

Watermark attacks

S. Voloshynovskiy, S. Pereira, T. Pun

Computer Science Department,
Centre Universitaire Informatique (CUI)
University of Geneva
Switzerland

Contact:

<http://cuiwww.unige.ch/~vision>



Content

1. Introduction

- 1.1 Why deal with attacks
- 1.2 Goals of watermarking attacks
- 1.3 Families of watermark attacks
- 1.4 Benchmarking watermarking methods
- 1.5 Benchmarking watermark attacks

2. Stochastic attacks

- 2.1 Introduction
- 2.2 Stage 1: watermark estimation
- 2.3 Stage 2: noise addition
- 2.4 Results of stochastic watermark removal

3. Synchronization attacks

- 3.1 Introduction
- 3.2 ACF analysis
- 3.3 Template removal

4. Conclusions

1. Introduction

1.1 Why deal with attacks

Market is lukewarm towards watermarking technology:

- non-disclosed methods;
- no standard, general purpose benchmark;
- 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.

1.2 Goals of watermarking attacks

Notations:

- x: original (cover image), size $N = M \cdot M$,
- n: noise-like watermark,
- y: stego-image, with

$$y = x + n \quad (2.1)$$

y': **attacked** stego-image.

Main goals of attacks on watermarks:

- preserve image quality:

$$y' \cong x \quad (2.2)$$

- render watermark undetectable/undecodable.

Our goal is to use prior knowledge:

- of watermark and image probability distributions;
- of the watermarking method used.

1.3 Families of watermark attacks

Main attack families we are concerned with:

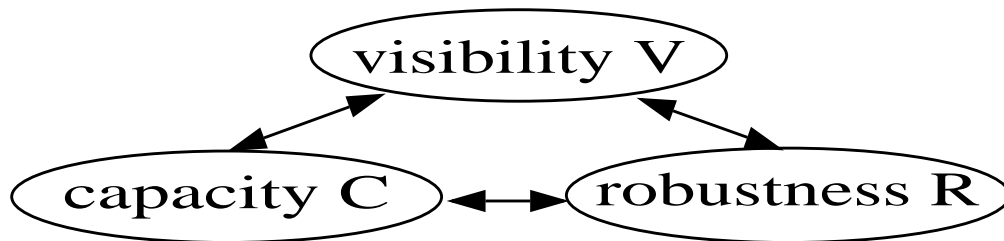
- geometric → desynchronization, e.g.:
 - affine transforms;
 - cropping, row/column removal;
 - random local distortions;
 - mosaicing;
- signal processing → desynchronization, watermark drowning, e.g.:
 - lossy compression, (re)quantization, dithering;
 - linear, non-linear and adaptive filtering, denoising;
 - multiple watermarks, noise addition;
 - collage, superimposition;
 - **stochastic attacks**;
- specialized, based on knowledge of method:
 - **desynchronization attacks**;
 - chrominance attack;
 - etc.

We ignore here cryptographic attacks, system-based attacks (e.g. Oracle, counterfeit original, averaging).

Stirmak: geometric, signal processing.

1.4 Benchmarking watermarking methods

3 related criteria for watermarking, reflected in the benchmarks:



Visibility V:

- subjective human evaluation;
- HVS-based computer model;
- PSNR:

$$\text{PSNR} = 10 \log \frac{\text{max_luminance_x}^2}{\|(y - x)\|^2} \quad (2.3)$$

Capacity C: bits, typically 64 .. 100.

Robustness R:

- bit error rate;
- binary decision:
 - watermark detected;
 - watermark not detected.

Stirmark: subjective evaluation, binary answer only.

1.5 Benchmarking watermark attacks

Visibility V:

- subjective human evaluation;
- HVS-based computer model;
- **weighted PSNR** measured on $y' - x$:

wPSNR =

$$10\log \frac{\max_lum_x^2}{\|(y' - x)\|_{NMF}^2} = 10\log \frac{\max_lum_x^2}{\|(y' - x) \cdot NMF\|^2} \quad (2.4)$$

(e.g. flat region: $NMF = 1 \rightarrow$ max penalization)

Capacity C: given number of bits.

