



Watermarking / Pirate identification Fingerprinting / Content identification

Frédéric LEFEBVRE, PhD

frederic.lefebvre@technicolor.com

technicolor

Security Competence Center, Rennes



Outline

- ❑ Introduction
- ❑ Watermarking
- ❑ Fingerprinting
- ❑ Applications
 - UGC Filtering
 - Pirate seat localization
- ❑ Conclusion & future work



Introduction

Quelques chiffres:

- ❑ Plus de 92% des films piratés sont disponibles avant leur sortie en DVD en France.
- ❑ Plus d'un tiers des films sortis en salle sont piratés sur internet.
- ❑ Plus d'un tiers des films piratés sont disponibles avant même leur sortie en salle.
- ❑ Les films sont disponibles en moyenne 45 jours après leur sortie en salle.



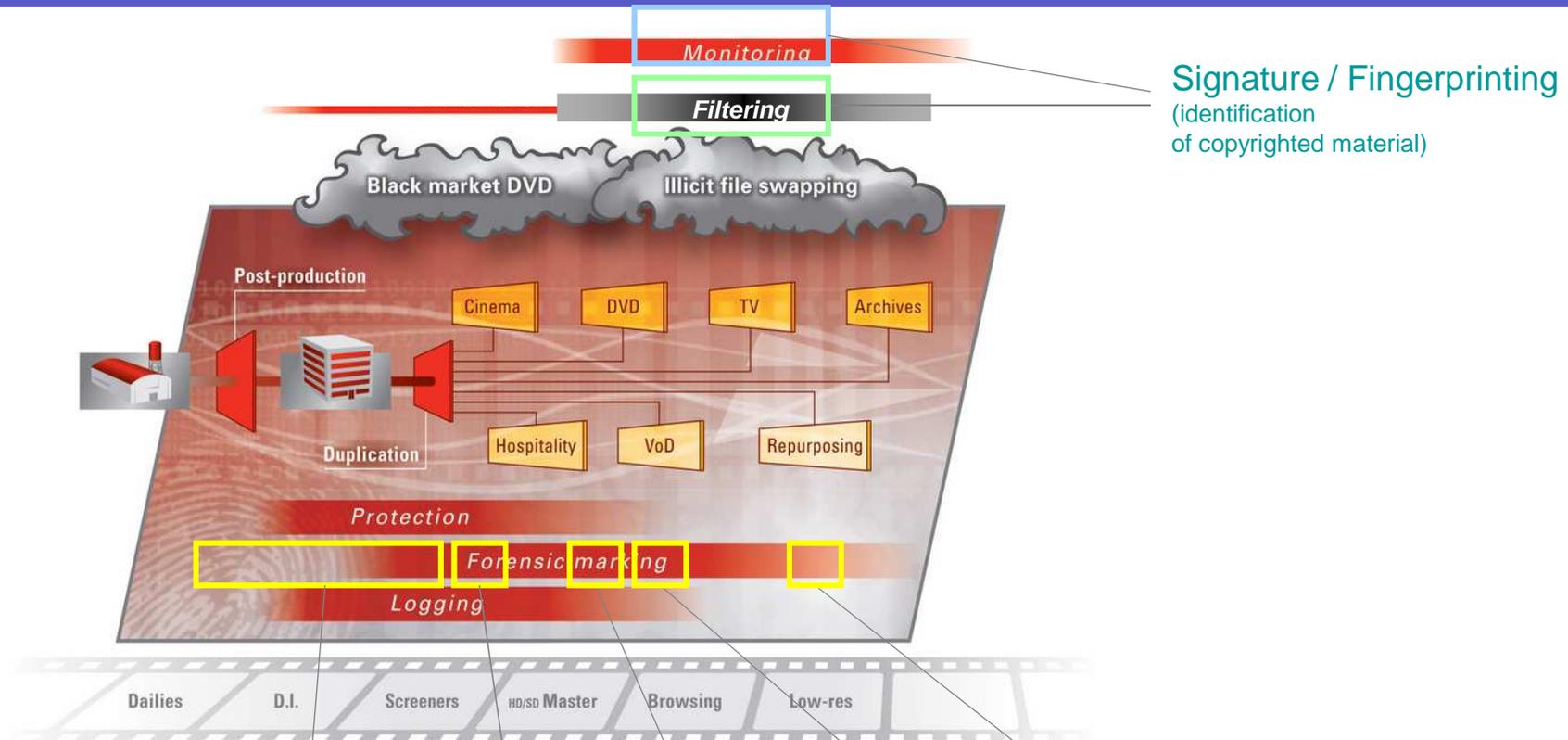
Guide des bonnes pratiques pour combattre la piraterie audiovisuelle (ALPA)

Les 10 Commandements:

- ❑ L'ensemble des professionnels doit être sensibilisé aux risques de piratage.
- ❑ Un responsable "sécurisation et traçabilité" doit être désigné au sein de chaque entreprise.
- ❑ Un interlocuteur "traçabilité" doit être désigné au sein de chaque entreprise
- ❑ Le nombre de copies doit être limité au minimum requis
- ❑ **Toute copie doit être marquée et toute copie numérique complète de l'œuvre doit être sécurisée**
- ❑ Toute copie doit être réalisée en fonction des besoins de son destinataire
- ❑ Toute copie doit être transportée dans un emballage sécurisé
- ❑ Tout mouvement de copie doit être organisé
- ❑ Toute copie complète de l'œuvre doit être conservée dans un lieu sécurisé
- ❑ **Toute copie promotionnelle doit être sécurisée et comporter une mise en garde spécifique.**



Content distribution



Signature / Fingerprinting
(identification of copyrighted material)

Forensic marking
Cinema
Master / Screener

Forensic marking
(DVD,...)

Forensic marking

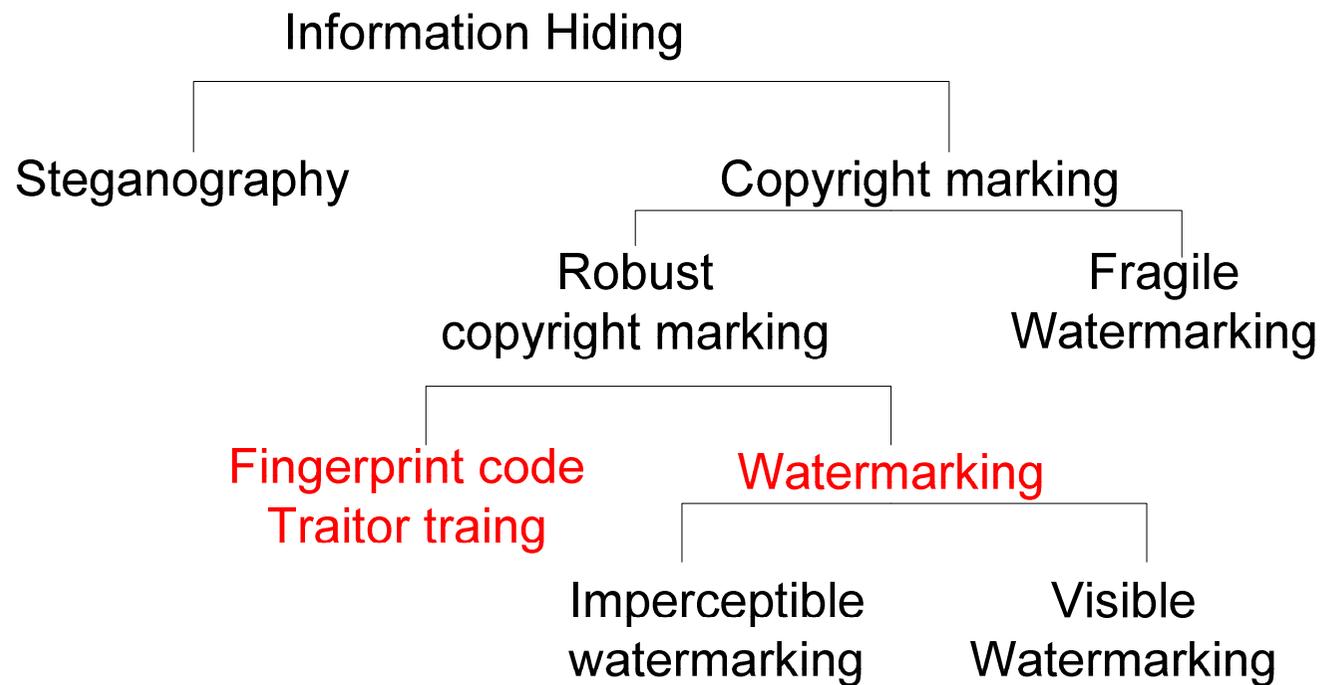


Outline

- Introduction
- Watermarking
 - **Introduction**
 - Study of an algorithm
 - Applications
- Fingerprinting
- Applications
- Conclusion & future work



Data Hiding



(*) B.Pfitzmann, « Information Hiding Terminology », pp.347-350, ISBN 3-540-61996-8



Data Hiding

❑ Steganography

- Maximize capacity (some KBYTES)
- The channel is totally hidden
- Very sensitive against attacks
- The opponent is passive

❑ Watermarking

- Maximize robustness against attacks
- Perceptually not detectable
- Small capacity (few bits)
- The opponent is active



Watermarking: applications

- ❑ Copyright protection.
- ❑ Copyright verification: monitoring.
- ❑ Multimedia streaming tracking.
- ❑ Copy attack protection, e.g. DVD copy.
- ❑ Document authentication.
- ❑ Labeling or indexing tool in a database.



Principles

The three main concepts are :

- Robustness
- Invisibility
- « Security »

The aim of the watermarking is to **embed** a **robust** and **not perceptive message** (#Gaussian noise) in a content.



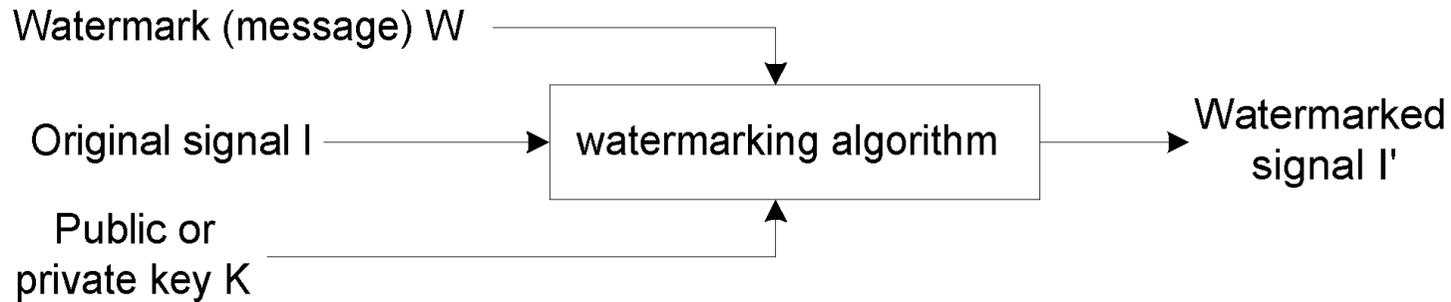
Principles

- ❑ The robustness is guaranteed by the redundancy and the strength of the watermark
- ❑ The invisible property is given by psycho visual laws.
- ❑ The security is guaranteed by
 - ❖ the algorithm confidentiality
 - ❖ Keys

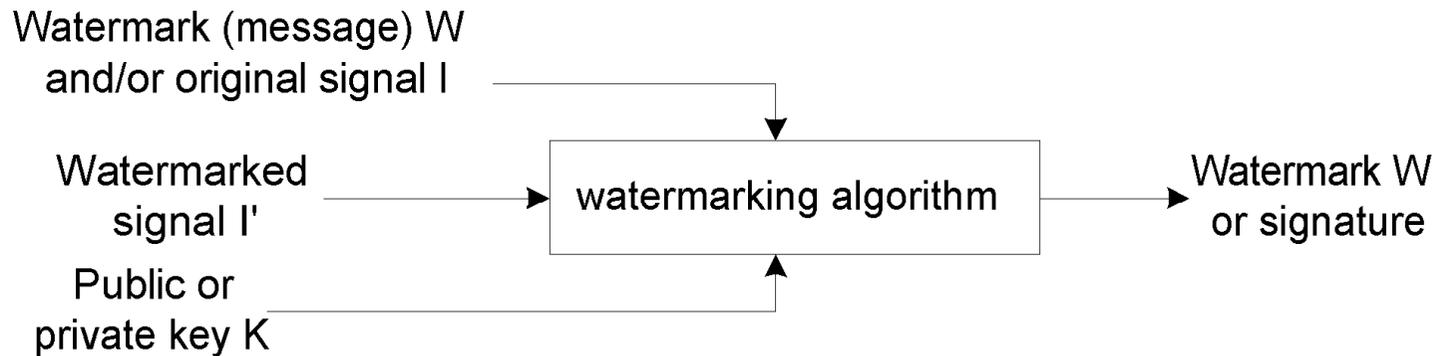
The tradeoff robustness/invisibility/security depends on the usage scenario



Insertion/Extraction



Insertion scheme



Extraction scheme



Watermarking systems

Four types of watermarking systems:

- Private watermarking (non blind watermarking).
 - ❖ $(I, I', K) \rightarrow W$.
 - ❖ $(I, I', K, W) \rightarrow \{0, 1\}$.

- Semiprivate watermarking (semi blind watermarking).
 - ❖ $(I', K, W) \rightarrow \{0, 1\}$.

- Informed watermarking
 - ❖ $(I', K, f(I)) \rightarrow W$

- Public watermarking (blind watermarking)
 - ❖ $(I', K) \rightarrow W$.



Watermarking systems

□ The system is

- ❖ Asymmetric,
if the keys K and the algorithm are different
in the insertion and in the detection
processes
- ❖ Symmetric,
if the keys K and the algorithm are the
same in the insertion and in the detection
processes
- ❖



Outline

- Introduction
- Watermarking
 - Introduction
 - **Study of an algorithm**
 - ❖ Spatial domain
 - ❖ Transform domain
 - Applications
- Fingerprinting
- Applications
- Conclusion & future work



Study of an algorithm

“Print and scan optimized watermarking scheme”,
IEEE Multimedia Signal Processing, 2000.

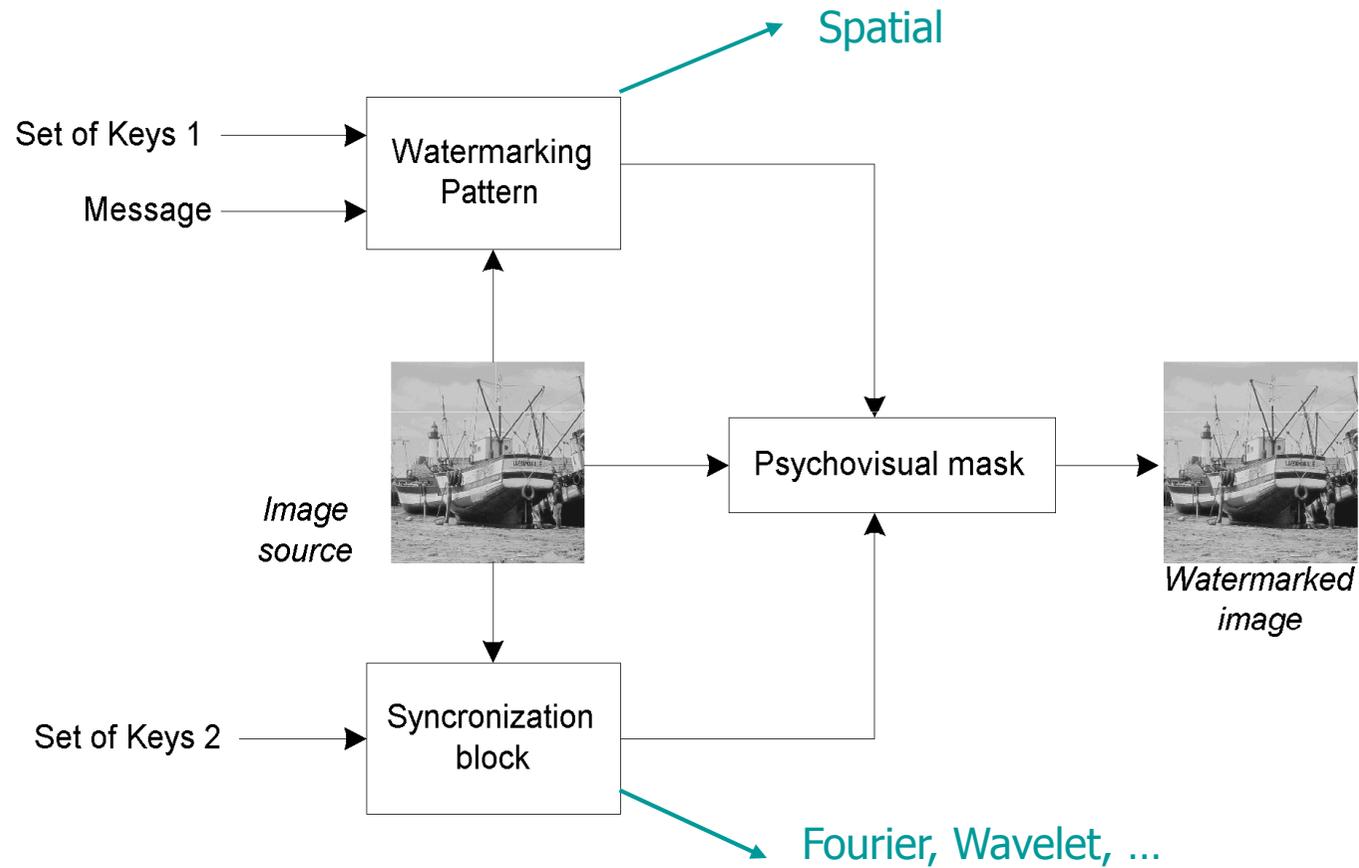
It combines 2 watermarking schemes:

- The message is embedded in spatial domain.
- The resistance against geometric attacks (rotation, scaling) is guaranteed by the insertion in Fourier domain.

