien values of $\Sigma_1^{-1}\Sigma_2$.
Hence eigen value matrix

$$\theta = \begin{pmatrix} \mu_1 & 0 & 0 & . & . & . & 0 \\ 0 & \mu_2 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & 0 & \mu_n \end{pmatrix}$$

Now $\Sigma_1^{-1}\Sigma_2 A = AA^T(A^T)\theta = AI\theta = A\theta$
Where $A^T\Sigma_1 A = I$, $\Sigma_1 = (A^T)^{-1}A^{-1} \Rightarrow \Sigma_1^{-1} = AA^T$
& $A^T\Sigma_2 A = \theta \Rightarrow \Sigma_2 A = (A^T)^{-1}\theta$
Hence A is the eigen vector matrix of $\Sigma_1^{-1}\Sigma_2$.
Application : Let us see how we can apply the above equations to solve our problem.

Let us consider n=no. of feature of the given image.We choose n such that determinent of variance covariance (Σ) matrix for image I should not zero.

Important observation :

From our experimental observation we have seen that from the distortionless neighborhood N(I), of watermarked image (I) contains I_0 including other images (say I').

Host Image, I_0	Watermarked Image,	Attacked Image, I_0
Variance-ovariance matrix $\sum_0$	Variance -covariance matrix $\sum_1$	Variance -covariance matrix $\sum_2$

First component of Bhattacharyya distance $\mu_1 = \mu_1'$ is very small as compared to 2nd component of the distance

$\mu_2 = \mu_2'$.

That is, ratio of this two distance $R = \mu_2 / \mu_1$ ***is very large***.On the other hand if we pick up uniformly a sample image from N(I) , most of the cases the said ratio $R < R_0$ (for host image $R_0 = \mu_2 / \mu_1$).

Now, if we can create the our required neighborhood of I ,N(I) in such a way that the ratio R becomes near the critical value $=R_{cr}$ (say) which is of order $10^{7\text{-}8} = R_{cr}$ (say) for an image of size 256x256 (gray level image). Now ,if we pick up the sample image from this N(I) ,most of the cases we get success. The chances of watermark present in the N(I) is considerably very less. Using random search technique we can find an Image I' from N(I) whose R value is very large.

PROBABILISTIC OPTIMISATION TECHNIQUE

Algorithm : [In spatial Domain]

i) Initialize percentage p=5 (roughly), and distance ratio $R_{max}=0$, count =0.

ii) Find the interval length $k = (p*v_k)/100$, for each pixel value.

iii) Find $\alpha_k = v_k - d_k$, $\beta_k = v_k + d_k$ for all pixel where $v_k \in I$ (watermarked image)

iv) While (count < large_no) {

a) Take any pixel value v_k from the given image (I) and replace it by a random value v'_k from $[\alpha_k, \beta_k]$

b) If $(v'_k < \alpha_k)$ set $v'_k = \alpha_k$
If $(v'_k > \beta_k)$ set $v'_k = \beta_k$

c) Compute Σ_1 , Σ_2 for both I and I' respectively.

d) Compute $\mu_1 = \mu 1'$ and $\mu_2 = \mu_2'$

e) Compute distance ratio $R = \mu_2 / \mu_1$.

f) If $(R > R_{max})$ {
Set $R_{max} = R$.
$v_k = v'_k$
count = 0
}
else
count = count + 1

g) Get the output image.

N.B.- Here we have applied our heuristics on A ROBUST BLOCK ORIENTED WATERMARKING SCHEME IN SPATIAL DOMAIN which is implemented by Tanmoy Kanti Das and Subhamoy Maitra, Computer Vision and Pattern Recognition Unit, ISI,Kolkata-700108 .

Results ::

In the said spatial domain watermarking scheme amount of watermark is detected by a factor called similarity factor (SIM) .Value of SIM is varied from 50% to 100%.If the value of SIM is less than 60 % then it is assumed that the watermark is not present in the **TESTED IMAGE.**
Using the above ***HEURISTICS*** we have got output image as our *TARGET* image and most of them do not contain any watermark (i.e., SIM factor becomes less than 60%).

