

# Generalized watermarking attack based on watermark estimation and perceptual remodulation

S.Voloshynovskiy<sup>ab</sup>, S. Pereira<sup>a</sup>,  
A. Herrigel<sup>b</sup>, N. Baumgartner<sup>b</sup>, T. Pun<sup>a</sup>

<sup>a</sup>University of Geneva, Switzerland

<sup>b</sup>Digital Copyright Technologies, Zurich, Switzerland



# Outline

1. Introduction
2. Watermarking Paradigm
3. Weak Points of Watermarking Algorithms
4. Watermark Attacks
5. Results
6. Conclusions

# 1. Introduction

## Problem:

Market requires copyright protection technology

for images, MPEG, and audio since copying is easy, quick and with the most recent technology lossless (e.g. CDROM writing)

Possible Solution: insertion of a watermark (WM) which can be used to authenticate an image, audio soundtrack, MPEG etc.

Watermark contains for example: unique identifier specifying buyer and seller, most useful are WM which have at least 60 bits of information.

# 1. Introduction

- Robustness: WM must resist main:
  - *geometrical attacks* (desynchronization), e.g.: cropping and translation, affine transformations (like rotation, scaling, aspect ratio changes), row/column removal, random local distortions;
  - *signal processing attacks*: non-linear and adaptive filtering, compression, (re)quantization, multiple watermarks and noise addition.
- WM should be invisible and detectable without original image (oblivious)!

# 1. Introduction

## Problems:

- no standard, general purpose benchmark
- lack of robustness to attacks

## Why work on attacks:

- develop better watermarking methods
- define better benchmarking

Pioneering work: Stirmark (benchmarking), Unzign

# 1. Introduction

## Goals of watermarking attacks:

Notations:

$x$  : original (cover image),

$w$  : noise-like watermark,

$y$  : stego image, with

$$y = x + w$$

## Main goal of watermarking attacks:

- preserve image quality

$y' \cong x$  : **attacked** stego-image.

- render watermark undetectable/undecodable

# 1. Introduction

## Goals of watermarking attacks:

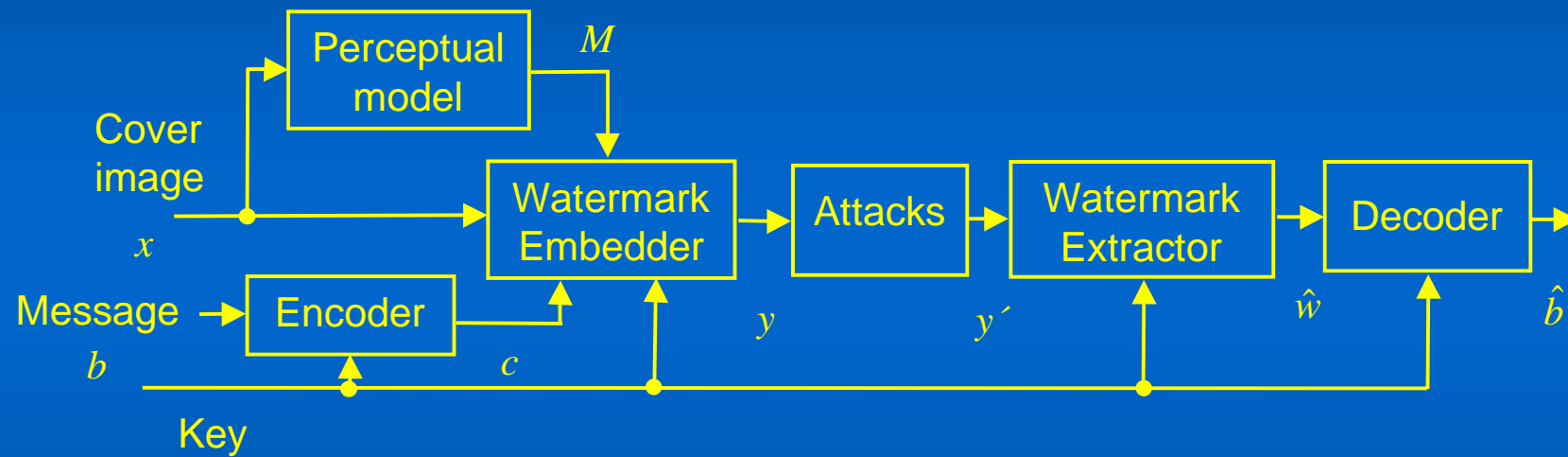
Our goal is to use prior knowledge:

- of watermark and cover-image statistics;
- of watermarking method used

to develop generalized attack on watermarking schemes:

- that takes into account *human perception*;
- *is stochastic*: applicable to a wide class of image and video watermarking algorithms operating in coordinate or transform (FT, DCT, wavelet) domains.

## 2. Watermarking Paradigm



## 2. Watermarking Paradigm

### Message embedding:

- Message  $b$  is encoded using either  $M$ -ary modulation or Error Correction Codes (ECC);
- The encoded message  $c$  is spreaded over the cover-image based on an orthogonal projection function  $p$ ;
- The spreaded message is masked by a perceptual mask  $M$  resulting in noise-like watermark  $w$  and added to the cover-image.