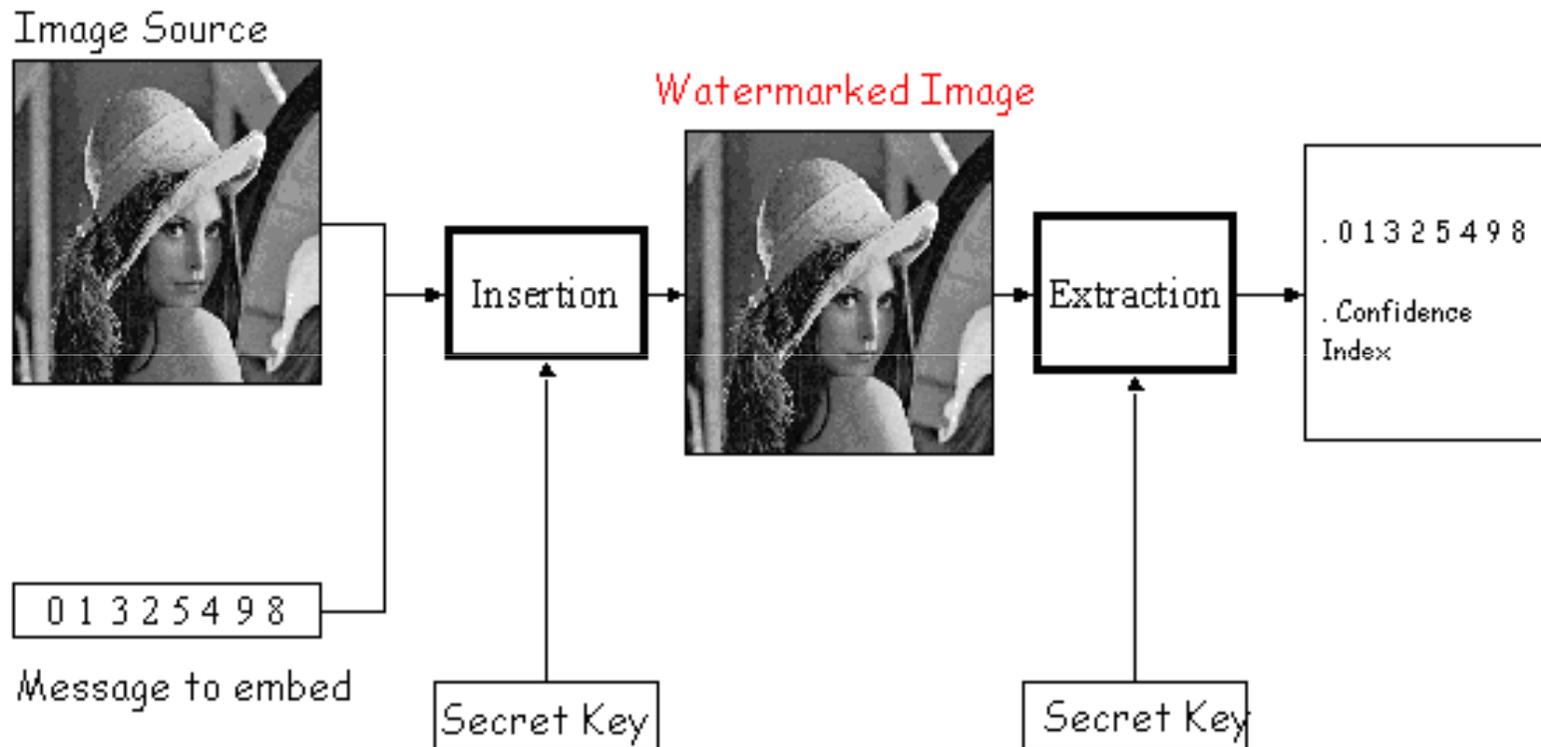
The algorithm is blind and symmetric



Insertion scheme



Blind symmetric algorithm



Outline

- Introduction
- Watermarking
 - Introduction
 - Study of an algorithm
 - ❖ **Spatial domain**
 - ❖ Transform domain
 - Applications
- Fingerprinting
- Applications
- Conclusion & future work

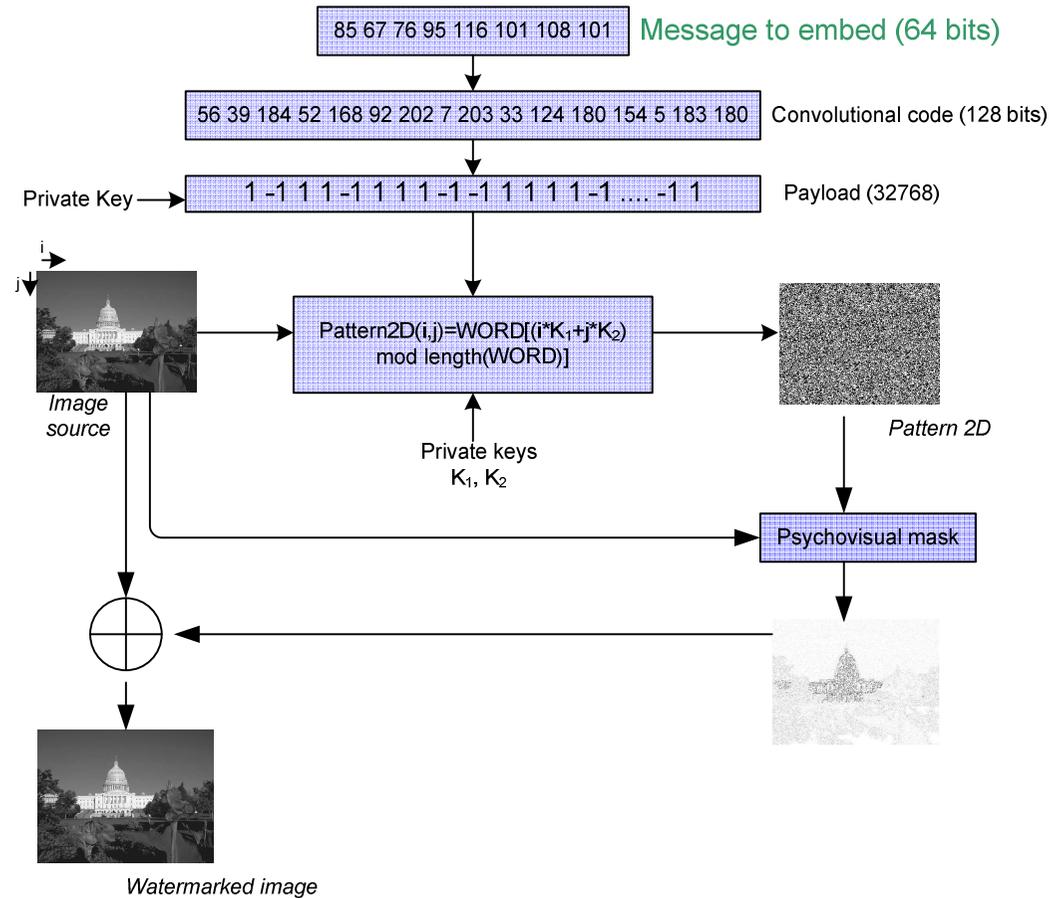


Watermarking scheme in spatial domain

- ❑ It is based on the redundancy of the message in the Image. The main blocks are:
 - ❖ Error correcting code : convolutional code
 - ❖ Pseudo random generator: Maximum Length Shift Register (MLS)
 - ❖ Algorithm to map the 1D code to the 2D Image.
 - ❖ Psycho-visual model in spatial domain

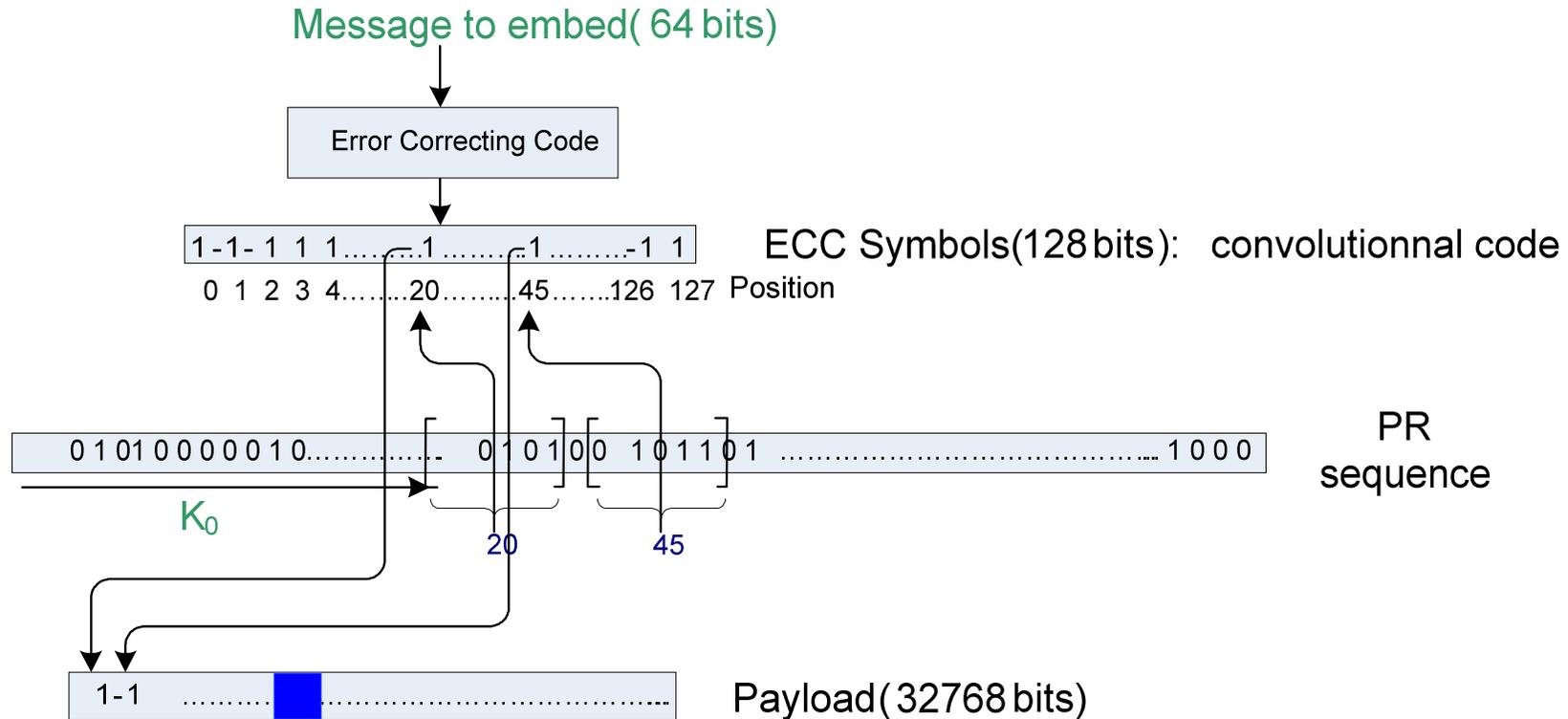


Spatial domain: insertion



Generation of the payload

1/4



(*) Introduced by Tirkel, 1994, « Electronic Water Mark »



Generation of the payload

2/4

❑ Error Correcting Code (ECC):

➤ Why:

- ❖ To spread randomly the possible corrupted bits along the payload.
- ❖ To recover the initial message if some bits are corrupted.

➤ How:

- ❖ We use convolutional code to encode the original message.
- ❖ We use Soft Viterbi to recover the original message.



Generation of the payload

3/4

□ Pseudo-random sequence (PR Sequence)

➤ why:

- ❖ To create a secure random sequence.

➤ How:

- ❖ We use a Linear Feedback Shift Register (LFSR). The maximum Length shift register is a class of cyclic codes. A linear code C is called a cyclic code if every cyclic code shift of a code vector in C is also a code vector in C . The generator polynomial for encoding a (n,k) cyclic code is given by:

$$g(X) = 1 + g_1(X) + g_2(X) + \dots + g_{n-k-1}(X) + X_{n-k}$$

The length of this cyclic sequence is $n = 2^m - 1$, where m is the number of stages.

- ❖ For secure extraction, we define a key Key0, as the secret seed for the generation of our LFSR code.
- Advantages:
 - ❖ The implementation is low cost.
 - ❖ This code generates a Gaussian noise appearance and provides interesting detection properties (So any attacks represented by a shifting in the LFSR code can easily be detected by cross-correlation with the original sequence).



Generation of the payload

4/4

□ Payload generation

➤ Why:

- ❖ To create a secure and robust sequence which carries the message to dissimulate.

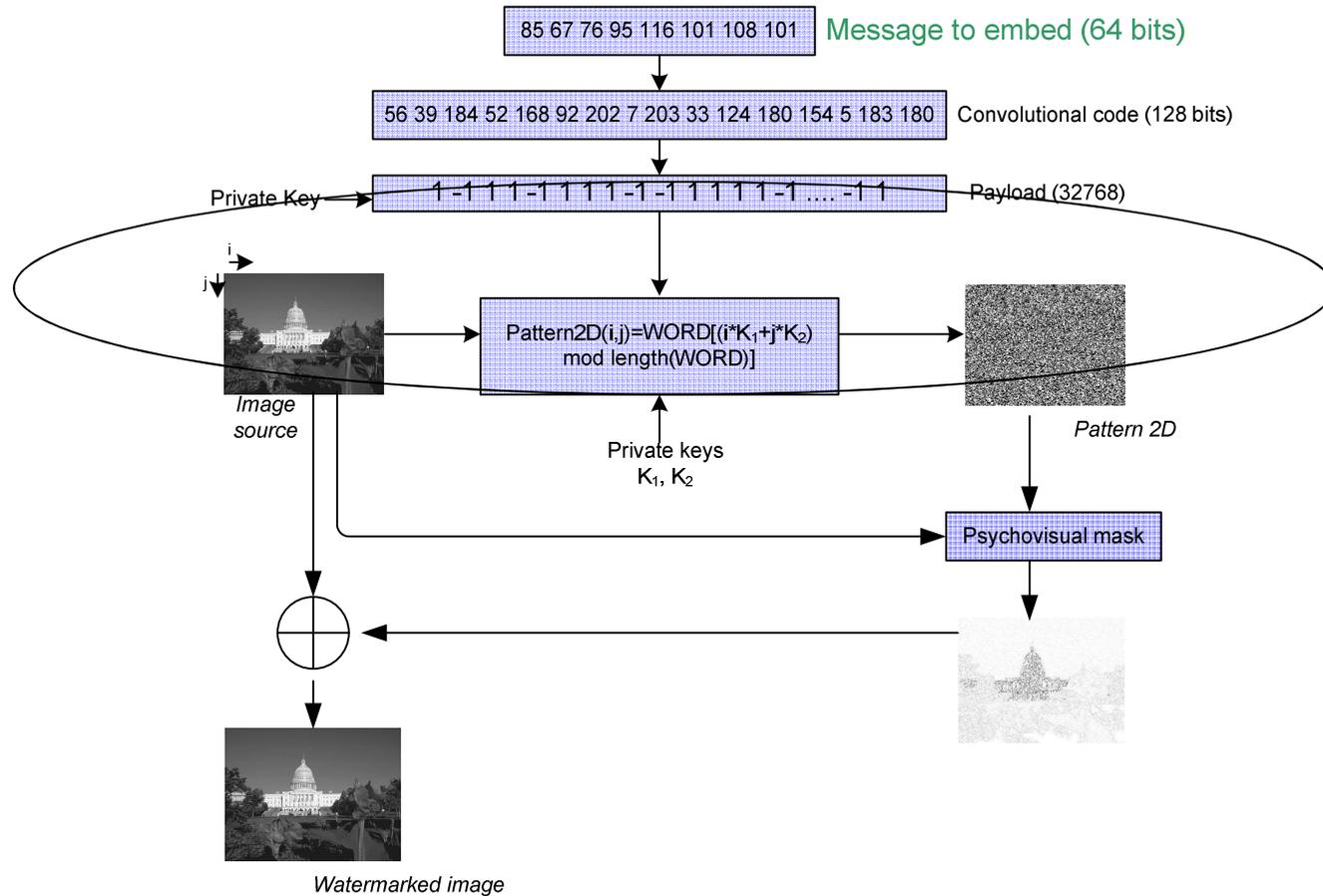
➤ How:

- ❖ We extract the first 7 bits of the PR sequence. This value corresponds to the index of the previous convolutional code.
- ❖ We extract the bit corresponding to the index of the convolutional code. This bit is the first bit of the new sequence called *Payload*.
- ❖ We continue until that all bits of the convolutional code are represented 256 times in *Payload*. The length of the *Payload* is 32768 bits.

$$Payload(j) = ECC(index)^2 - 1 \Big|_{index = \sum_{i=0}^6 PR_{7k+i} \cdot 2^i, k = j \Big|_{\#index < 256}, j = 1..32768}$$



Pattern 2D Generation



Pattern 2D Generation

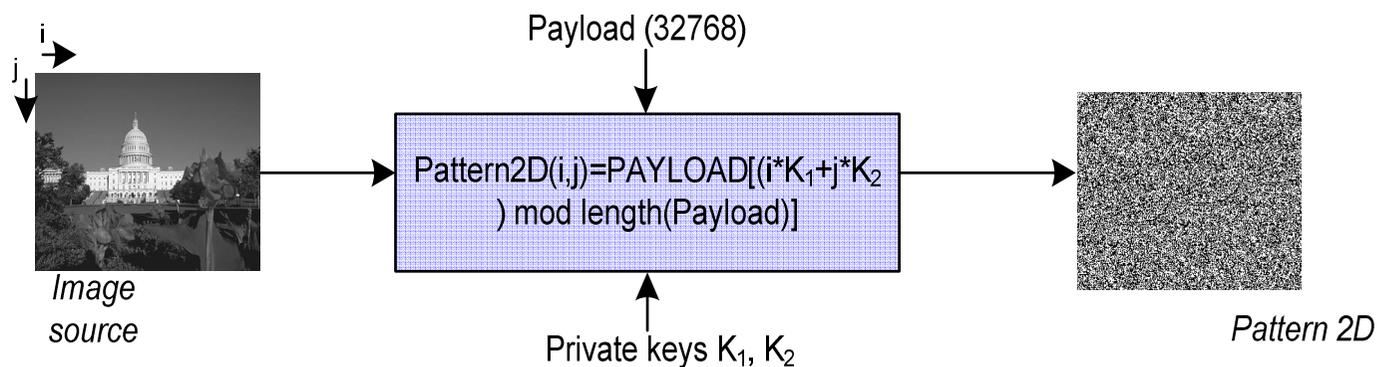
□ Pattern 2D generation

➤ Why

- ❖ To map the 1D cyclic payload onto the 2D matrix (Image).

➤ How

$$\text{Pattern}(i, j) = \text{Payload}(k) \mid k = (i \cdot K_1 + j \cdot K_2) \bmod (\text{card}\{\text{Payload}\})$$



Human Visual System

- The pixel intensity (luminance) are increased/decreased regarding contrast and neighbors.
- The amount of a modified pixel depends on its intensity (luminance): Weber-Fechner law



Weber-Fechner laws

It represent the amount of light necessary to add to a visual field of intensity B to become visible.

low intensities region:

$$\Delta B_T = \sqrt{x_1 x_2} * \beta * \left(\frac{\Delta B}{B} \right)_{\max} \text{ for } B \leq x_1$$

De Vries-Rose region:

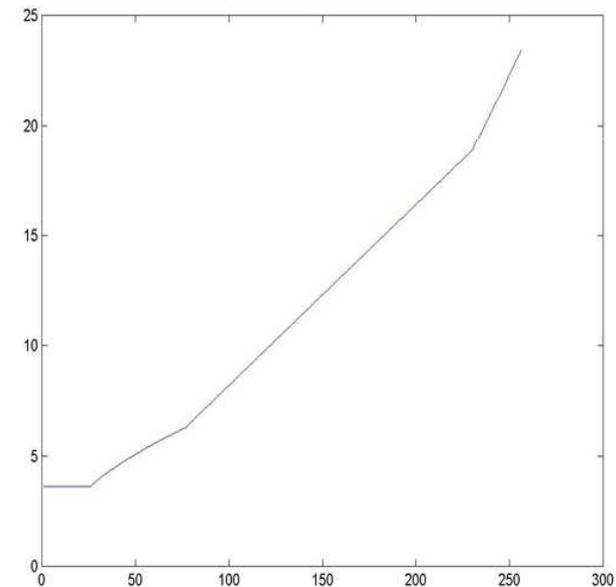
$$\Delta B_T = K_2 * \sqrt{B} \text{ for } x_1 \leq B \leq x_2$$

Weber region:

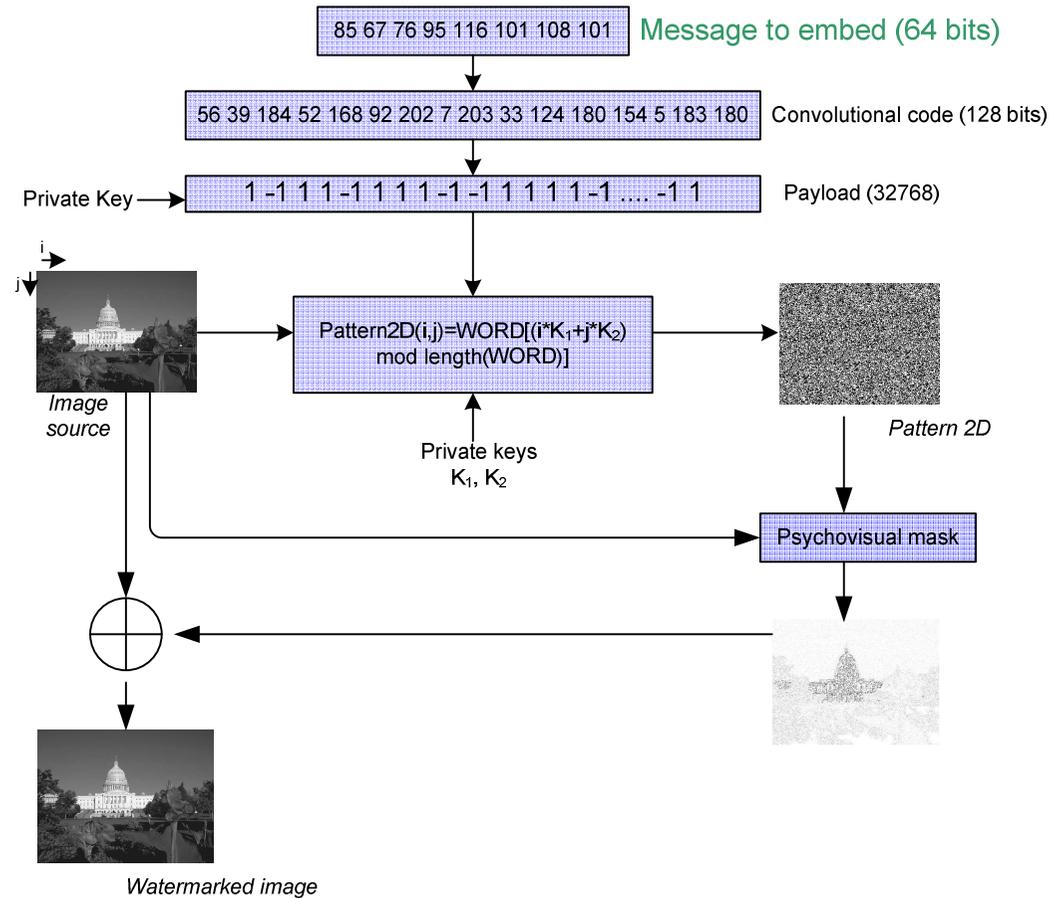
$$\Delta B_T = K_1 * B \text{ for } x_2 \leq B \leq x_3$$

Saturation region:

$$\Delta B_T = K_3 * B^2 \text{ for } B \geq x_3$$



Spatial domain: insertion



Watermarking pattern in spatial domain

□ Benefits:

- ❖ Robust against most of natural attacks.
- ❖ It is content dependent.
- ❖ Capacity allows to embed 64bits.
- ❖ It is fast to compute.

□ Weakness:

- ❖ Sensitive against geometrical distortion

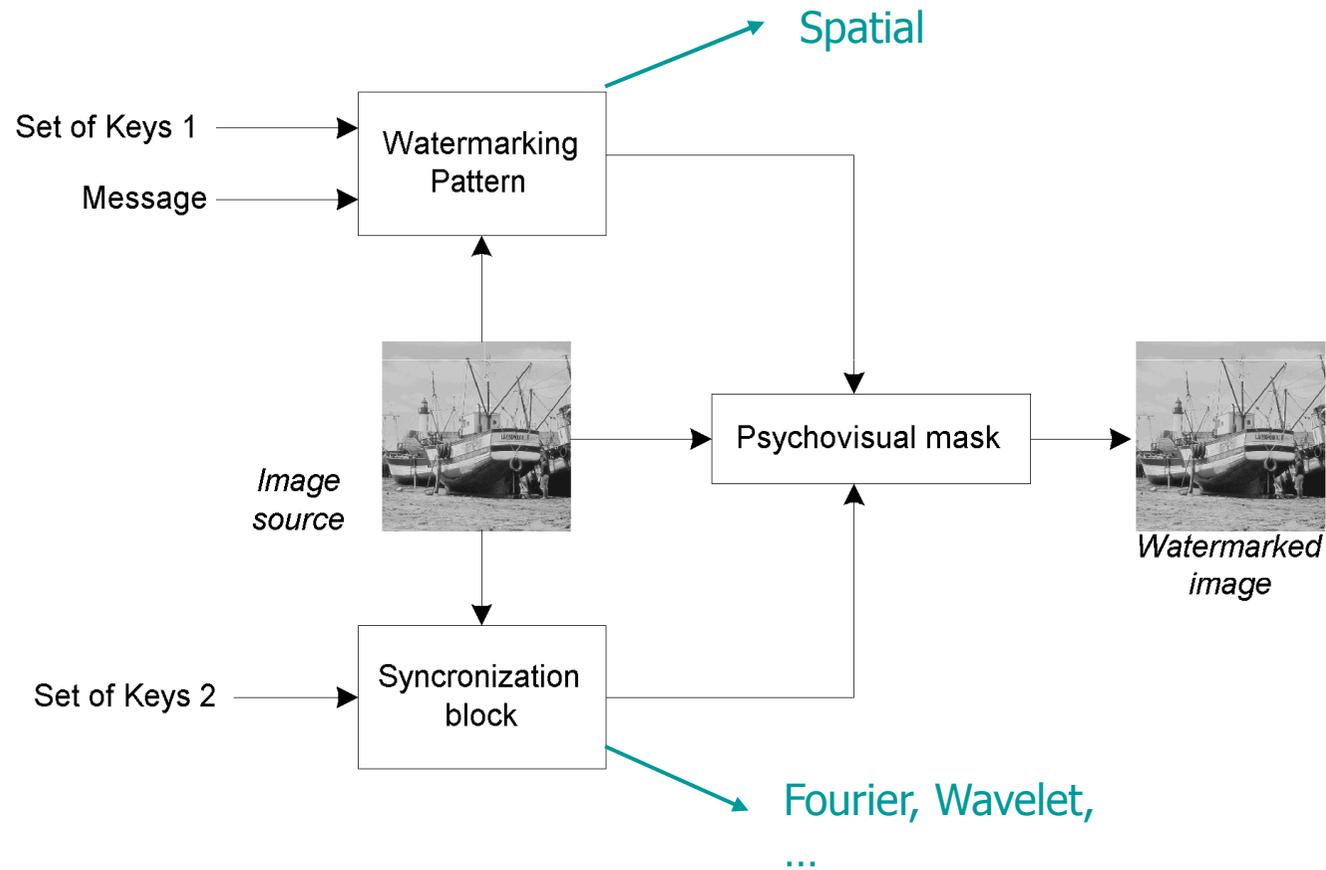


Outline

- Introduction
- Watermarking
 - Introduction
 - Study of an algorithm
 - ❖ Spatial domain
 - ❖ **Transform domain**
 - Applications
- Fingerprinting
- Applications
- Conclusion & future work



Insertion scheme



Watermarking in Fourier domain: requirements

- ❑ To be resistant against natural attacks such as JPEG.
- ❑ To be invisible.
- ❑ To extract some geometrical patterns in order to re-synchronize spatial domain.
- ❑ To keep the watermark secure

↪ Watermark is embedded in medium frequencies and managed by a key



Fourier domain

□ Resistance against scaling:

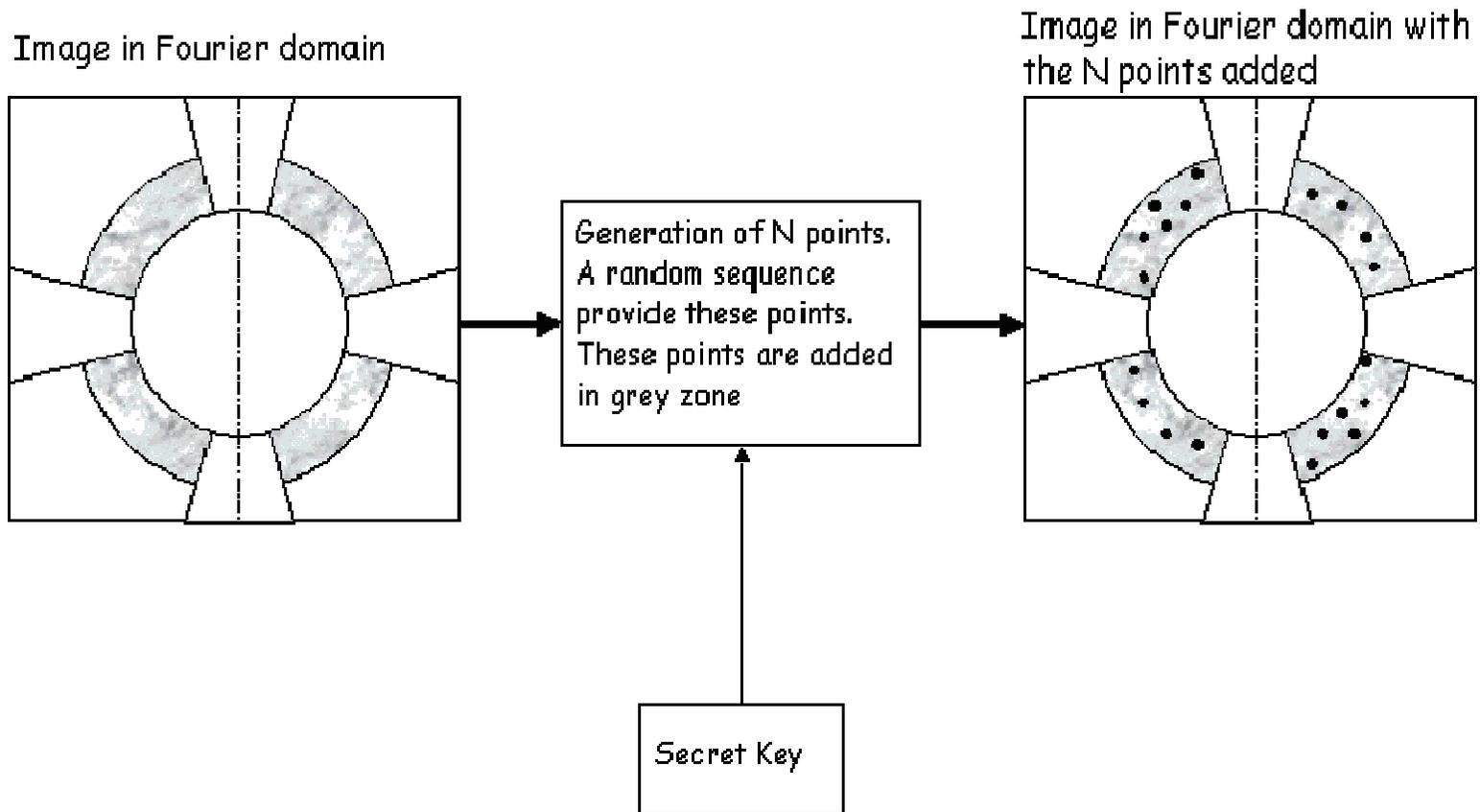
$$\begin{aligned}TF(f \circ S(S_x, S_y))(u, v) &= \alpha \int_{\mathbb{R}^2} f(S_x \cdot x, S_y \cdot y) e^{-(ux+vy)} dx dy \\ &= \alpha \int_{\mathbb{R}^2} f(X, Y) e^{-\left(\frac{ux}{S_x} + \frac{vy}{S_y}\right)} dx dy \\ &= TF(f) \cdot S\left(\frac{1}{S_x}, \frac{1}{S_y}\right)(u, v)\end{aligned}$$

□ Resistance against rotation:

$$\begin{aligned}TF(f \circ R_\theta)(u, v) &= \alpha \int_{\mathbb{R}^2} f(R_\theta(x, y)) e^{-(ux+vy)} dx dy \\ &= \alpha' \int_{\mathbb{R}^2} f(X, Y) e^{-((u, v) \cdot R_{(-\theta)}(X, Y))} dXdY \\ &= \alpha' \int_{\mathbb{R}^2} f(X, Y) e^{-R_\theta(u, v) \cdot (X, Y)} dXdY \\ &= \alpha'' TF(f) \cdot R_\theta(u, v)\end{aligned}$$



Fourier domain



Synchronisation block

❑ Benefits:

- ❖ Robust against geometrical distortion.
- ❖ Detect geometrical distortion.