Robustness R:

- bit error rate;
- binary answer:
 - watermark detected;
 - watermark not detected.
- ternary answer:
 - watermark present & detected,
 - watermark present & not detected,
 - watermark not present.

The wPSNR is closer to perception than the PSNR:

stego-image
PSNR 24.6dB
wPSNR 26.4dB



stego-image
PSNR 24.6dB
wPSNR 27.9dB



stego-image
PSNR 24.6dB
wPSNR 29.3dB



2. Stochastic attacks

2.1 Introduction

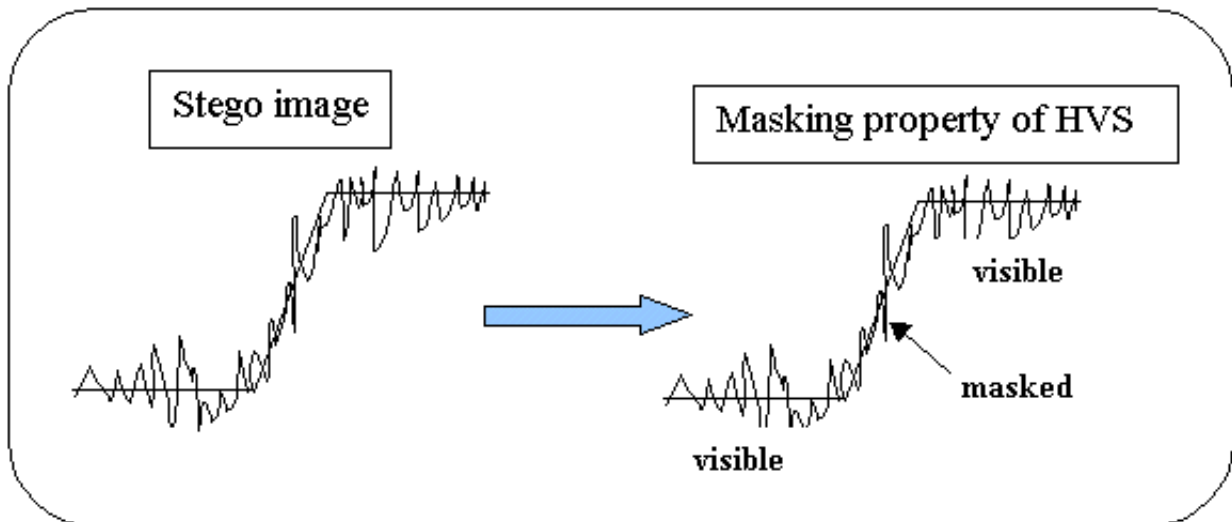
Goal: general attack on watermark schemes.

The attack:

- takes into account **human perception**;
- is **stochastic**: applicable to a wide class of image and video watermarking schemes.

Can be used against embedding schemes operating in coordinate or transform (FT, DCT, wavelets) domains.

Masking property:

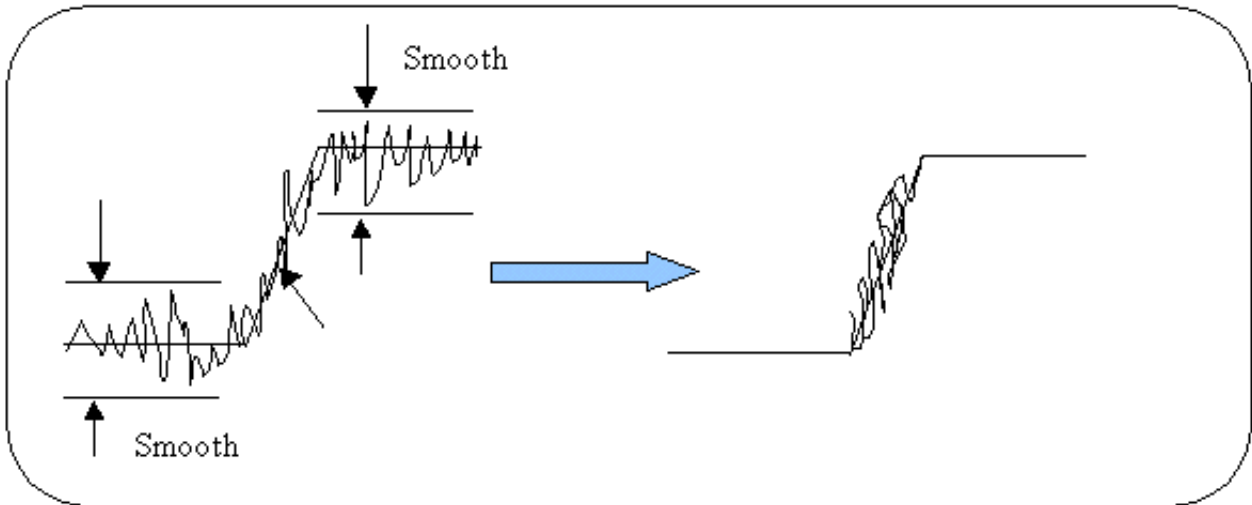


(Details in Information Hiding 1999 paper.)

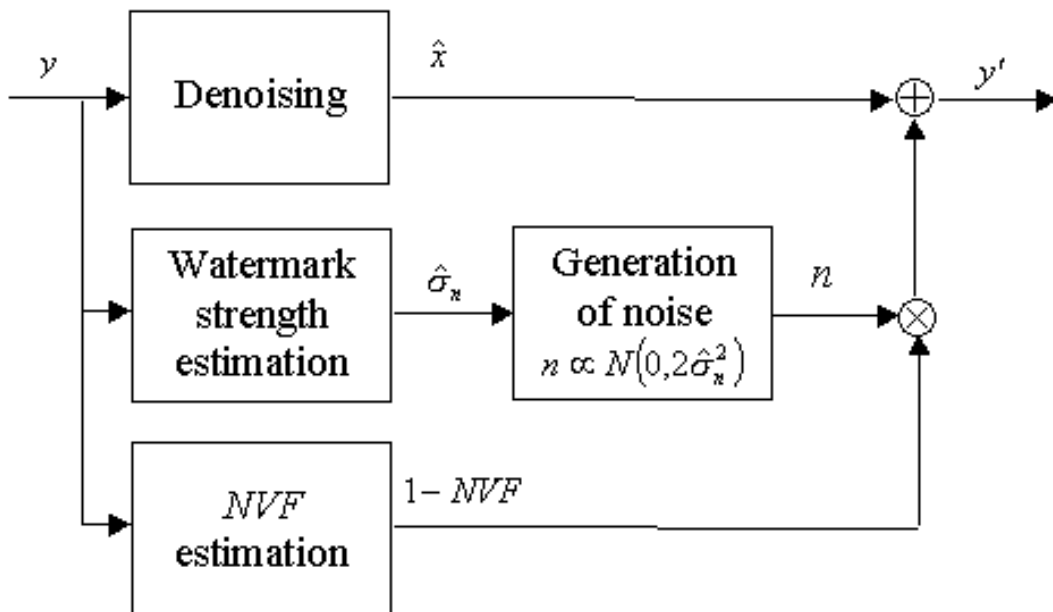
Two stages attack:

- watermark estimation and removal: denoise;
- watermark hiding: add noise, using watermark statistics and HVS properties.

Basic idea:



Implementation:



2.2 Stage 1: watermark estimation

Goal: remove watermark from flat regions.

Watermark:

$$\hat{n} = y - \hat{x}, \quad (2.5)$$

where \hat{n} , \hat{x} are estimates of watermark & cover image.

Assumptions:

- watermark = Gaussian r.v., indep. ident. distributed samples (spread spectrum wm, binary wm + NVF):

$$p_n(n) \propto \text{i.i.d. } \mathcal{N}(0, \sigma_n^2) \quad (2.6)$$

- cover image: stationary Generalized Gaussian distribution, i.i.d. samples:

$$p_x(x) \propto \text{i.i.d. } GG(\bar{x}, R_x) \quad (2.7)$$

for which the shape parameter γ can vary:

- $\gamma = 2$: Gaussian distribution,
- $\gamma = 1$: Laplacian distribution,
- $0.3 \leq \gamma \leq 1$: real cover images.