## 2. Watermarking Paradigm

### Message extraction:

- *Watermark extraction* is based either on Maximum Likelihood (ML) or Maximum a Posteriori (MAP) estimators:

$$\text{ML: } \hat{w} = \arg \max_{\tilde{w} \in \mathfrak{R}^N} p_X(y' | \tilde{w})$$

$$\text{MAP: } \hat{w} = \arg \max_{\tilde{w} \in \mathfrak{R}^N} \{p_X(y' | \tilde{w}) \cdot p_W(\tilde{w})\} \Rightarrow \hat{w} = \frac{R_w}{R_w + \hat{R}_x} (y' - \bar{y}')$$

$p_X(\cdot)$  : p.d.f. of the cover-image

$p_W(\cdot)$  : p.d.f. of the watermark

## 2. Watermarking Paradigm

### Message extraction:

- *Message decoding* is based mostly on ML decoder:  
Despreading part is designed based on ML detection under assumption of known signal in additive Gaussian noise:

$$r = \langle \hat{w}, p \rangle$$

the central limit theorem (CLT) approximation for large sample spaces:

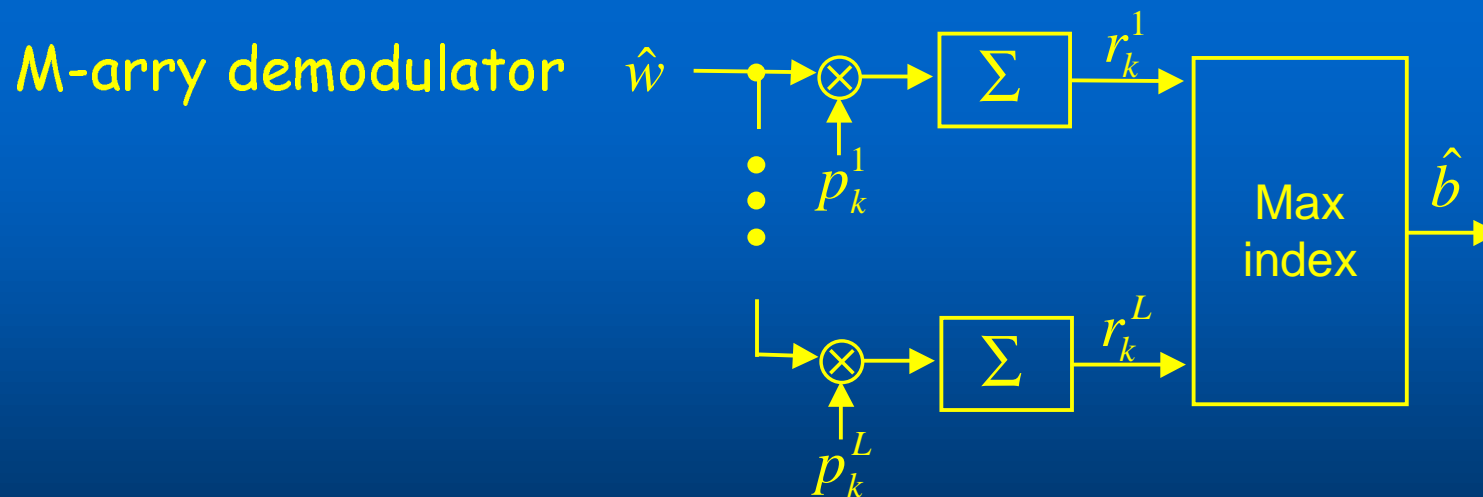
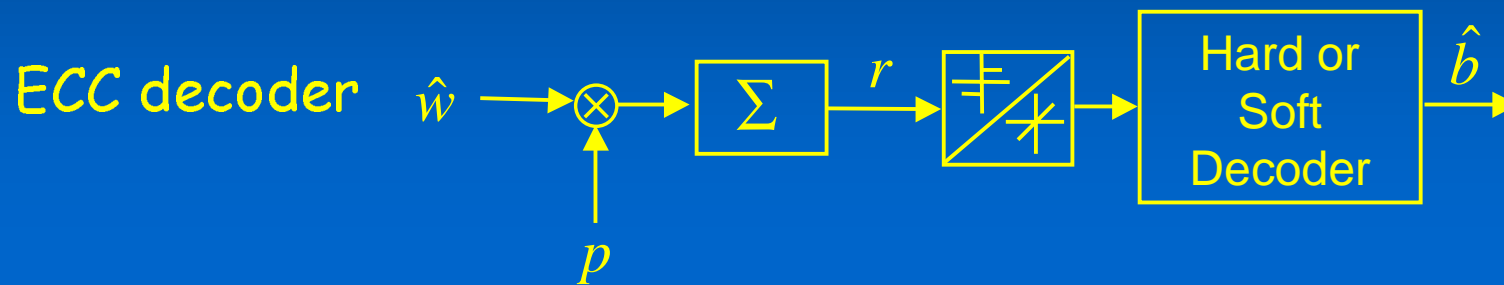
$$r = Ac + n$$

ML message decoder for additive white Gaussian noise:

$$\hat{b} = \arg \max_{\tilde{b}} p(r | \tilde{b}, x)$$

## 2. Watermarking Paradigm

Message extraction:



### 3. Weak Points of Watermarking Algorithms

- watermark prediction is key-independent

$$\hat{w} = \frac{R_w}{R_w + \hat{R}_x} (y' - \bar{y}') \begin{array}{l} \rightarrow \text{Destroy WM} \\ \rightarrow \text{Copy WM} \end{array}$$

- despreading part is designed as ML detector for AWGN channel

$$r = \langle \hat{w}, p \rangle$$

- decoder is designed in assumption of AWGN:  $n \sim N(0, \sigma_n^2 I)$

$$r = Ac + n$$

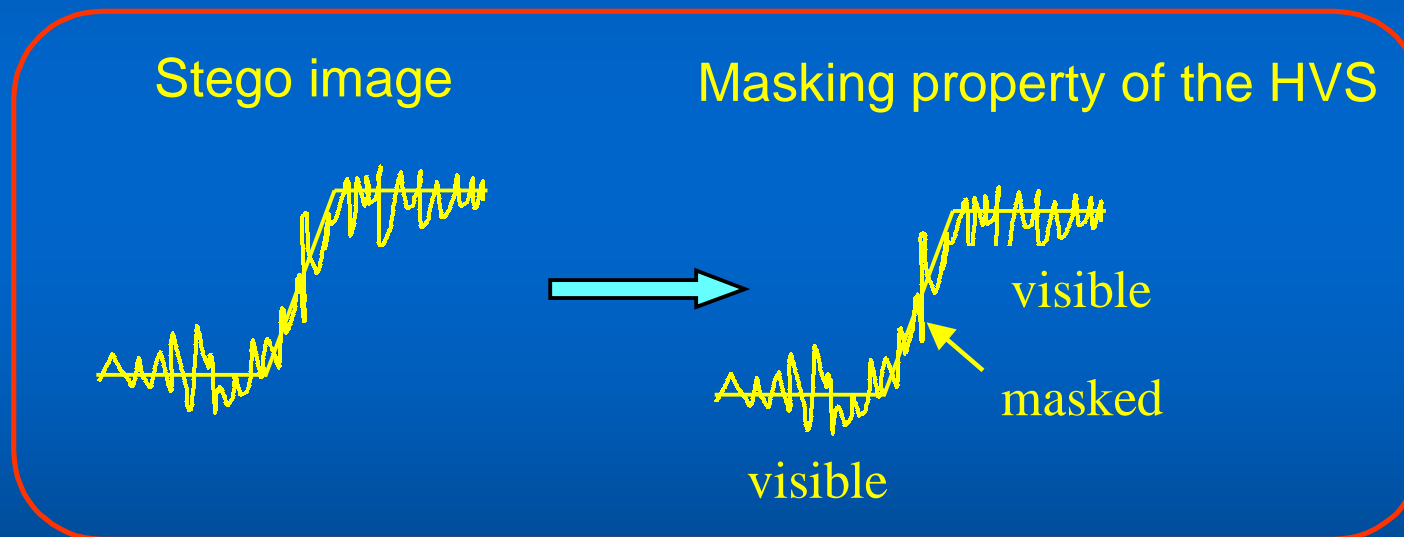
## 4. Watermark Attacks

The main attacks based on the weaknesses of watermarking algorithms:

- decrease the watermark redundancy using watermark removal based on denoising/compression;
- create the least favorable statistics for the AWGN decoder using perceptual remodulation of the watermark.

## 4. Watermark Attacks

Masking property of Human Visual System (HVS):