❑ Weaknesses:

- ❖ Time consuming.
- ❖ Security is not proved.



Outline

- ❑ Introduction
- ❑ Watermarking
 - Introduction
 - Study of an algorithm
 - **Applications**
- ❑ Fingerprinting
- ❑ Applications
- ❑ Conclusion & future work



Applications: Fast Versioning

□ 3 different cases

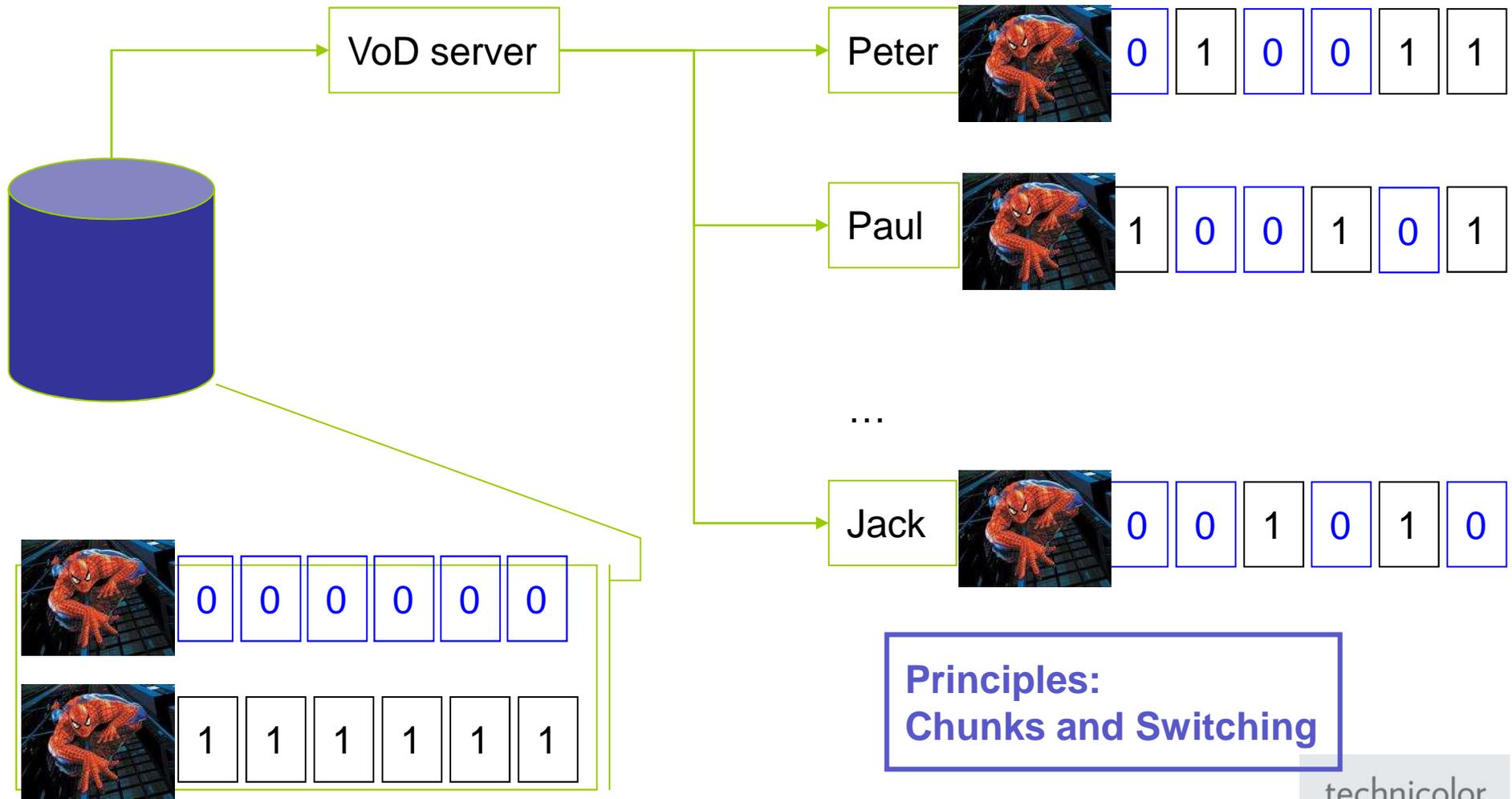
- Video on Demand (VoD)
 - ❖ Unicast
 - ❖ The server sends a personal copy.
- Blu-Ray Disc
 - ❖ Multicast
 - ❖ Hollywood prepares versioning, the device plays a personal copy.
- Setup box
 - ❖ Broadcast
 - ❖ The setup box outputs a personal copy.

□ Accusation is offline

- Hollywood forensics labs (subcontractor)



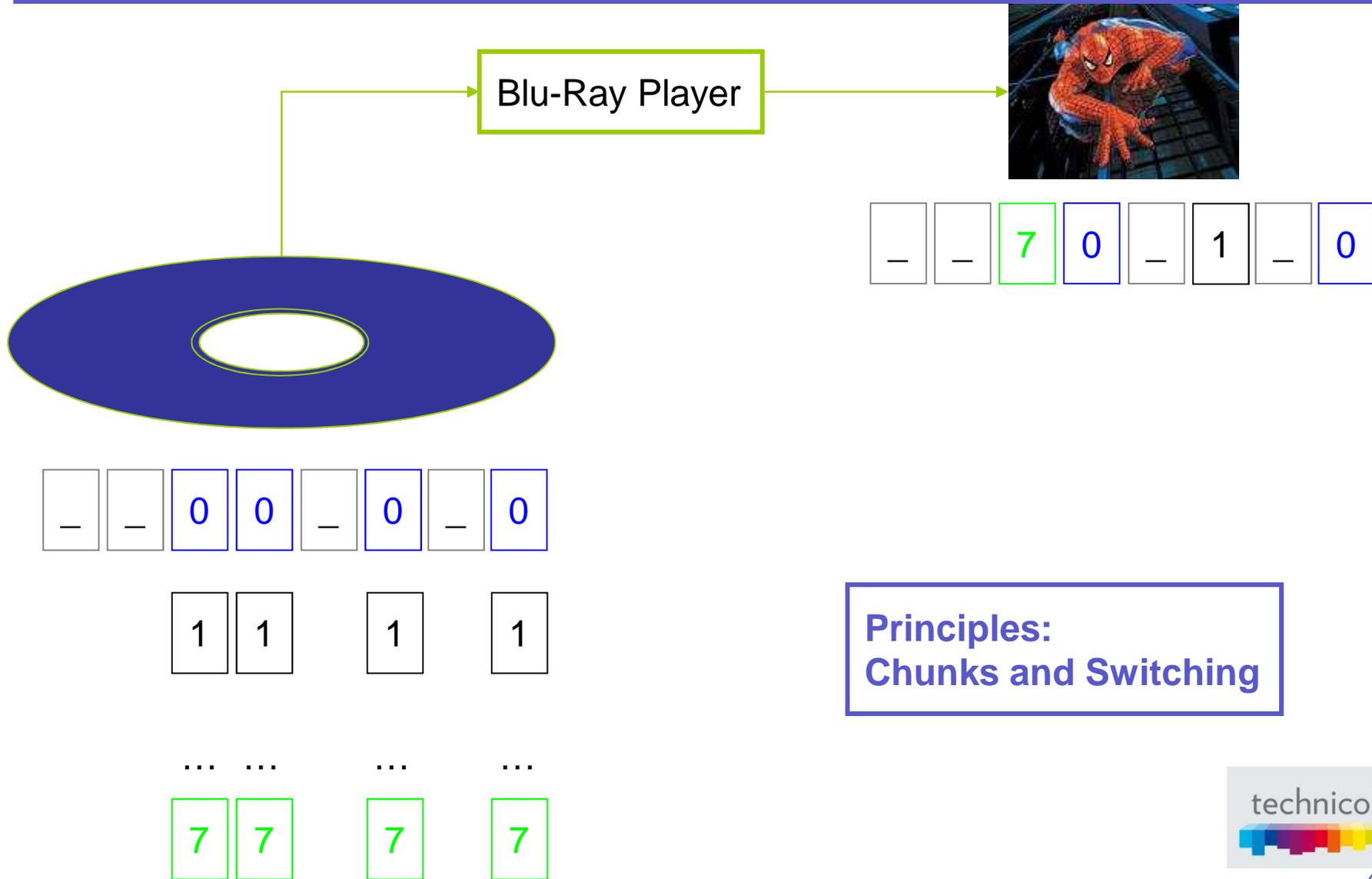
Video on Demand



**Principles:
Chunks and Switching**



Blu-Ray Disc



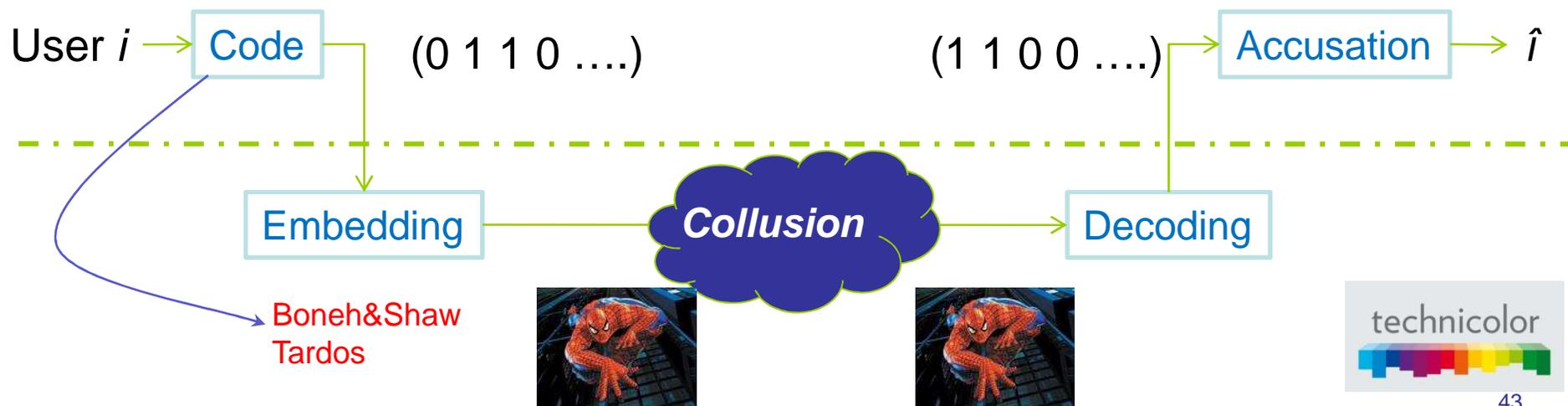
The Collusion

- ❑ Several dishonest users mix their versions to forge a pirate copy.
- ❑ Academic chimera?
 - The problem is trivial otherwise!
 $m = \lceil \log_Q(n) \rceil$ with Q the size of the alphabet
 - Closest example: The 12 Indian setup boxes
- ❑ Argument of the accused user:
 - « I am not a pirate but the victim of a collusion ».
 - The anti-collusion code convinces the judge this argument cannot hold.



Structure

- ❑ A 2 layers approach: data transport over a physical layer
 - The anti-collusion code (matrix $n \times m$)
 - ❖ Directory user \leftrightarrow sequence of m symbols
 - ❖ A unique sequence per user
 - The watermarking technique
 - ❖ Embed one symbol per content block
 - ❖ Text: synonyms to encode a binary symbol
 - ❖ Multimedia: a real-world technique
 - Any technique will fit? Requirements?



Watermarking: conclusion

- ❑ The design of a watermarking algorithm depends on the usage scenario
- ❑ The domain insertion and resistance have an important relationship
- ❑ The current watermarking schemes are not Kerckhoff compliant
- ❑ It is an intrusive technique

Outline

- Introduction
- Watermarking
- **Fingerprinting**
 - Introduction
 - Perceptual hash functions
 - Robust content representation
 - Fingerprint Database
- Applications
- Conclusion & future work

Fingerprinting principles

❑ What ?

Technique which **automatically extracts representative features**, called fingerprint, perceptual digest, or image/video/audio DNA

❑ Why ?

To **identify** image/video/audio or a fragment of image/video/audio

❑ Main properties

- ❖ To be **unique**
- ❖ To be **robust** against several distortions



Content Identification

- ❑ Perceptually similar contents may have very different binary representations
 - Calls for new technologies to unequivocally identify multimedia content
 - ❖ Robust hash
 - ❖ Visual hash
 - ❖ Perceptual hash
 - ❖ Soft hash
 - ❖ ...
 - ❖ Content fingerprinting (misleading terminology)

Introduction: Applications

❑ Concerning Security Applications, Image and Video forensics toolbox aims at deterring copyright infringements and tracing pirates.

- Video fingerprint

 - copy identification (on p2p networks or community sites UGC)

- Forensic tracking watermark

 - theater (and date + exhibition time) identification

- Analysis of geometric (keystone) distortions

 - localization of the pirate in the theater

- Sensor forensics

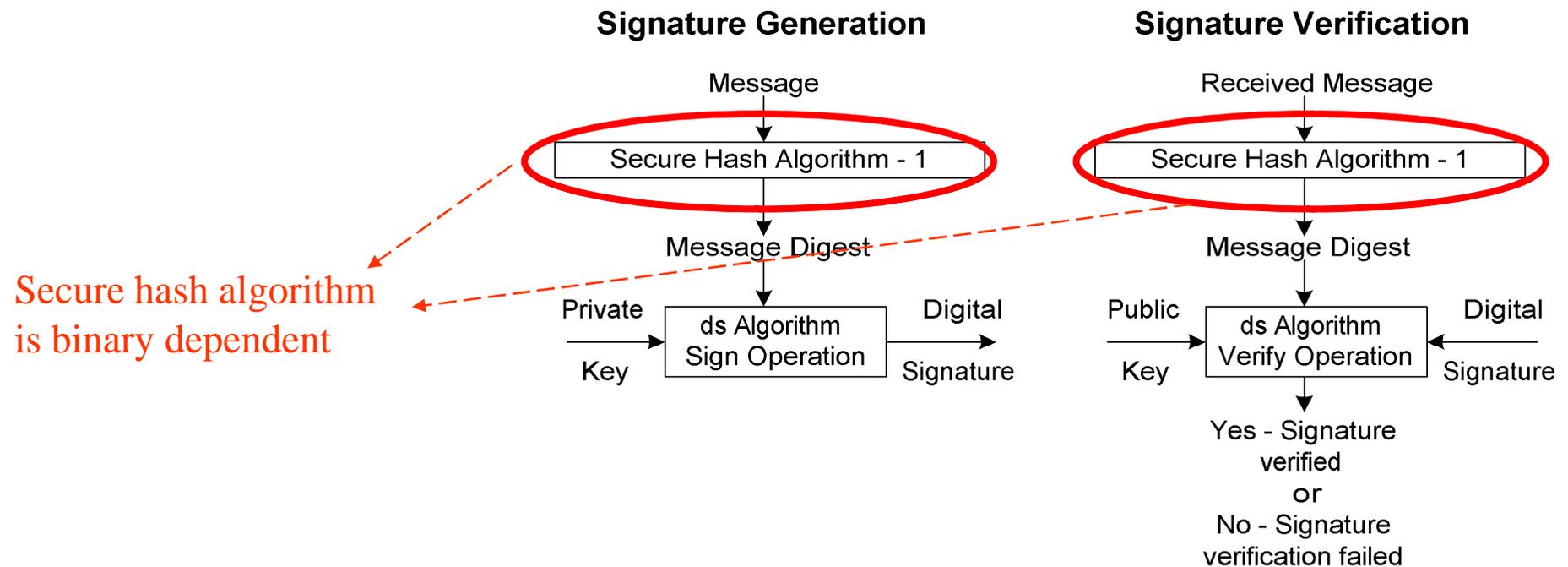
 - camcorder identification

Outline

- Introduction
- Watermarking
- Fingerprinting
 - Introduction
 - **Perceptual hash functions**
 - ❖ Generic constructions
 - ❖ Global features extraction
 - ❖ Local features extraction
 - Robust content representation
 - Fingerprint Database
- Applications
- Conclusion & future work

Digital Signatures

- ❑ Authentication of document
- ❑ Data integrity
- ❑ Non-repudiation



Cryptographic Hash Functions

❑ Ease of computation

- For every input \mathbf{x} (from domain of f) $f(\mathbf{x})$ is 'easy' to compute.

❑ Fixed output bit length

- A hash function f maps an input \mathbf{x} of arbitrary bit length to an output $f(\mathbf{x})$ of fixed bit length.

❑ Pre-image resistant

- Given any image \mathbf{y} , for which there exists an \mathbf{x} with $f(\mathbf{x})=\mathbf{y}$, it is computationally infeasible to compute any pre-image \mathbf{x}' with $f(\mathbf{x}')=\mathbf{y}$.