Other possibility: non-stationary Gaussian pdf for cover image (see Information Hiding 1999 paper).

Estimation of \hat{x} :

$$\hat{x} = \operatorname{argmax}\{\ln p_n(y|\tilde{x}) + \ln p_x(\tilde{x})\}, \tilde{x} \in \mathfrak{R}^N \quad (2.8)$$

Iterative RLS - Reweighted Least Squares solution:

$$\hat{x}^k \rightarrow \hat{w}^{k+1} \rightarrow \hat{x}^{k+1} \quad (w: \text{weight}) \quad (2.9)$$

Resulting formulation, similar to the Lee filter:

$$\hat{x}^{k+1} = \hat{x}^k + \frac{\hat{\sigma}_x^2}{\hat{w}^k \hat{\sigma}_n^2 + \hat{\sigma}_x^2} (y - \hat{x}^k) \quad (2.10)$$

Equivalent form as generalized Wiener filter:

$$\hat{x}^{k+1} = \frac{\hat{w}^k \hat{\sigma}_n^2}{\hat{w}^k \hat{\sigma}_n^2 + \hat{\sigma}_x^2} \hat{x}^k + \frac{\hat{\sigma}_x^2}{\hat{w}^k \hat{\sigma}_n^2 + \hat{\sigma}_x^2} y \quad (2.11)$$

where for one iteration step k :

- $\hat{\sigma}_n^2$: wm variance estimate, eg. on flat regions;
- $\hat{\sigma}_x^2 \rightarrow \hat{\sigma}_{x_i,j}^2, 1 \leq i, j \leq N$: local img variance estimate;
- $\hat{w}^k(i, j) = \frac{\gamma[\eta(\gamma)]^\gamma}{|r^k(i, j)|^{2-\gamma}}, \hat{r}(i, j) = \frac{\hat{x}(i, j) - \hat{\hat{x}}(i, j)}{\hat{\sigma}_x}$;
- γ : estimated using moment matching;
- $\eta(\gamma) = \sqrt{\Gamma(3/\gamma)/\Gamma(1/\gamma)}$, with Gamma fonction.

2.3 Stage 2: noise addition

Goal: add noise to hide/cancel watermark.

Noise visibility function (assuming noise $N(0, 1)$):

$$\text{NVF}(i, j) = \frac{w(i, j)\sigma_n^2}{w(i, j)\sigma_n^2 + \sigma_x^2} \rightarrow \frac{w(i, j)}{w(i, j) + \sigma_x^2} \quad (2.12)$$

Behavior:

- flat regions: $\text{NVF} \rightarrow 1$;
- textured regions and edges: $\text{NVF} \rightarrow 0$

Watermark drowning:

$$y' = \hat{x} + \begin{matrix} [1 - \text{NVF}(i, j)] \cdot m(i, j) \cdot S_e + & \text{(edges)} \\ \text{NVF}(i, j) \cdot m(i, j) \cdot S_f & \text{(flat areas)} \end{matrix} \quad (2.13)$$

where:

- m : factor used to remodulate the watermark:

$$m(i, j) = -1 \cdot \text{sgn}[\hat{n}(i, j)] \quad (2.14)$$

- $\hat{n}(i, j)$: estimated from (2.11) and (2.5);
- S_e : strength factor for edge regions;
- S_f : strength factor for flat regions.

(If e.g. $S_f = 0$ and $S_e = 0$: pure denoising attack.)