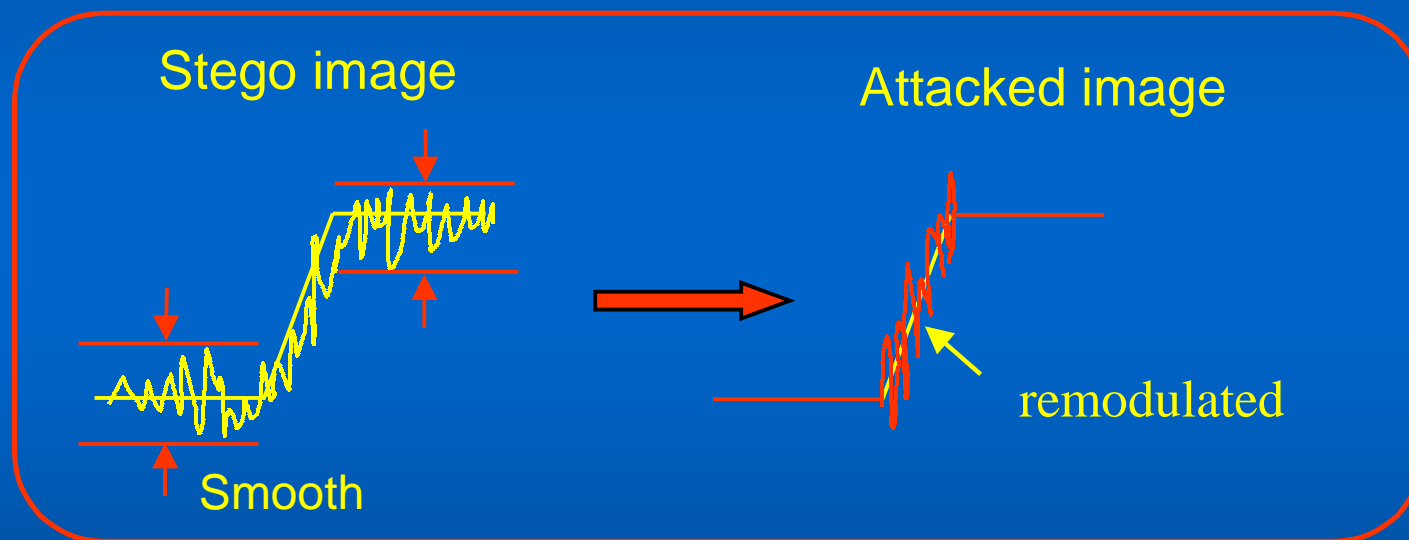
## 4. Watermark Attacks

Two stage attack:

- Stage 1: watermark estimation and removal based on denoising/compression;
- Stage 2: watermark remodulation using watermark statistics and properties of the HVS.

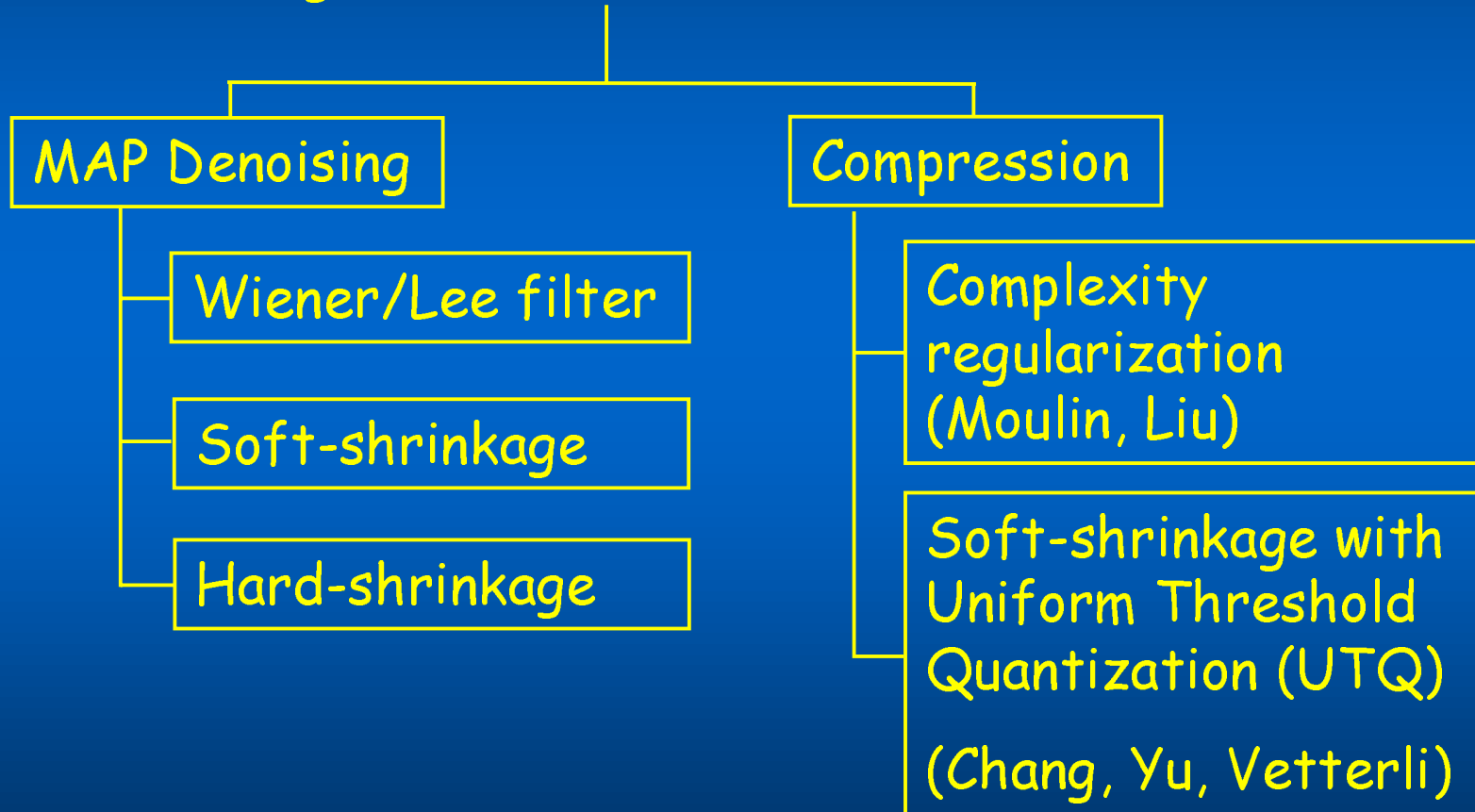
## 4. Watermark Attacks

Basic idea:



## 4. Watermark Attacks

Stage 1: watermark removal:  $y' \cong \hat{x}$  and  $w \rightarrow 0$



## 4. Watermark Attacks

MAP Denoising:

$$\hat{x} = \arg \max_{\tilde{x} \in \mathfrak{R}^N} \{ \ln p_w(\mathbf{y} | \tilde{x}) + \ln p_x(\tilde{x}) \}$$