❑ Weak collision resistance

- Given any pre-image \mathbf{x} it is computationally infeasible to find a 2nd pre-image $\mathbf{x}' \neq \mathbf{x}$ with $f(\mathbf{x})=f(\mathbf{x}')$.



Perceptual Hash Functions

- ❑ Heavily inspired from cryptographic one way hash functions
 - Two perceptually similar contents should hash to the same binary digest
 - Two perceptually dissimilar contents should hash to different binary digests
- ❑ Combination of cryptographic hash function properties with signal processing constraints
 - Easy to compute, very fast, resistant against collisions
 - Resistant against signal processing distortions (compression, resizing,...)

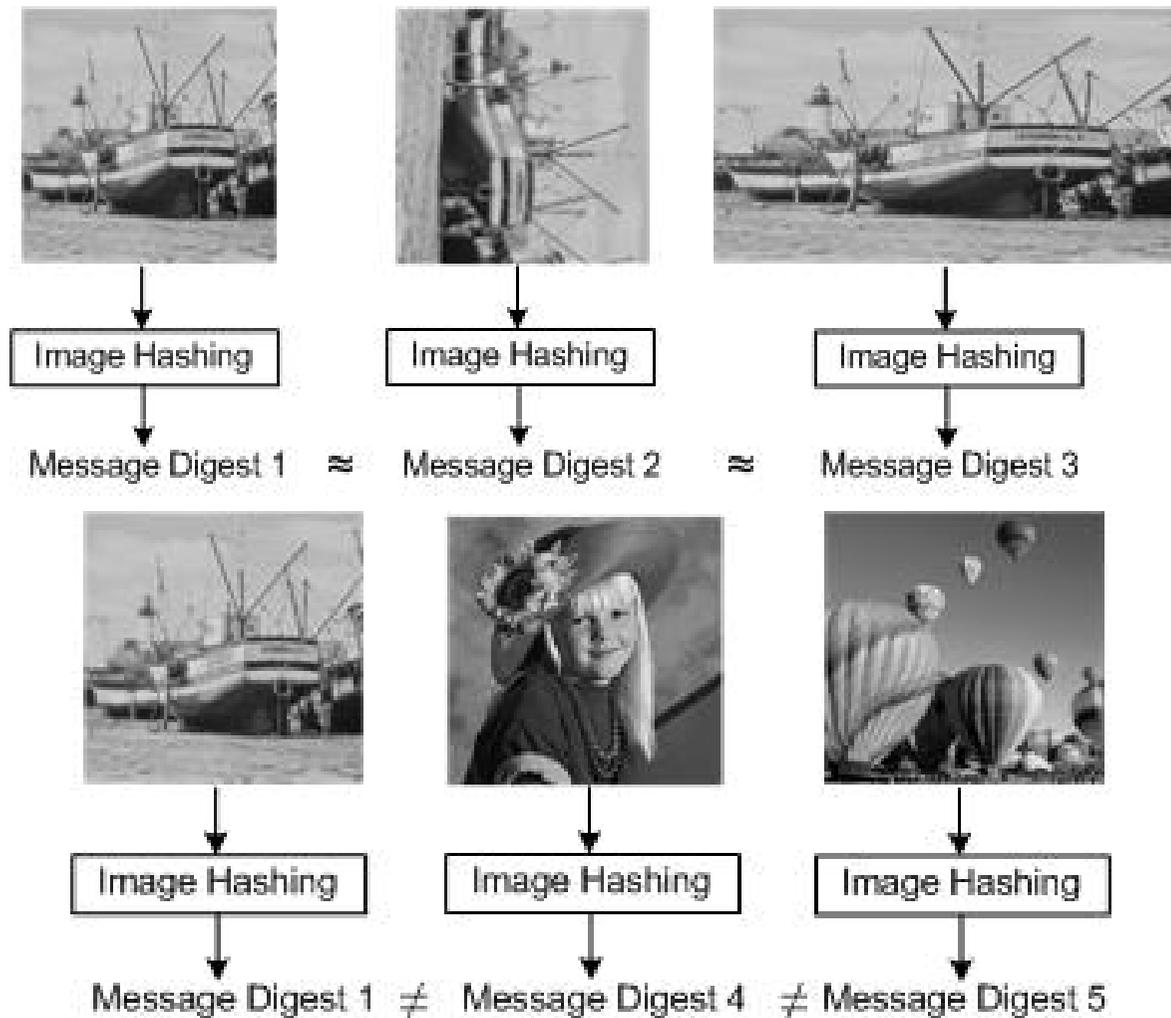


Definition

- ❑ Easy to compute
- ❑ Fixed output bit length
- ❑ Pre-image resistant
- ❑ A soft (perceptual) digest
 - The image $f(\mathbf{x})$ must be resistant and robust, i.e. it shall remain nearly the same before and after attacks, if these attacks do not alter the perceptual components of the content i.e. $f(\mathbf{x}) \approx f(\mathbf{x}')$ if $\mathbf{x} \approx \mathbf{x}'$, $\mathbf{x} \approx \mathbf{x}'$ meaning that \mathbf{x}' is a perceptually similar version of \mathbf{x} e.g. same visual content.
- ❑ Weak collision resistance
 - Given any pre-image \mathbf{x} it is computationally infeasible to find a 2nd pre-image $\mathbf{x}' \neq \mathbf{x}$ with $f(\mathbf{x}) = f(\mathbf{x}')$. Two pre-images \mathbf{x} , \mathbf{x}' are different if and only if their contents are perceptually different.



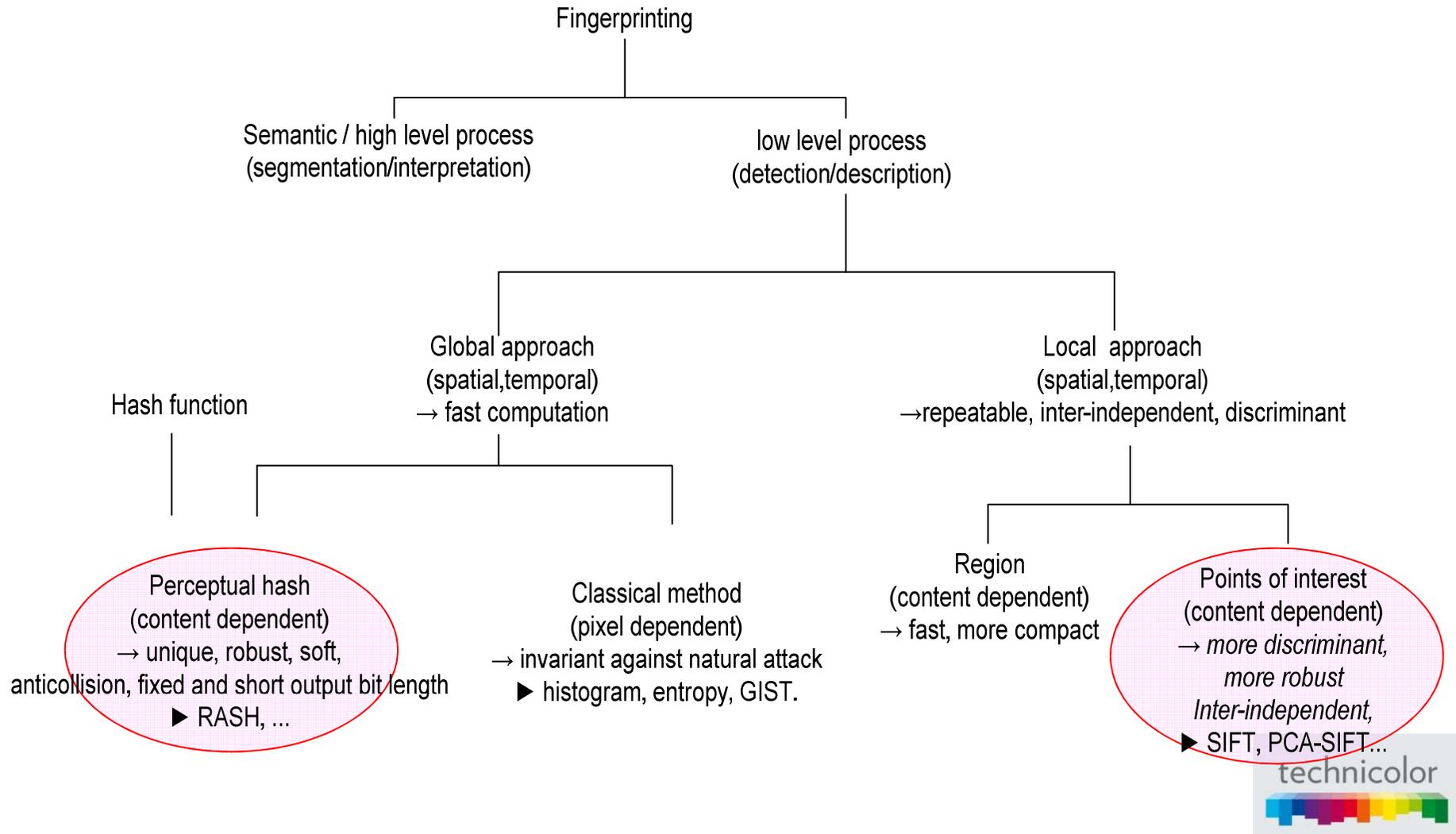
Properties



Outline

- ❑ Introduction
- ❑ Watermarking
- ❑ Fingerprinting
 - Introduction
 - Perceptual hash functions
 - ❖ Generic constructions
 - ❖ Global features extraction
 - ❖ Local features extraction
 - Robust content representation
 - Fingerprint Database
- ❑ Applications
- ❑ Conclusion & future work

Generic Constructions



Generic Constructions

- ❑ Global approach (fast, robust against natural distortion)
 - Classical methods which are pixel dependent
 - Perceptual hash which are content dependent with hash function constraints

- ❑ Local approach (inter-independent, discriminant, strong robustness)
 - Points of interest



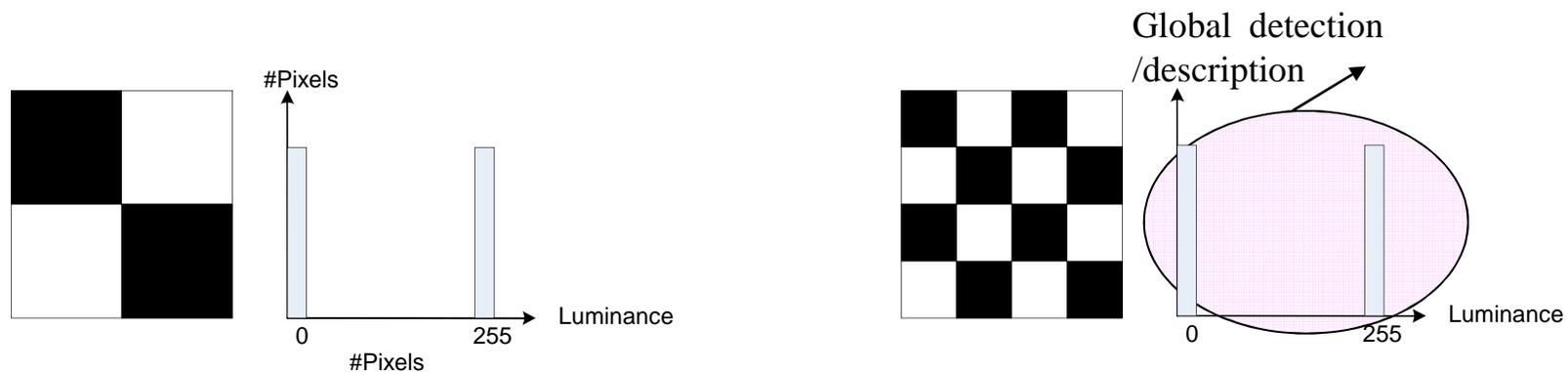
Outline

- ❑ Introduction
- ❑ Watermarking
- ❑ Fingerprinting
 - Introduction
 - Perceptual hash functions
 - ❖ Generic constructions
 - ❖ **Global features extraction**
 - ❖ Local features extraction
 - Robust content representation
 - Fingerprint Database
- ❑ Applications
- ❑ Conclusion & future work

Global Fingerprinting

- Global fingerprinting manages an image as a global content and describes the global content with as set of global attributes.
 - E.g: Luminance Histogram

$$\text{Hist}(k) = |\{ \text{Lum}(i, j) = k, (i, j) \in [1:w] \times [1:h] \}|, k = 0 : 255$$



□ Benefits

- Fast

□ Weaknesses

- lack of robustness against strong local distortion and global distortion
- Not collision resistant, false positive issue



Global Attributes

❑ Colour:

- Usually combined with texture. It is sensitive to color attacks (gamma, contrast, illumination conditions). E.g : Luminance histogram

❑ Texture:

- Discriminant (usually defined as a low level descriptor) but sensitive text addition or redundant pattern. It generates some collisions in case of scalability. E.g : Gabor filters and Wavelet decomposition [3].

❑ Shape:

- Two main classes: region based and edge based (e.g Fourier [4]). First one is more robust but less discriminant. The second one is largely used in local fingerprinting as descriptor (SIFT).

❑ Motion:

- Motion is only video oriented. It describes motion vector such as in MPEG. It is sensitive to motion algorithms and to bit rate compression changes.



Outline

- Introduction
- Watermarking
- Fingerprinting
 - Introduction
 - Perceptual hash functions
 - ❖ Generic constructions
 - ❖ Global features extraction
 - ❖ **Local features extraction**
 - Robust content representation
 - Fingerprint Database
- Applications
- Conclusion & future work

Local Fingerprinting

Local fingerprinting manages an image (video) as a multitude of characteristics spatially (spatio-temporal) localized



A Two Steps Process

1. Detection of points of interest.

- Detection of repeatable key points i.e. location is detectable after attacks.
- Points detection must be robust against distortions e.g. change of scale, rotation, filtering...

2. Description of points of interest.

- Characterization of each key point.
- The descriptor must be
 - ❖ Discriminant i.e. it provides representative and different value for each different content.
 - ❖ invariant to a certain number of transformations.



Detection Criteria

□ Repeatability:

- It defines the ability of a given algorithm to detect similar structures before and after distortions
- It highlights the scale-space representation: the ability to detect structures at different scales
- It is given by precision/recall, or repeatability:

$$\rho = \frac{|\{(P_1, P_2) | P_1 \in L_{\text{ref}}, P_2 \in L_{\text{copy}}, P_2 = T(P_1)\}|}{|L_{\text{repeatable}}|}$$

□ Accuracy:

- Accurate localization of the detected feature points (pixel, ...)

□ Complexity:

- Computational cost of detecting feature points (time, memory)



Description Criteria

❑ Discriminative power

- A local fingerprint is discriminant if it uniquely characterizes the local zone of interest
- Discriminant descriptors minimize collisions

❑ Invariance

- The invariance (or robustness) is evaluated against a range of transforms (or distortions)
- A local fingerprint is invariant against a given transform if it
- remains almost the same before and after image transform

⇒ An efficient descriptor performs a **trade-off** between discriminative power and invariance



Outline

- ❑ Introduction
- ❑ Watermarking
- ❑ Fingerprinting
 - Introduction
 - Perceptual hash functions
 - Robust content representation
 - ❖ Image Fingerprinting
 - ❖ Video Fingerprinting
 - Fingerprint Database
- ❑ Applications
- ❑ Conclusion & future work

Random Partitioning Hash (*)

- ❑ Step 1: Random tiling transform and statistics calculation
- ❑ Step 2: Randomized rounding
- ❑ Step 3: Creation of an intermediate secure and robust digest.
- ❑ Step 4: Mapping the current intermediate hash value from step 3 into an shorter digest

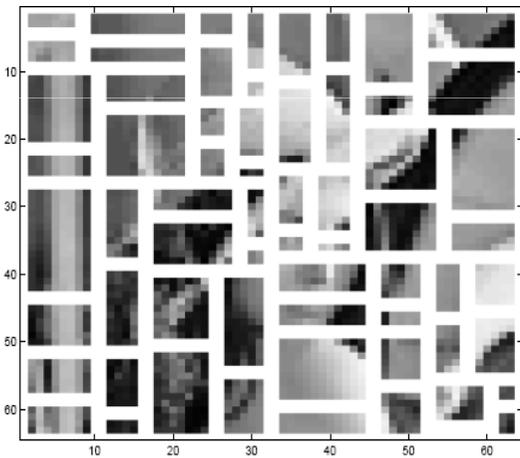
(*) R. Venkatesan, S.M. Koon, M.H. Jakubowski, and P. Moulin, "Robust image hashing", ICIP, 2000.



Random Partitioning Hash

□ Step 1: Random tiling transform and features extraction

- First, a wavelet transform is applied to the image
- Then, the wavelet subbands are partitioned into random tiles (seed K):



- Finally, l features, noted \mathbf{m} , are calculated from the subband random tiling:
 - ❖ Averages of coefficients in the rectangles in the coarse subband.
 - ❖ Variances in the other subbands.