2.4 Results of stochastic watermark removal

Software A, image 1:

original x



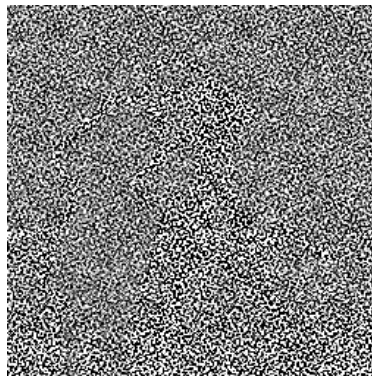
stego-image y
PSNR 34.7dB
wPSNR 35.7dB



y' ($S_e=2, S_f=1.5$)
PSNR 34.5dB
wPSNR 37.2dB



$y - x$



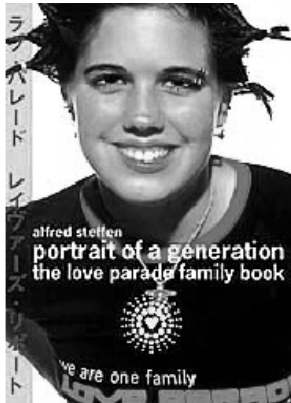
$y' - x$



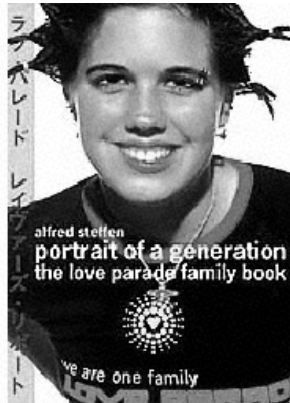
Message: *no watermark detected.*

Software A, image 2:

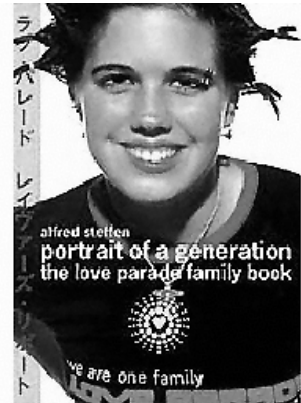
original x



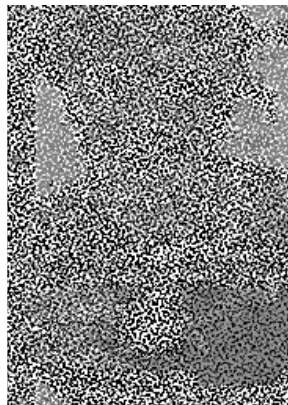
stego-image y
PSNR 35.8dB
wPSNR 37.4dB



y' ($S_e=2, S_f=1.5$)
PSNR 35.3dB
wPSNR 38.5dB



$y - x$



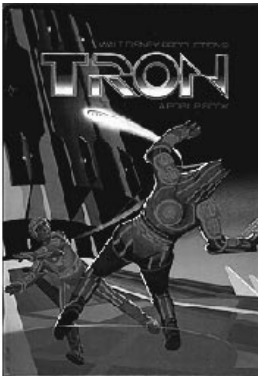
$y' - x$



Message: *no watermark detected.*

Software A, image 3 (synthetic image):

original x



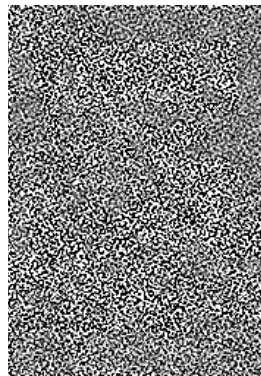
stego-image y
PSNR 35.4dB
wPSNR 36.6dB



y' ($S_e=2, S_f=1.5$)
PSNR 35.1dB
wPSNR 38.1dB



$y - x$



$y' - x$



Message: *no watermark detected.*

Software B, image 1:

original x



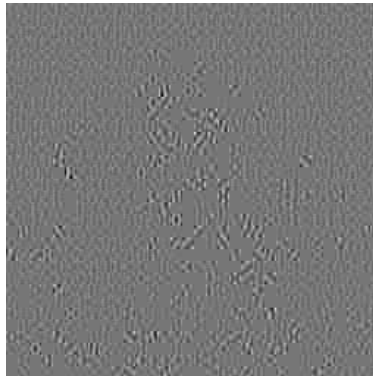
stego-image y
PSNR 41.5dB
wPSNR 42.5dB



y' ($S_e=2, S_f=1.5$)
PSNR 39.1dB
wPSNR 40.6dB



$y - x$



$y' - x$



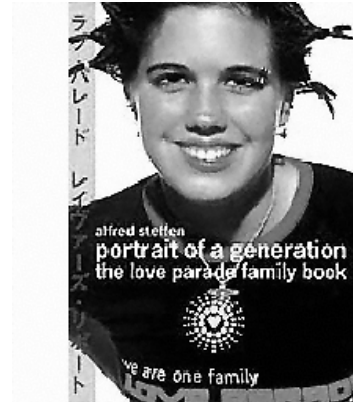
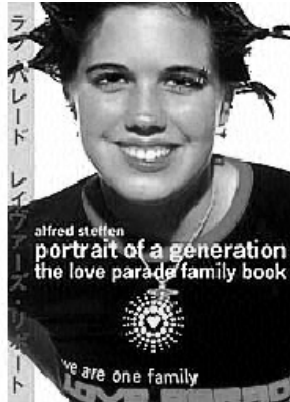
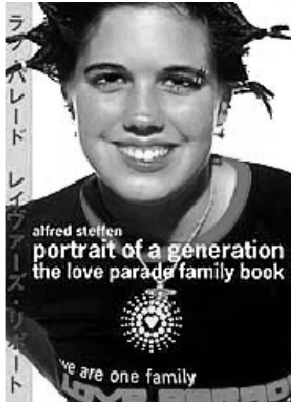
Message: *no watermark detected.*

Software B, image 2:

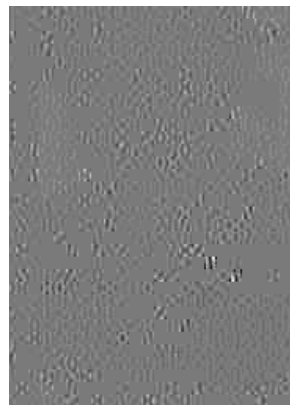
original x

stego-image y
 PSNR 41.5dB
 wPSNR 42.9dB

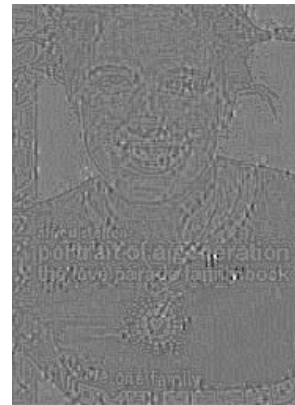
y' ($S_e=2, S_f=1.5$)
 PSNR 38.7dB
 wPSNR 41.3dB



$y - x$

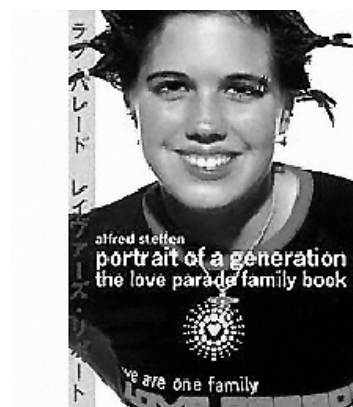


$y' - x$



Other parameters:

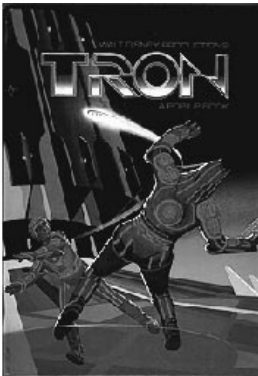
y' ($S_e=1, S_f=1.2$)
 PSNR 40.5dB
 wPSNR 42.6dB