TESTED IMAGES ARE SHOWN BELOWN ::

Image before watermarking
(Host Image)

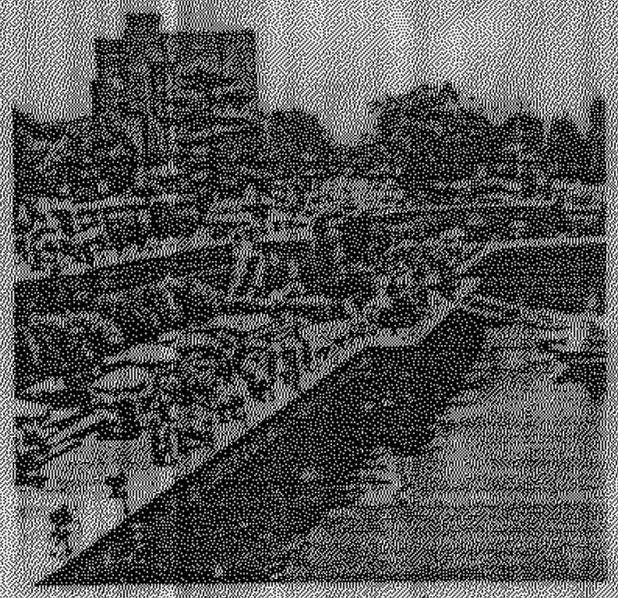


Fig :victoria

Image after watermarking
(Watermarked Image)

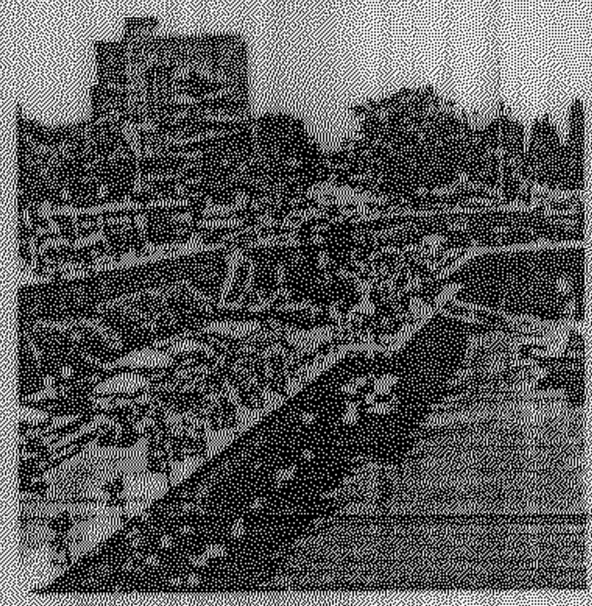


Fig : wvictoria

ALL IMAGE ARE TAKEN OF SIZE 256x256.

Image after attack(Attacked Image)

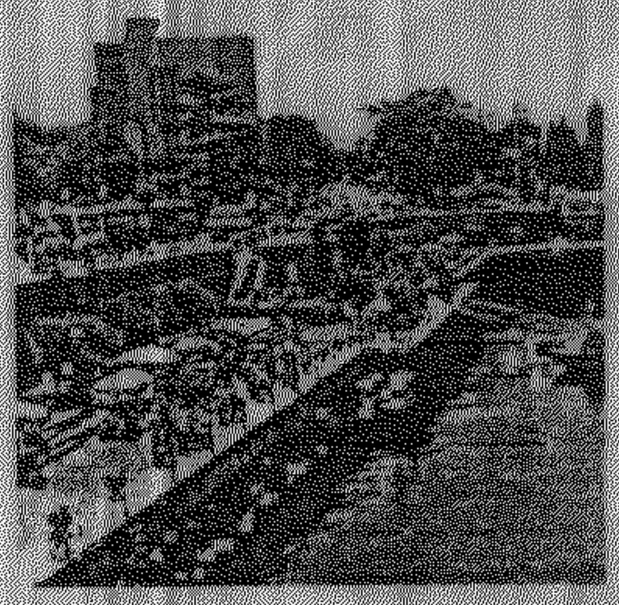


Fig: Attacked victoria

Bhattacharya distance, d=0.35566.

Bhattacharya distance ratio R= 4.1770 x 10^{-8}.

SIM FACTOR = 0.538346

Image before watermarking (Host Image)

Image after watermarking (Watermarked Image)

Image after attack(Attacked Image)

Bhattacharya distance, d= 1.2560.

Bhattacharya distance ratio R=1.2611x 10^{8}.

SIM FACTOR = 0.59634

Image before watermarking (Host Image)