□ Step 2: Randomized quantization

$$\mathbf{x} = \mathbf{Q}(\mathbf{m}, K) \in \{0, \dots, 7\}^l$$

Dimension unchanged



Random Partitioning Hash

□ Step 3: Creation of an shorter intermediate secure and robust digest

- The vector \mathbf{x} is decoded by a first order Reed-Muller error correcting code decoder D .

$$\mathbf{h} = D(\mathbf{x}) \in \{0,1\}^n$$

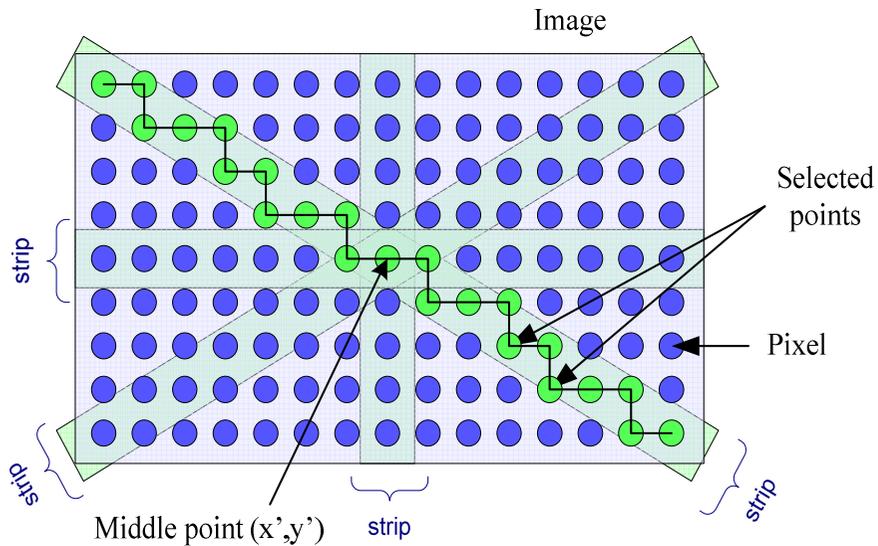
- \mathbf{h} is shorter than \mathbf{x} ($n < l$) and its symbols are uncorrelated, hence avoiding potential collision.

□ Step 4: Dimension reduction

- The vector \mathbf{h} is reduced using another decoder stage



Radon Soft Hash (RASH) (*)



1. Select a strip (set of points on a line passing through the image center), with orientation $\theta \in [1:180]$
 2. Compute the pixel variance
- ⇒ 180-D feature vector

□ Properties

➤ Resizing

$$g(ax, ay) \leftrightarrow \frac{1}{|a|} Rg(ap, \theta)$$

➤ Rotation by an angle θ_0

$$g(x \cos \theta_0 - y \sin \theta_0, x \sin \theta_0 + y \cos \theta_0) \leftrightarrow Rg(p, \theta + \theta_0)$$

(*) F. Lefebvre, B. Macq, "RASH:RADon Soft hash algorithm", European Signal Processing Conference 2002, Toulouse, France



RASH in Action



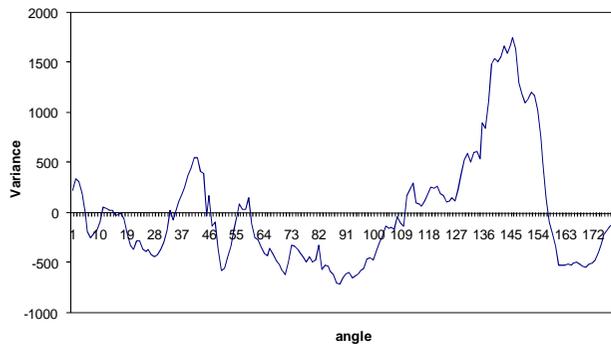
Monster



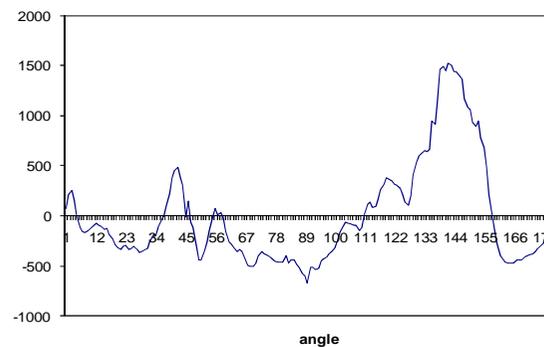
spatial blur



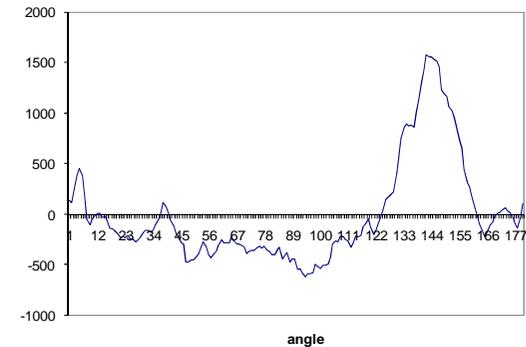
Camcorder and cropping



angle

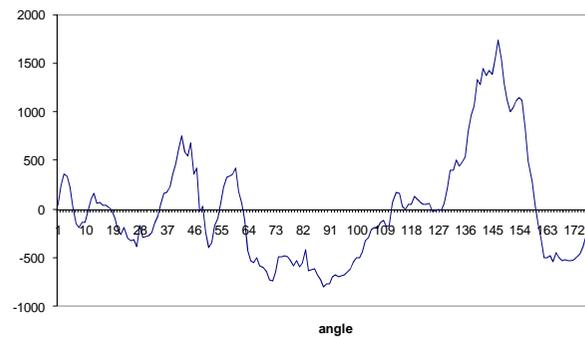


angle



angle

Rotation 1°



angle



RASH Fact Sheet

❑ Benefits:

- Robust against most of natural attacks.
- It is content dependent.
- Two close contents have close visual digests.
- Very short visual digest (180 bits/image)
- It is very fast to compute

❑ Weaknesses:

- Sensitive against cropping attack
- Not discriminant in case of local distortion
- One-way property is not proved



A Two Steps Process

1. Detection of points of interest.

- Detection of repeatable key points i.e. location is detectable after attacks.
- Points detection must be robust against distortions e.g. change of scale, rotation, filtering...

2. Description of points of interest.

- Characterization of each key point.
- The descriptor must be
 - ❖ Discriminant i.e. it provides representative and different value for each different content.
 - ❖ invariant to a certain number of transformations.



Feature Points Detectors

- ❑ The main local/key/feature/interest points detectors are based on:
 - Radial symmetry interest points detector
 - Moravec detector
 - Harris corners detector
 - DoG detector
 - Harris-Laplace

Harris Detector

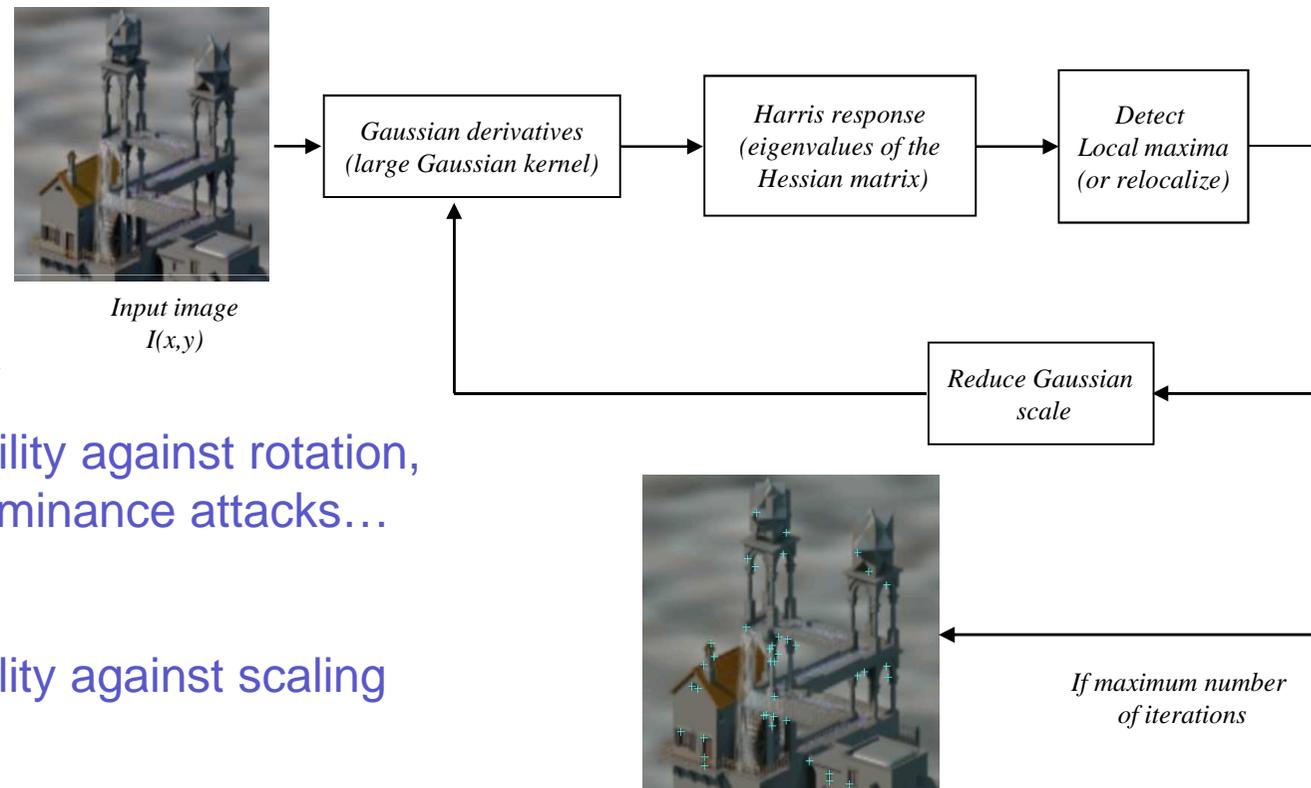
Detection of salient points characterized by a high photometric frequency in several directions (*)

□ Benefits

- Fast
- High accuracy
- High repeatability against rotation, filtering and luminance attacks...

□ Cons

- Low repeatability against scaling
- Complex



(*) **Chris Harris and Mike Stephen**, "A combined Corner And Edge Detector", Proceedings of The Fourth Alvey Vision Conference, Manchester, pp 147-151. 1988.

Scale-Space Representation

- The scale-space representation addresses the scale invariance.
- The linear scale-space representation is the solution of the diffusion equation:

$$\partial_{\sigma} \mathbf{G} = \frac{1}{2} \nabla^2 \mathbf{G} \quad (1)$$

- It can be represented by the convolution with a Gaussian kernel $\mathbf{G}(i,j,\sigma) = (\mathbf{g}_{\sigma} * \mathbf{f})(i,j)$ with

$$\mathbf{g}_{\sigma}(i,j) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{i^2 + j^2}{2\sigma}\right] \quad (2)$$

- By replacing $\mathbf{G}(i,j,t)$ from (2) in (1), an approximation of the first term is:

$$\partial_{\sigma} \mathbf{G} = \frac{1}{\sqrt{2\pi} \delta\sigma} \left[\frac{1}{(\sigma + \delta\sigma)} \exp\left(\frac{i^2}{2(\sigma + \delta\sigma)^2}\right) - \frac{1}{\sigma} \exp\left(\frac{j^2}{2\sigma^2}\right) \right]$$

- If we compute the Laplacian of the Gaussian, an approximation of difference of Gaussian (DoG) is:

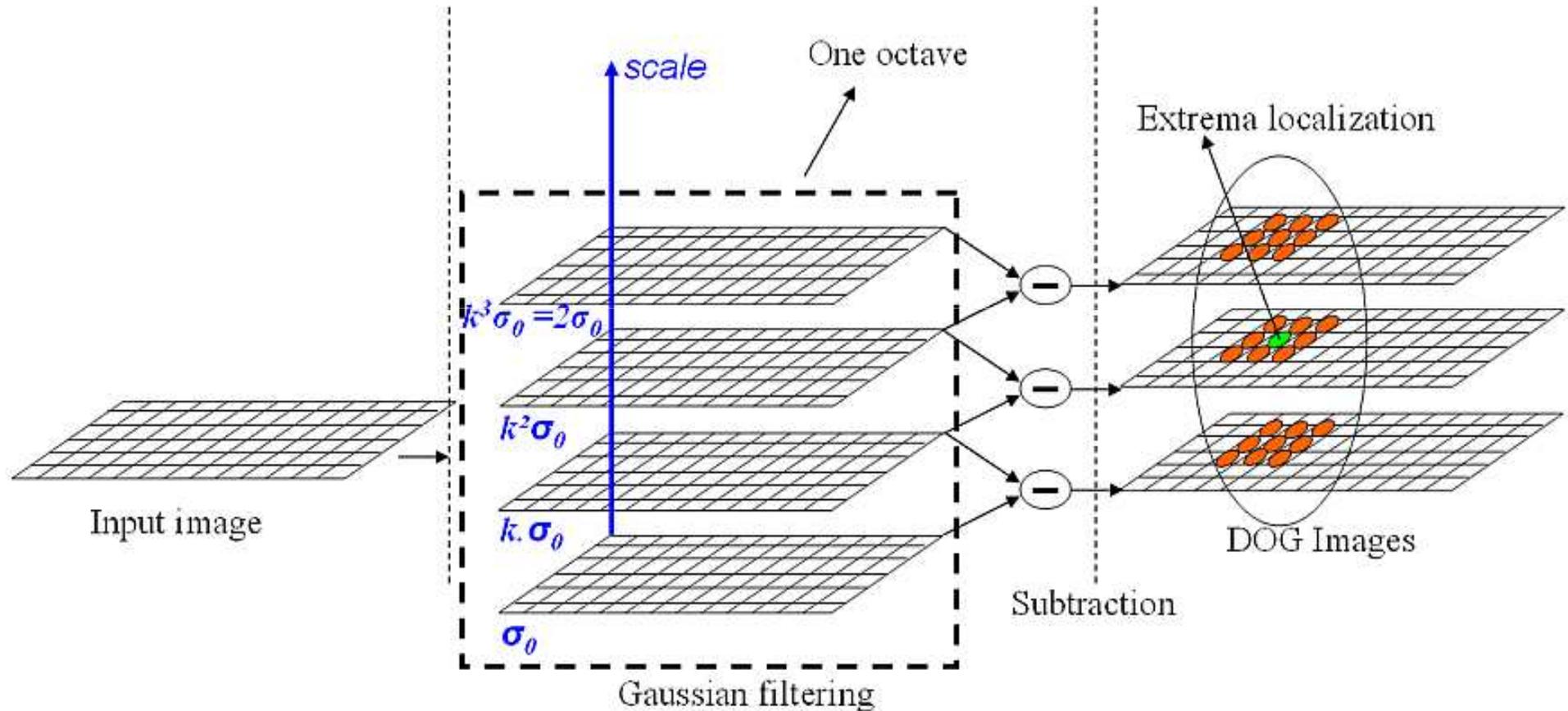
$$\text{DoG} \approx \sigma^2 \nabla^2 \mathbf{G}$$

Normalized term given by (*) for the scale invariance



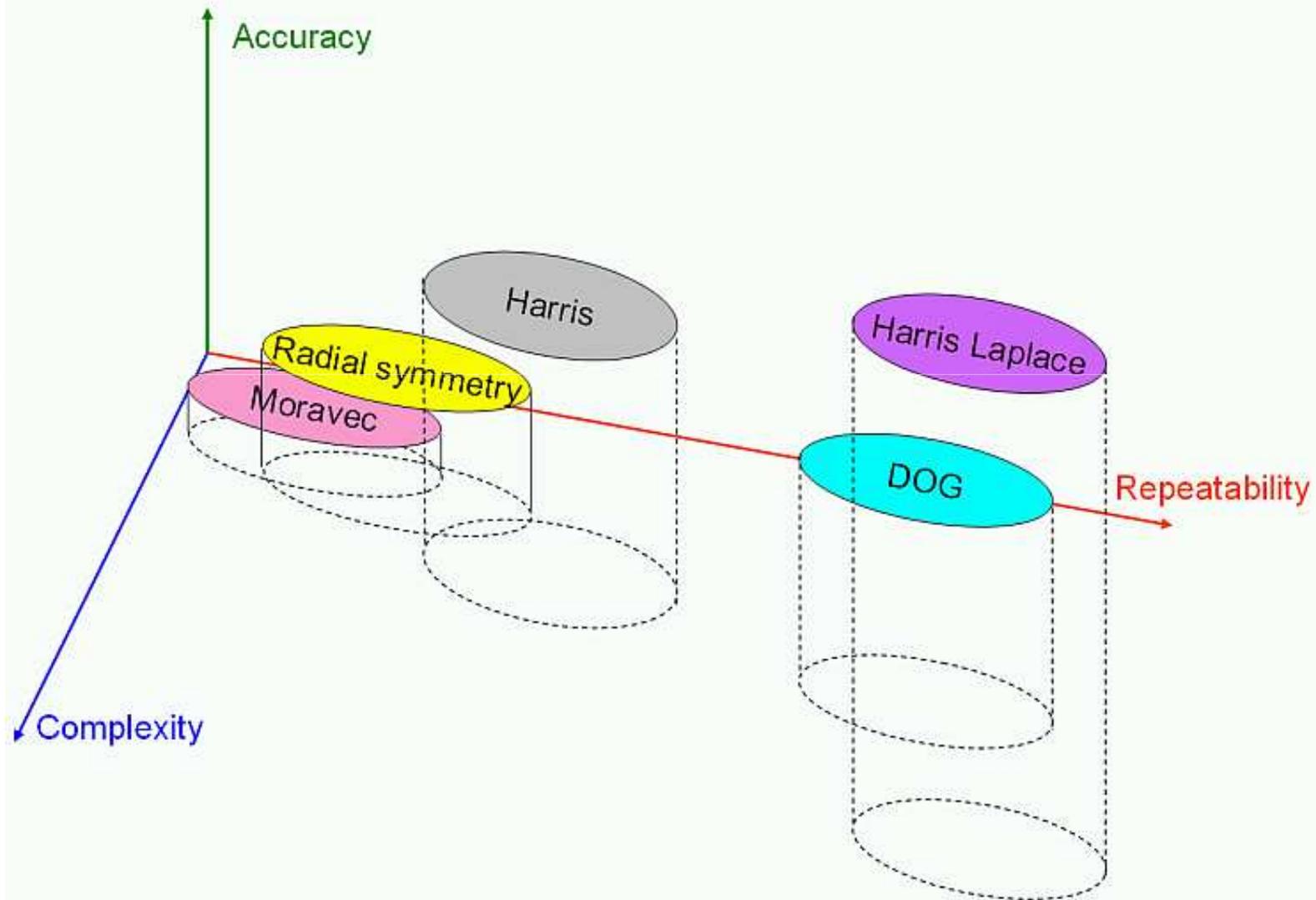
(*) Laptev and T. Lindeberg, "Space-time Interest Points", In Proc. ICCV, France, pp. 432-439, 2003

DoG: Finding Key Points (*)



(*) D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", IJCV, pp. 91-110, 2004.