Message: *no watermark detected.*

Software B, image 3 (synthetic image):

original x



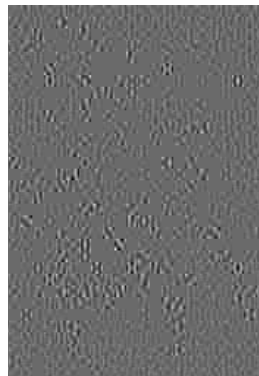
stego-image y
PSNR 41.2dB
wPSNR 43.1dB



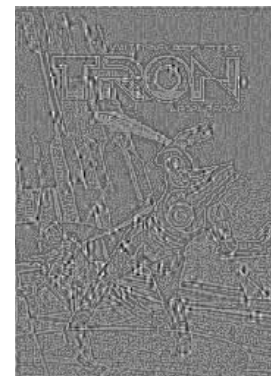
y' ($S_e=2, S_f=1.5$)
PSNR 38.9dB
wPSNR 41.4dB



$y - x$



$y' - x$



Message: *no watermark detected.*

3. Synchronization attacks

3.1 Introduction

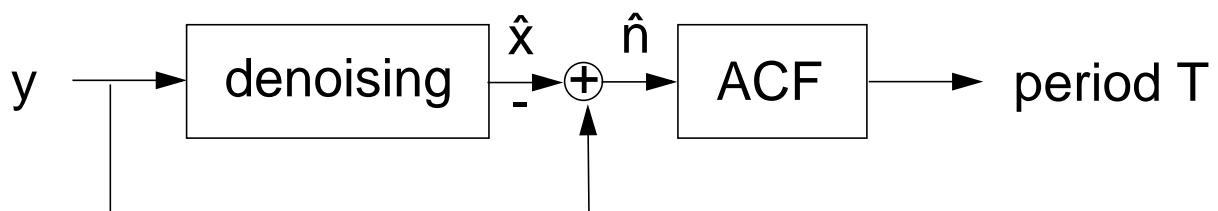
Goal: desynchronize spread-spectrum sequence.

Means of attack:

- (geometric distortions;)
- template search and removal:
 - known pattern (cross, sine wave);
 - peaks;
- ACF analysis.

3.2 ACF analysis

Use knowledge from ACF to determine period T:



Knowing T:

- better estimate of watermark \rightarrow easier removal;
- modify estimated watermark \hat{n} to cancel ACF.

3.3 Template removal

Goal: remove synchronizing template.

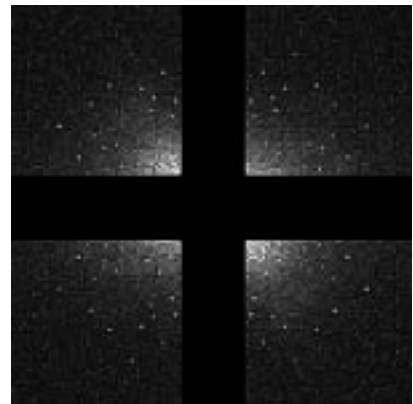
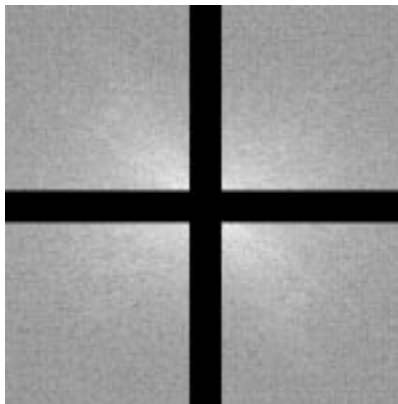
Principle: identify template peaks in FT domain.

Algorithm:

- cut the stego-image y into adjacent blocks;
- average the Fourier transforms of the blocks;
- estimate stable peaks as template peaks;
- Fourier transform the entire image;
- remove template peaks at the identified locations.

Example:

	FT(y)	FT(y)
stego-image y	no visible peaks	after blocking and averaging



4. Conclusions

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

Final remarks:

- very useful to study watermark attacks;
- watermarking methods should make use as much as possible of image and watermark statistics;
- assume attackers know your method
→ Kerckhoff's principle.

Final final remark: the bad guys are always one step ahead ...

Acknowledgements: CUI people (G. Csurka, F. Deguillaume, J. O'Ruanaidh), DCT people (A. Herrigel, N. Baumgärtner), EPFL-LTS people, and others ... Swiss Priority Program on Information and Communication Structures, ESPRIT OMI Project JEDI-FIRE.