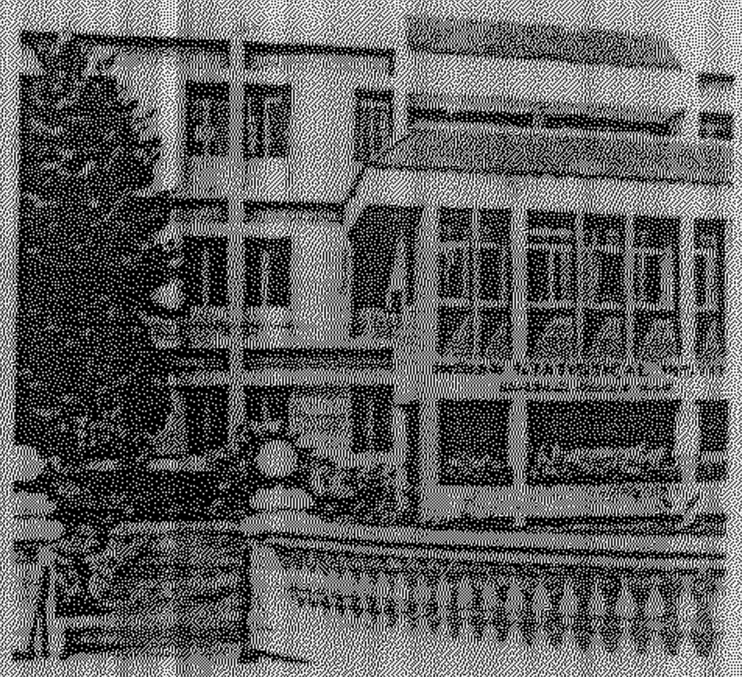


Image after watermarking (Watermarked Image)

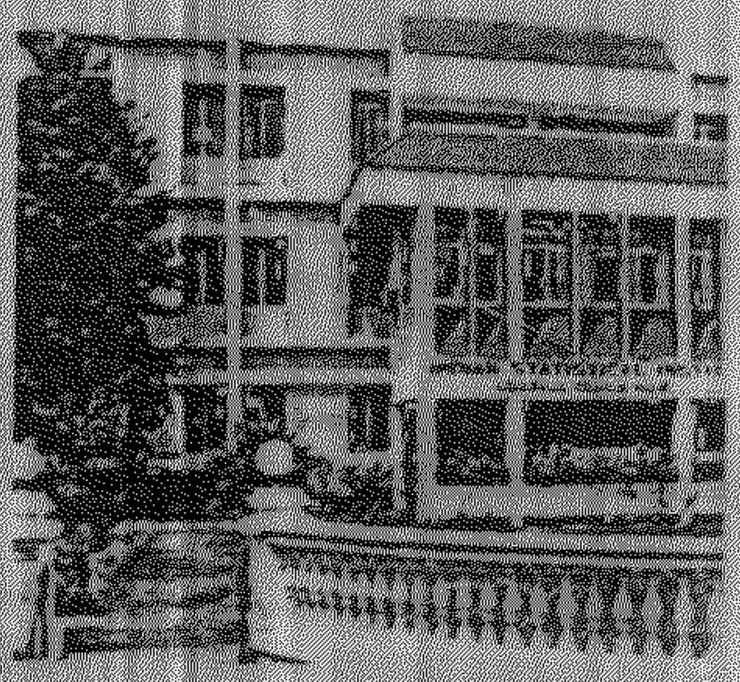
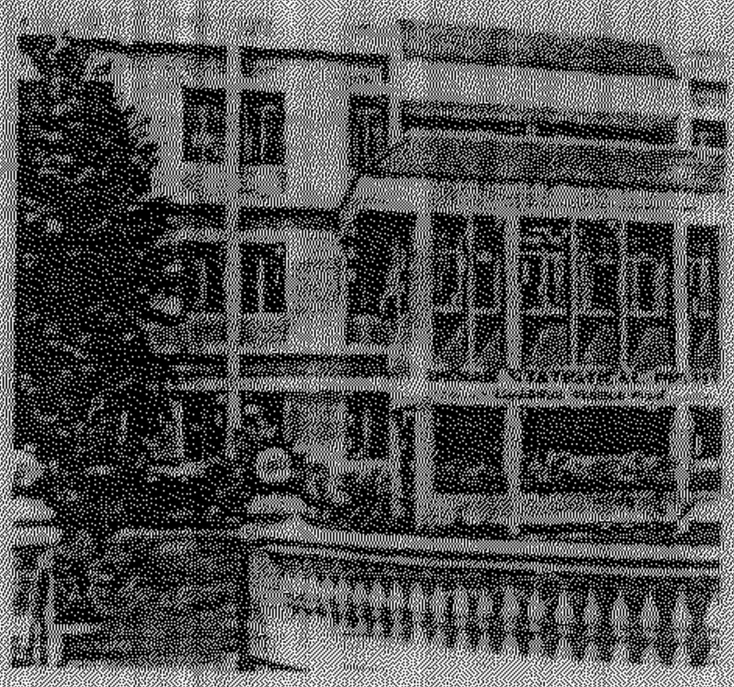


Image after attack(Attacked Image)



Bhattacharya distance, d= 0.14787

Bhattacharya distance ratio R= 2.9711x 10 ^{10}.

SIM FACTOR = 0.582105

Image after attack(Attacked Image)

Bhattacharya distance, d= 0.085827

Bhattacharya distance ratio R= 2.5182x 10^{9}.