Assumptions:  $w \sim i.i.d.N(0, \sigma_w^2 I)$   
 $x \sim i.i.d.GG(\bar{x}, \gamma, \sigma_x^2 I)$ : Generalized Gaussian

$$\hat{x} = \arg \min_{\tilde{x} \in \mathfrak{R}^N} \left\{ \frac{1}{2\sigma_w^2} \|\mathbf{y} - \tilde{x}\|^2 + \rho(res) \right\}$$

$\rho(res)$ : the energy function for the GG model

## 4. Watermark Attacks

MAP Denoising: (i.i.d. Gaussian watermark)  $w \sim N(0, \sigma_w^2 I)$

- Wiener/Lee filter (locally i.i.d. Gaussian image)

$$\hat{x} = \bar{y} + \frac{\sigma_x^2}{\sigma_w^2 + \sigma_x^2} (y - \bar{y})$$

- Soft-shrinkage (globally i.i.d. stationary Generalized Gaussian (sGG) image  $x \sim sGG(\bar{x}, 1, \sigma_x^2 I)$  (Laplacian  $\gamma = 1$ ))

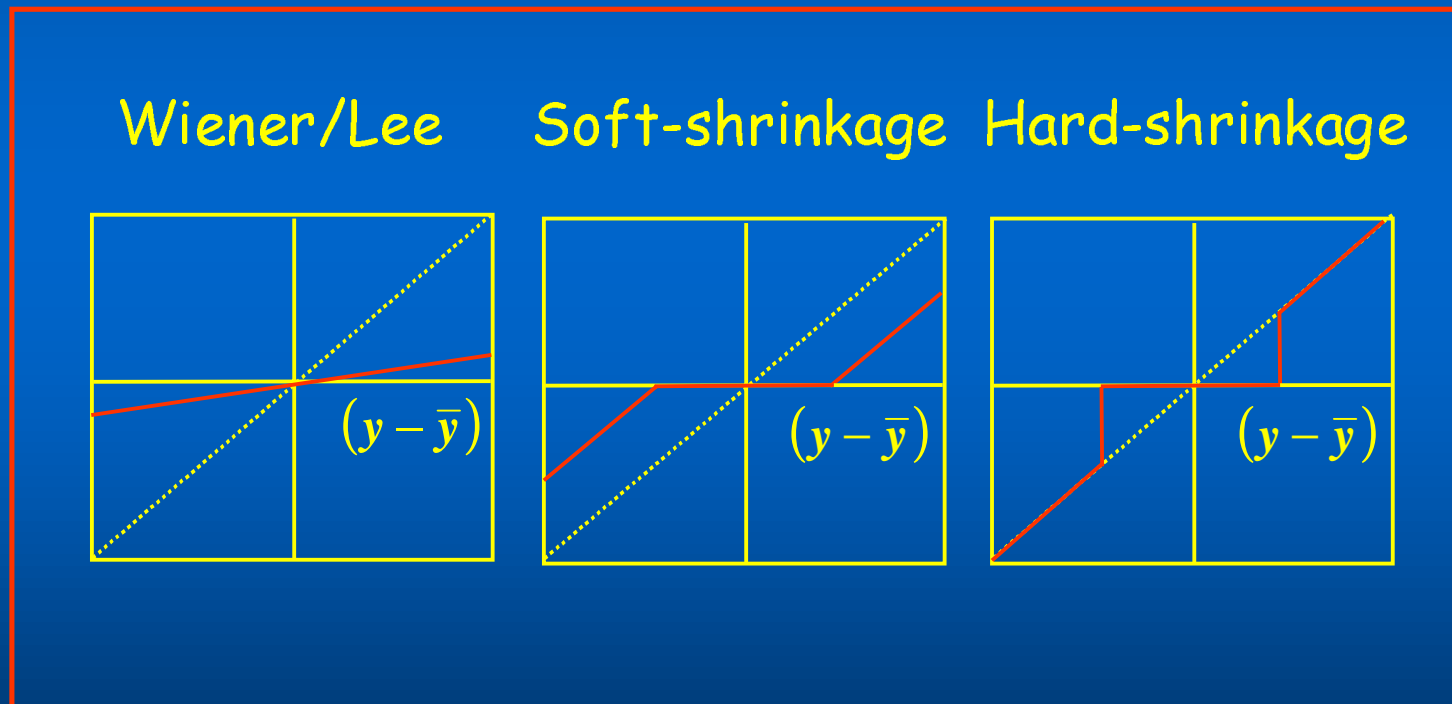
$$\hat{x} = \bar{y} + \max(0, |y - \bar{y}| - T) \text{sign}(y - \bar{y})$$

- Hard-shrinkage (globally i.i.d. sGG image:  $\gamma \rightarrow 0$ )

$$\hat{x} = \bar{y} + (y - \bar{y}) I(|y - \bar{y}| > T)$$

## 4. Watermark Attacks

Scaling/shrinking denoising functions:

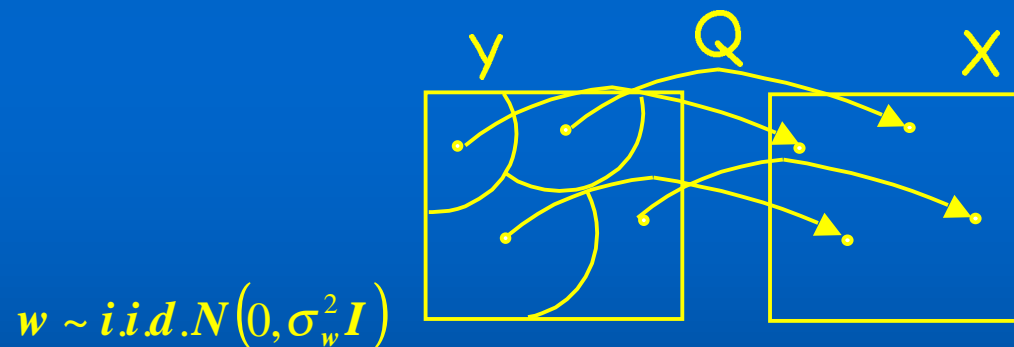


## 4. Watermark Attacks

Watermark removal based on lossy compression

- **Complexity regularization:** (MAP:  $\tilde{x} \in \mathfrak{X}^N \rightarrow \tilde{x} \in \Gamma$ )

$$\hat{x} = \arg \max_{\tilde{x} \in \Gamma} \{ \ln p_w(y | \tilde{x}) + \ln p^\Gamma(\tilde{x}) \} \quad \Gamma = \{ \tilde{x}_j, 1 \leq j \leq J \}$$

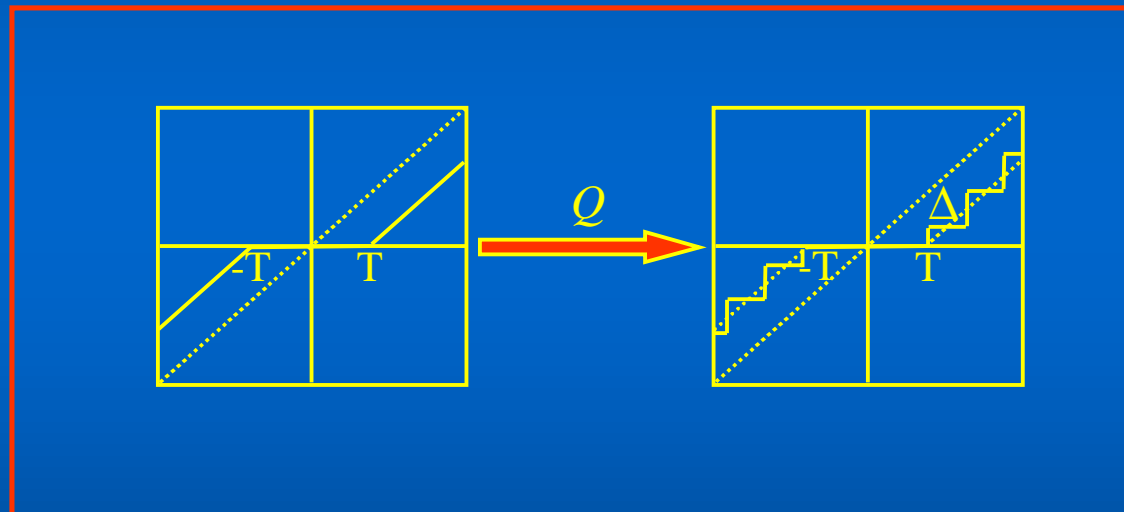


$$\hat{x} = \arg \min_{\tilde{x} \in \Gamma} \left\{ \frac{1}{2(\ln 2)\sigma_w^2} \|y - \tilde{x}\|^2 + \ell(\tilde{x}) \right\} = \arg \min_{\tilde{x} \in \Gamma} \left\{ \|y - \tilde{x}\|^2 + 2(\ln 2)\sigma_w^2 \ell(\tilde{x}) \right\}$$

## 4. Watermark Attacks

Watermark removal based on lossy compression

- Uniformly quantized soft-shrinkage



## 4. Watermark Attacks

Stage 2: watermark perceptual remodulation:

Particularities:

- stage 1: kills (removes) watermark (mostly in flat regions);
- stage2: changes the sign of watermark on opposite and creates outliers (mostly in the arias of edges and textures).

## 4. Watermark Attacks

Stage 2: watermark perceptual remodulation:

- estimation of watermark:  $\hat{w} = y - \hat{x}$
- estimation of watermark sign:  $s = \text{sign}(\hat{w})$
- perceptual remodulation:  $y' = \hat{x} + \{(1 - NVF) \cdot S_e + NVF \cdot S_f\} \cdot (-s) \cdot p'$

*NVF*: Noise Visibility Function;

$S_e$  : strength factor for edge regions ( $NVF \rightarrow 0$ );

$S_f$  : strength factor for flat regions ( $NVF \rightarrow 1$ ).