Trade-off



A Two Steps Process

1. Detection of points of interest.

- Detection of repeatable key points i.e. location is detectable after attacks.
- Points detection must be robust against distortions e.g. change of scale, rotation, filtering...

2. Description of points of interest.

- Characterization of each key point.
- The descriptor must be
 - ❖ Discriminant i.e. it provides representative and different value for each different content.
 - ❖ invariant to a certain number of transformations.



Local Jet Descriptor

A compact representation of the Taylor expansion of the image luminance around a feature point

$$\text{LocalJet}(\mathbf{I}, x, y) = \left(\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y}, \frac{\partial^2 \mathbf{I}}{\partial x^2}, \frac{\partial^2 \mathbf{I}}{\partial y^2}, \frac{\partial^2 \mathbf{I}}{\partial xy} \right)$$

□ Pros

- Low dimensionality
- Fast computation
- Robustness against luminance attacks

□ Cons

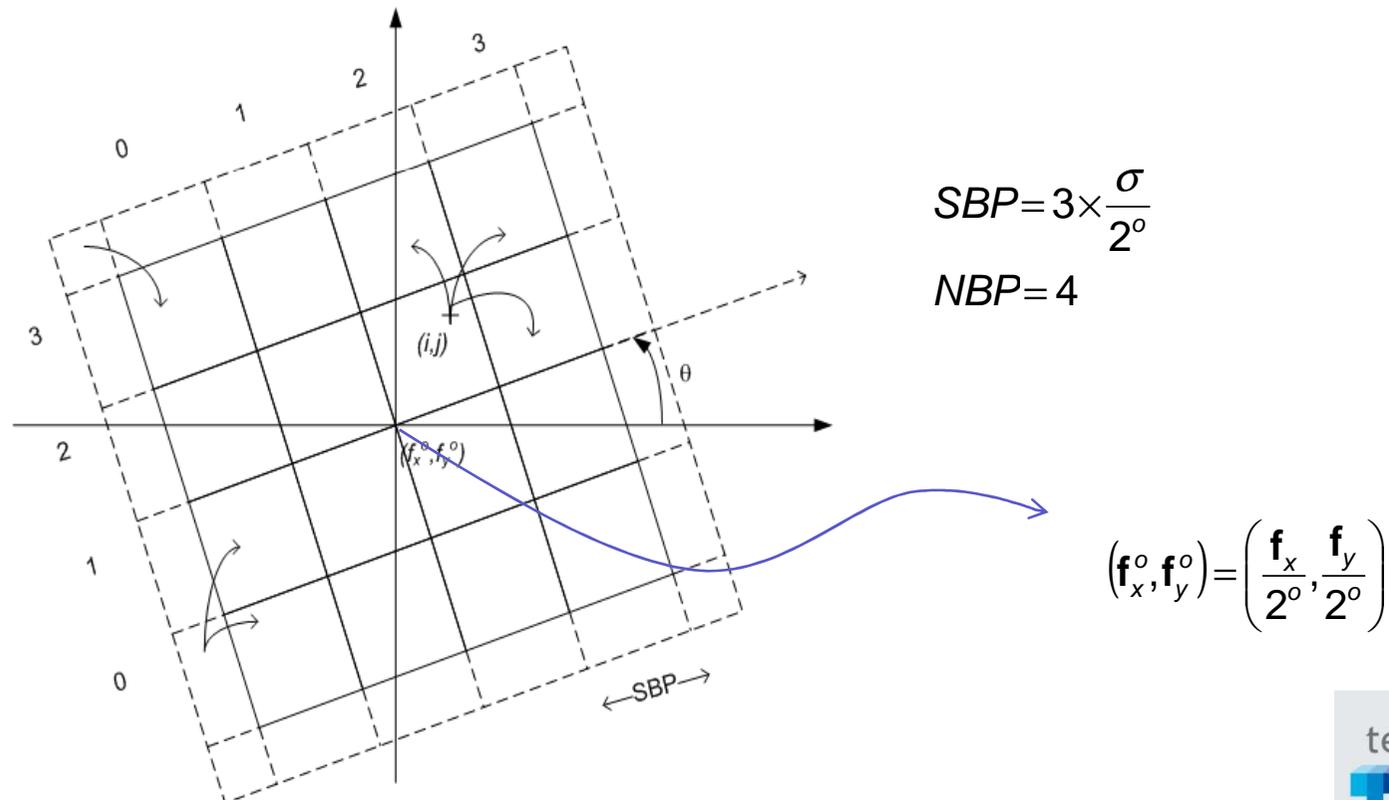
- Low discriminative power
- Low robustness against scaling and rotation



SIFT Descriptor

□ Scale Invariant Feature Transform (*)

- Distribution of gradient orientations in the spatial neighborhood of the Gaussian image \mathbf{G}_σ^o (octave o , scale σ) where the feature point was detected



(*) D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", IJCV, pp. 91-110, 2004.

SIFT Descriptor

- For each pixel (i,j) in the neighborhood, the magnitude $m(i,j)$ of the gradient and its orientation $\theta(i,j)$ are computed

$$m(i,j) = \sqrt{\mathbf{G}_x(i,j)^2 + \mathbf{G}_y(i,j)^2} \quad \theta(i,j) = \text{atan}\left(\frac{\mathbf{G}_y(i,j)}{\mathbf{G}_x(i,j)}\right)$$

- Computation of the orientation relative to the average local orientation $\bar{\theta}$

$$\alpha(i,j) = \theta(i,j) - \bar{\theta}$$

- Quantization into 8 bins

- The contribution of each pixel (i,j) is weighted with a Gaussian function

$$w(i,j) = \exp\left[-\frac{2 \times r^2(i,j)}{SBP^2 \times NBP^2}\right]$$



Local Fingerprinting

❑ Benefits:

- Content dependent
- Inter-independence (robust against local attacks)
- Resistant against a wide range of attacks
- Accurate spatial localization of key points
- Possible detection of local distortions
- Strong discriminative power

❑ Weaknesses

- Time and memory consuming
- Complexity
- Anti-collision not proved
- Invertibility not proved



Outline

- Introduction
- Watermarking
- Fingerprinting
 - Introduction
 - Perceptual hash functions
 - Robust content representation
 - ❖ Image Fingerprinting
 - ❖ **Video Fingerprinting**
 - Fingerprint Database
- Applications
- Conclusion & future work

Global Motion Based Video Fingerprinting (*)

- It is based on the direct parameter estimation of the global motion \mathbf{V} contained in MPEG stream.

$$\mathbf{v} = \begin{pmatrix} z & -r \\ r & z \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

z : zoom factor
 r : rotation factor
 t_x : pan or track
 t_y : tilt or boom

- For each Group Of Picture, a set of histograms accumulate motion parameters.
- The video signature is composed of 8 descriptors per GOP computed from histograms of the translation (t_x, t_y)

(*) R. Coudray and B.Besserer, "Global Motion estimation for MPEG-Encoded streams", IEEE ICIP, 2004



Global Motion Estimation

- First, a_1, a_2, a_3, a_4 are calculated:

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} = \begin{pmatrix} \nabla_x \mathbf{V}_x & \nabla_y \mathbf{V}_x \\ \nabla_x \mathbf{V}_y & \nabla_y \mathbf{V}_y \end{pmatrix} \quad \begin{array}{l} \nabla_x : \text{spatial derivation along } x \\ \nabla_y : \text{spatial derivation along } y \end{array}$$

- Then, the motion vectors are compensated with a_1, a_2, a_3, a_4 and (t_x, t_y) are calculated:

$$\begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} \mathbf{V}_x \\ \mathbf{V}_y \end{pmatrix}$$



Global Motion Descriptor

- 2 descriptors from the moment order 2 and 1 of the histogram H of (t_x, t_y)

$$M_{10} = \frac{\sum_x \sum_y H(x, y) x}{\sum_x \sum_y H(x, y)} \quad \text{and} \quad M_{01} = \frac{\sum_x \sum_y H(x, y) y}{\sum_x \sum_y H(x, y)}$$

- 2 descriptors from the percentage of the null motion in a given vector field
- 4 descriptors from the distribution of similar motion parameter from 4 regions segmented around the vector field.

Key Frame Based Video Fingerprinting (*)

□ Three steps process

1. Detection of video fragments, called scenes, shots
 - ❖ Scene cut selection
 - ❖ Each shot is represented by a “representative frame”, called stable frame.
2. Extraction of image features
 - ❖ Fingerprinting (Visual Hash/Local Fingerprint) of all stable frames
3. Extraction of video features
 - ❖ Video fingerprint = {stable frames' fingerprint}.

(*) A. Massoudi, F. Lefebvre, C.-H. Demarty, L. Oisel, B. Chupeau, “A Video Fingerprint based on Visual digest and Local Fingerprints”, IEEE ICIP, 2006



Shot Boundaries Detection

- An automatic process using two thresholds determines brutal transitions along the video and detects shot boundaries.

- Pseudo-global threshold

- ❖ $\tau_{\text{global}}(i, L_1) = \mu(i) + \alpha_1 \cdot \sigma(i)$

- $\mu(i)$ and $\sigma(i)$ denote the mean and the variance of $\|\text{Rash}(k) - \text{Rash}(k+1)\|_2$ measured for all k in $S_1 = [i-L_1; i+L_1]$.

- Adaptive threshold

- ❖ $\tau_{\text{local}}(i, L_2) = \alpha_2 \cdot d_{\text{max}}(i)$

- $d_{\text{max}}(i)$ is the second maximum value of $\|\text{Rash}(k) - \text{Rash}(k+1)\|_2$ measured for k in $S_2 = [i-L_2; i+L_2]$.

- The shot boundary, denoted SB , is

- $SB = i \mid \|\text{Rash}(i) - \text{Rash}(i+1)\|_2 > \max(\tau_{\text{global}}(i, L_1), \tau_{\text{local}}(i, L_2))$



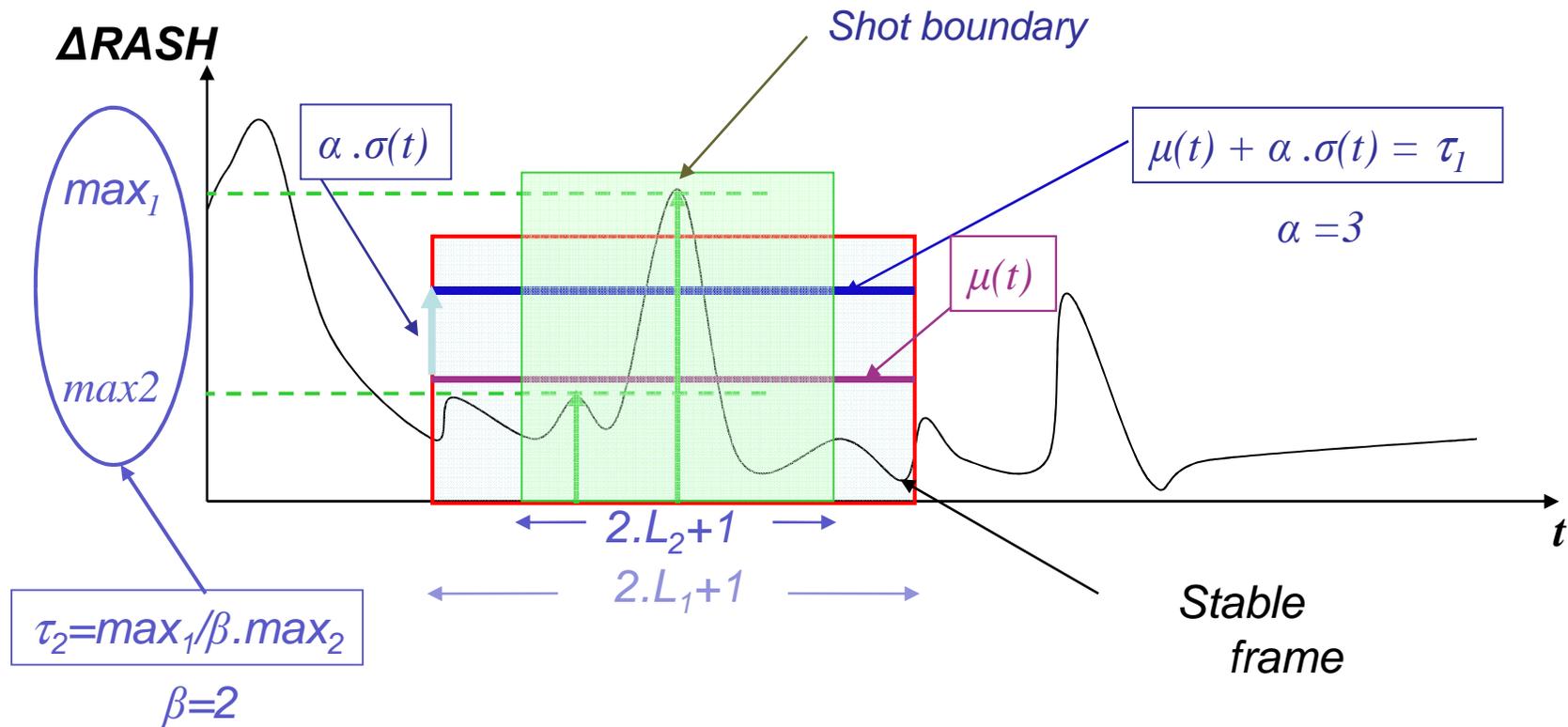
Stable Frame Detection

- A stable frame is the frame with the smallest content variation along a shot.
- For such a frame, the perceptual distance between this frame and the other neighbour frames will be very small

$$I^* = I \mid \left(\text{Dist}(I) = \min_{i \in S} \{ \text{Dist}(i) \} \text{ and } |\text{Entropy}(\text{RASH}(I))| \geq \tau \right)$$

$$\text{Dist}(i) = \frac{1}{2L_3 + 1} \sum_{\substack{j \in S \\ j \neq i \\ j=i-L_3 \\ \dots \\ j=i+L_3}} \|\text{RASH}(i) - \text{RASH}(j)\|_2$$

Stable Frame Detection



Shot boundaries = peaks with $\begin{cases} \Delta RASH(t) > \tau_1 & \text{Pseudo - global} \\ \Delta RASH(t) > \tau_2 & \text{Adaptive} \end{cases}$



Key Frame Based Video Fingerprinting

❑ Extraction of image features

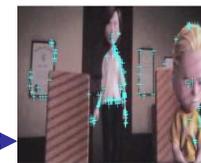
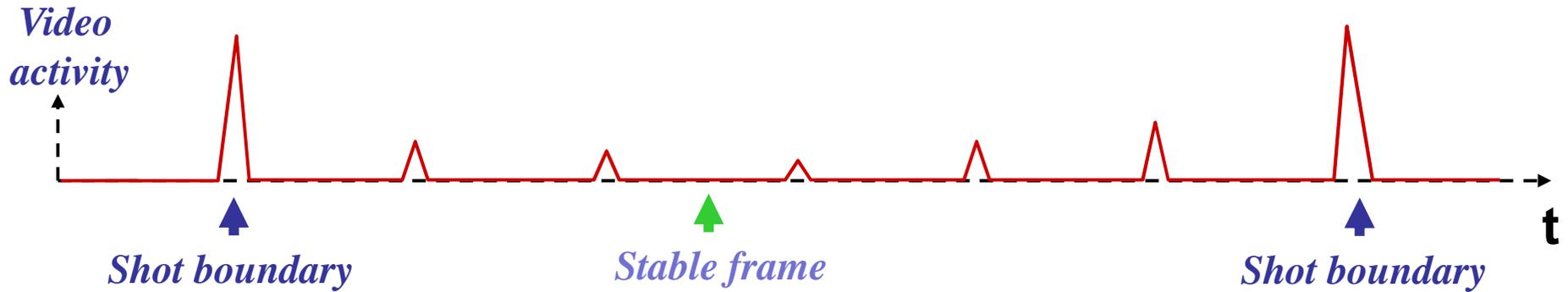
- Fingerprinting (Visual Hash/Local Fingerprint) of all stable frames

❑ Extraction of video features

- Video fingerprint = {stable frames' fingerprint}.