SIM FACTOR = 0.592105

Image after attack(Attacked Image)

Bhattacharya distance, d= 0.019492.

Bhattacharya distance ratio R= 3.9566e x 10^{8}.

SIM FACTOR = 0.600752

Image before watermarking (Host Image)

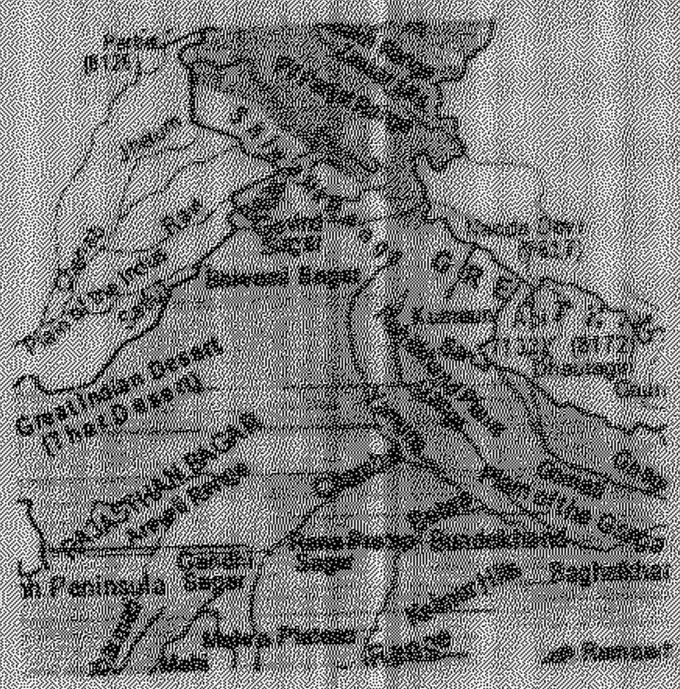


Image after watermarking (Watermarked Image)

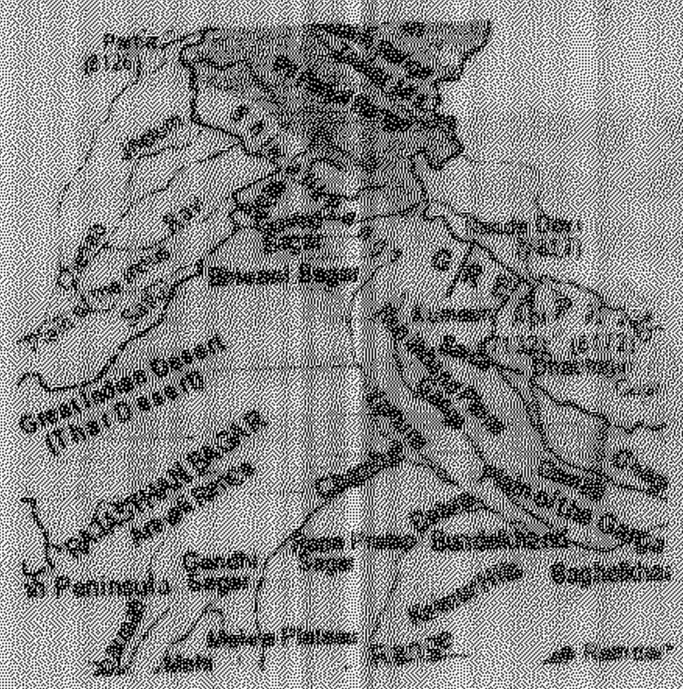


Image after attack(Attacked Image)

Bhattacharya distance, d= 0.0213492.

Bhattacharya distance ratio R= 3.86796e x 10^{-8}.

SIM FACTOR = 0.600652

D. Conclusion attack :

From the knowledge of our experimental the results ,it shows that we need to take care of Bhattacharya distance ratio which is an important measure for for creating our target distortionless specific compact neighbourhood N(I) of the given watermarked image I. Here compact neighbourhood which contain reduced number of redundant images.Keeping the said distance ratio in our mind , if we create N(I) from I we can find our target image , which is available in N(I) with very high probability.

E. Conclusion:

State-of-the-art: possible to hide/remove watermark while preserving image quality.

Final remarks:

- *very useful to study watermark attacks;*
- *attackers don't know about original image(host image) and watermark statistics;*

F. Criticism:

(i). In our methodology we are trying to find out our target image keeping only the visual quality of the given image (which may be errorious due to perception of Human Visual System) without maintaining PSNR(Pick Signal To Noise Ratio) properly.

(ii).The said modified N(I) which is created finally, cannot give guaranty that all images from N(I) do not contain any watermark.