## 5. Results

Different technologies have been investigated:

- DCT
- Digimark
- Eikonamark
- SysCop



Test cases:

15 images of size 256x256

### Software:

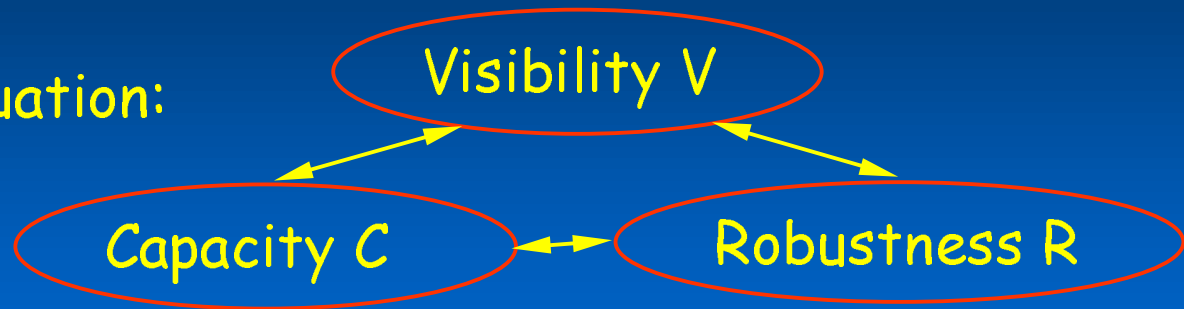
A: Coordinate domain, ECC, texture masking, WM 64 bits;

B: Coordinate domain, M-ary, luminance masking, WM 64 bits;

C: DCT domain, ECC, Just Noticeable Difference, WM 48 bits.

## 5. Results

Criteria for evaluation:



**Capacity C:** bits, typically 64...100.

**Robustness R:**

- bit error rate;
- binary decision: WM detected/not detected.

**Visibility V:**

- subjective human evaluation;
- HVS-based computer model;
- weighted PSNR.

## 5. Results

Weighted PSNR:

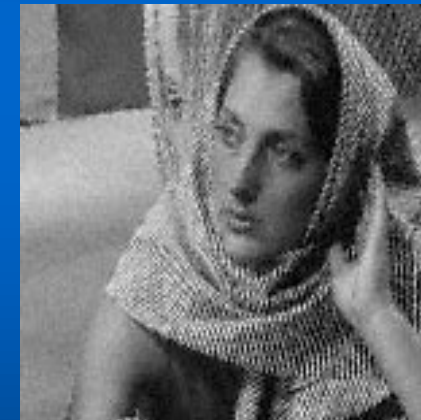
$$wPSNR = 10 \log_{10} \frac{\|255\|^2}{\|x - y\|_{NVF}^2}$$

wPSNR: 26.4dB

27.9dB

29.3dB

Stego image  
PSNR: 24.6dB



The wPSNR is closer to perception than the PSNR!

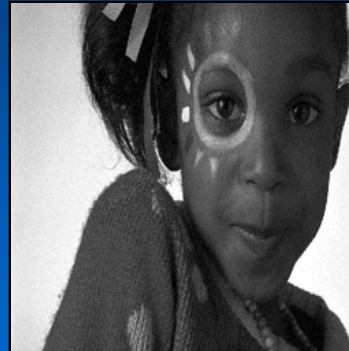
## 5. Results

### Software A

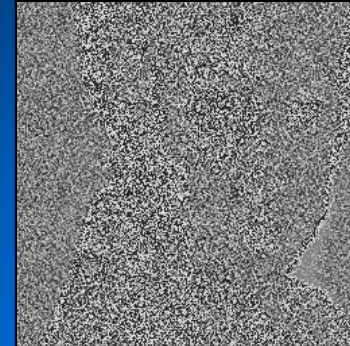
Stego image

PSNR: 35.3 dB

wPSNR: 36.9 dB



WM:  $y-x$



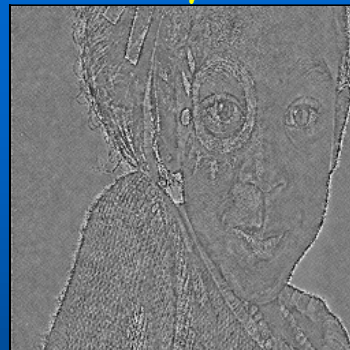
Attacked image

PSNR: 35.5 dB

wPSNR: 38.8 dB



WM:  $y'-x$



Message: watermark was not found

## 5. Results

### Software B

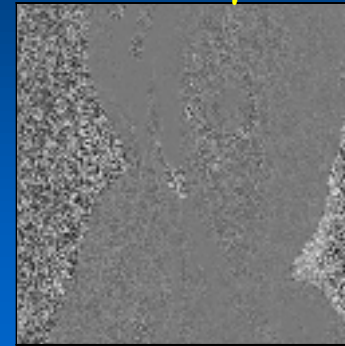
Stego image

PSNR: 36.8 dB

wPSNR: 37.4 dB



WM:  $y-x$



Attacked image

PSNR: 35.9 dB

wPSNR: 38.6 dB



WM:  $y'-x$



Message: watermark was not found

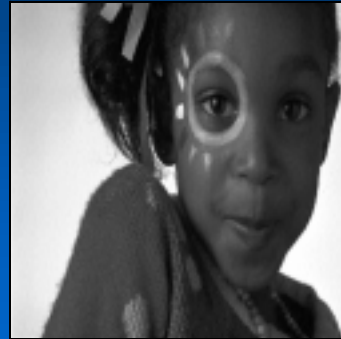
## 5. Results

### Software C

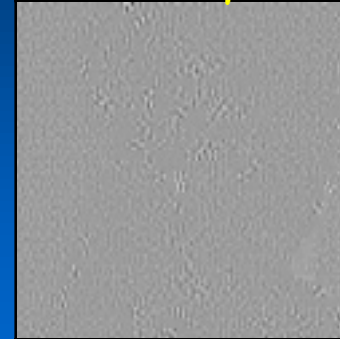
Stego image

PSNR: 40.8 dB

wPSNR: 42.4 dB



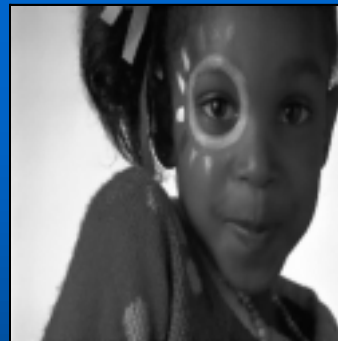
WM:  $y-x$



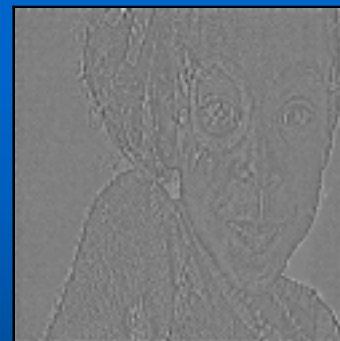
Attacked image

PSNR: 38.5 dB

wPSNR: 41.3 dB



WM:  $y'-x$



Message: watermark was not found

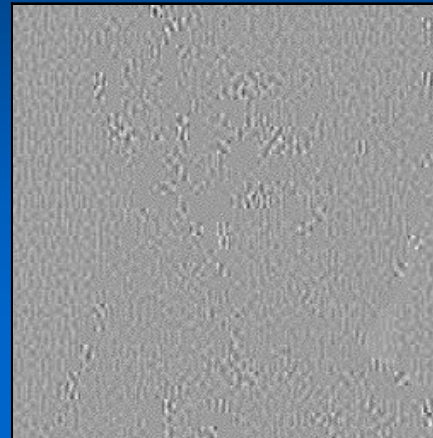
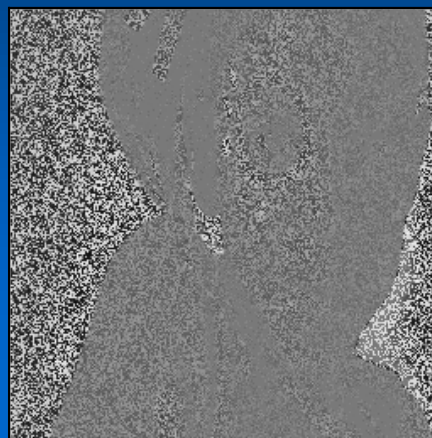
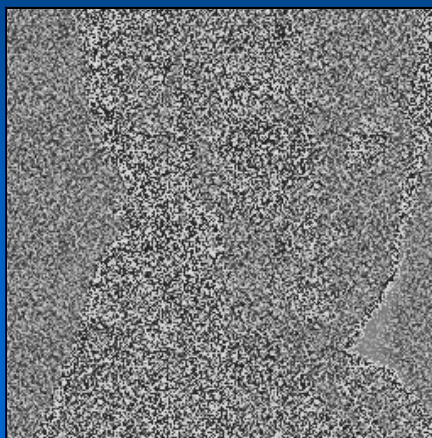
# 5. Results

A

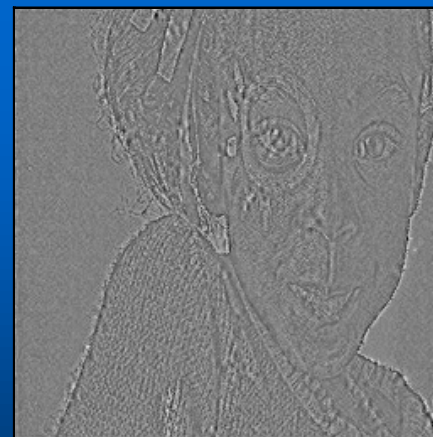
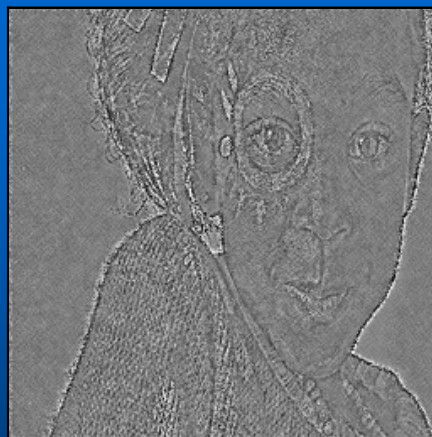
B

C

WM:  
 $y-x$



WM:  
 $y'-x$



## 6. Conclusions

- Possible to remove watermark while preserving image quality.
- Watermarking attacks should use as much as possible of image and watermark statistics.
- Watermarking algorithm development should assume that the attackers know your method (Kerhoff's principle).

Very useful to study watermark attacks for the development of the enhanced technologies assuming the bad guys are always one step ahead ...