Video Fingerprint Generation

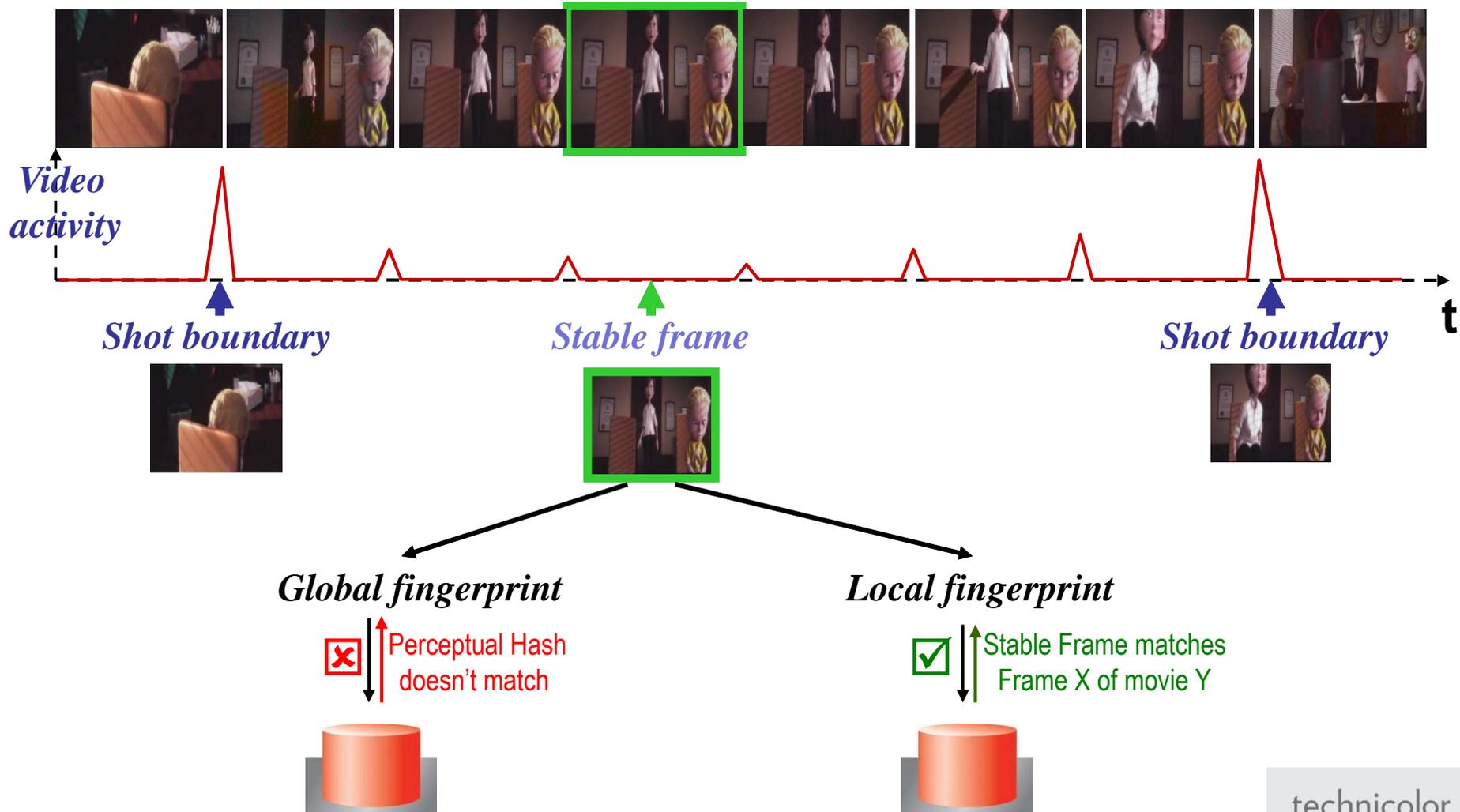


Local fingerprint

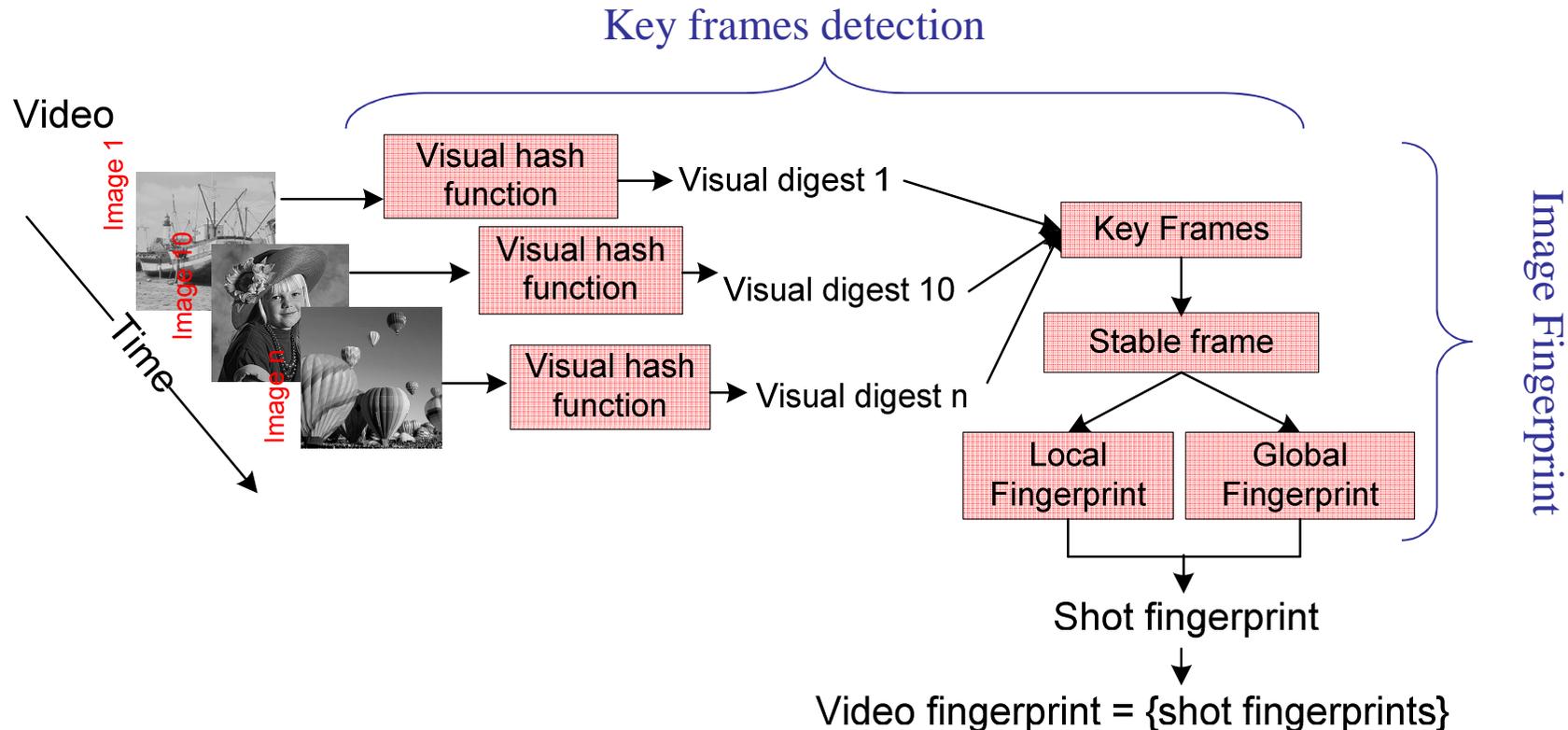
Global fingerprint



Video Fingerprint Detection



Key Frame Based Video Fingerprinting



- ❑ <1% of frames are fingerprinted
- ❑ For a full Perceptual Video hash process, the video fingerprint size < 210KB (movie=100min)



Performances Assessment

❑ Size of the database (hours):

- The larger the database, the higher the false positive and false negative rates.
- Database size has also usually an impact on detection speed.

❑ Definition of the attack(s):

- Camcorder, spatial stretching, frame rate changes, transcoding, compression...
- The more complex the attack is e.g. camcorder, the more difficult it is to correctly identify a copy.

❑ Duration of the candidate(s):

- The shorter the candidate, the more difficult the detection and the more false negatives.

❑ Detection speed:

- Fast detection reduces the number of required machines and allows live events filtering application.



Outline

- ❑ Introduction
- ❑ Watermarking
- ❑ Fingerprinting
 - Introduction
 - Perceptual hash functions
 - Robust content representation
 - **Fingerprint Database**
- ❑ Applications
- ❑ Conclusion & future work

Multimedia Database

❑ Objectives :

- To find near duplicate structures
- To organize the base of descriptors in order to optimize the tradeoff precision/recall/speed of query search
- To avoid/speed up the linear/exhaustive search

❑ Solutions :

- Return all elements in the database at a given distance ϵ from the query.
- Return the k nearest neighbors of the query in the database.
- Mono-dimensional vs. multi-dimensional.

❑ Warnings :

- Over a length-10 descriptor, the basic database suffers from the « dimension curse » (e.g: vanishing variance).



Outline

- ❑ Introduction
- ❑ Watermarking
- ❑ Fingerprinting
 - Introduction
 - Perceptual hash functions
 - Robust content representation
 - Fingerprint Database
 - ❖ Indexing strategies
 - ❖ Nearest neighbours search
- ❑ Applications
- ❑ Conclusion & future work

Mono-Dimensional Indexing

❑ It manages a point/vector (Point Access Method) or a more spatial complex structures (Spatial Access Method)

❑ The main techniques are:

➤ Hashing e.g.
$$h(\mathbf{k}) = \left(\left(\sum_i r_i a_i \right) \bmod P \right) \bmod m$$

with P a prime number, m an integer, a_i input, and r_i random value.

➤ B+ tree.

➤ Tf-idf (term-frequency inverse-document-frequency)

Multi-Dimensional Indexing

❑ Due to the “curse of dimensionality”, most of the data in the database populates a reduced space of the high dimension representation.

➤ Only this reduced space is indexed.

❑ Generic construction

➤ Partition/cluster the data (descriptors) in different cells.

❖ Using distance between descriptors (*K-means*)

❖ Using a partitioning of the high dimension space (*R-Tree, KD-tree*)

➤ The search is done in 2 steps

1. Identify the right cell

2. Find the best element inside the cell (linear search)

Evaluation

- ❑ Complexity / speed
- ❑ Standard metrics

$$\text{Recall} = \frac{\# \text{relevant_elements_in_the_returned_elements}}{\# \text{total_relevant_elements_in_the_database}}$$

$$\text{Precision} = \frac{\# \text{relevant_elements_in_the_returned_elements}}{\# \text{total_returned_elements}}$$

Outline

- ❑ Introduction
- ❑ Watermarking
- ❑ Fingerprinting
 - Introduction
 - Perceptual hash functions
 - Robust content representation
 - Fingerprint Database
 - ❖ Indexing strategies
 - ❖ Nearest neighbours search
- ❑ Applications
- ❑ Conclusion & future work

Locality Sensitive Hash Function

□ A hash function is said “locality sensitive” if 2 neighbour points have the same binary digest with a high probability while 2 distant points have the same digest with low probability.

□ Formal definition:

➤ A family functions $\mathcal{H}\{h:S\rightarrow\mathcal{U}\}$ is sensitive (r_1,r_2,p_1,p_2) with $r_1<r_2$ and $p_1>p_2$ if:

$$\forall \mathbf{p} \in \mathcal{B}(\mathbf{q}, r_1), \text{ then } Pr_{h \in \mathcal{H}} [h(\mathbf{q}) = h(\mathbf{p}) \geq p_1]$$

$$\forall \mathbf{p} \notin \mathcal{B}(\mathbf{q}, r_2), \text{ then } Pr_{h \in \mathcal{H}} [h(\mathbf{q}) = h(\mathbf{p}) \leq p_2]$$

where $\mathcal{B}(\mathbf{q}, r)$ is a ball, center \mathbf{q} and radius r .

Local Sensitive Hashing (LHS)

- ❑ Each descriptor \mathbf{p} (e.g. SIFT) is stored in l distinct hash tables \mathbf{g}_j .
- ❑ The output of each hash table \mathbf{g}_j has a dimension k .
- ❑ The tradeoff speed/precision is given by k and l (e.g. $l=550$ and $k=34$)
- ❑ \mathbf{g}_j functions are tuned with a couple of vectors \mathbf{D}_i and \mathbf{T}_i

$$\mathbf{D}_i = \langle \mathbf{D}_0^i, \mathbf{D}_1^i, \mathbf{D}_2^i, \dots, \mathbf{D}_{k-1}^i \rangle \quad \mathbf{T}_i = \langle \mathbf{T}_0^i, \mathbf{T}_1^i, \mathbf{T}_2^i, \dots, \mathbf{T}_{k-1}^i \rangle$$

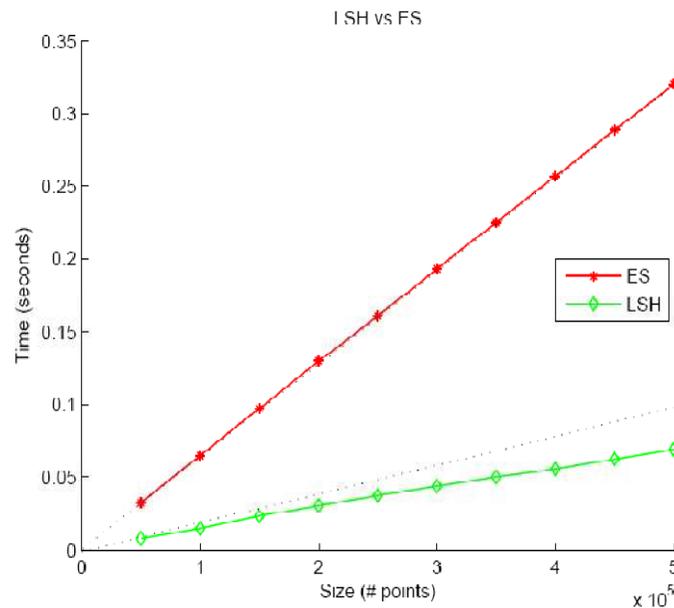
- ❑ The output \mathbf{b}_j of a descriptor \mathbf{p} is calculated as follows:

$$\mathbf{b}_j^i = \begin{cases} 0 & \text{if } (\mathbf{p})_{D_j^i} < \mathbf{T}_j^i \\ 1 & \text{otherwise} \end{cases}, \text{ with } j = 0..k-1, i = 0..l-1$$

Local Sensitive Hashing (LHS)

- ❑ $\langle \mathbf{b}_i^j \rangle, \dots, \langle \mathbf{b}_i^k \rangle$ are the has key (index) in the database.
- ❑ A linear search can be applied for all (potential) descriptors returned by the database engine.
- ❑ Some alternatives propose to hash $\langle \mathbf{b}_i^j \rangle, \dots, \langle \mathbf{b}_i^k \rangle$ in a single hash key.
- ❑ Short summary:
 - LSH is a the projection of descriptors along random lines
 - Speed/Precision are defined by the number of lines and number of segments in each line.
 - Indexes in the database are the projections

Exhaustive Search vs. LSH



| Size | ES (ms) | LSH (ms) | CPU Speedup |
|--------|---------|----------|-------------|
| 50000 | 32.3 | 7.6 | 4.246 |
| 100000 | 64.7 | 14.3 | 4.507 |
| 150000 | 96.9 | 22.8 | 4.244 |
| 200000 | 129.7 | 30.3 | 4.271 |
| 250000 | 160.7 | 37.3 | 4.306 |
| 300000 | 192.9 | 43.6 | 4.417 |
| 350000 | 224.8 | 49.8 | 4.508 |
| 400000 | 256.6 | 55.3 | 4.637 |
| 450000 | 288.4 | 62.4 | 4.617 |
| 500000 | 320.0 | 68.9 | 4.640 |

3.2GHz Intel Pentium 4, 2GB RAM, Linux kernel 2.6

- ❑ Benefit: LSH speeds up the exhaustive search.
- ❑ Weakness: LSH is RAM memory consuming.



New trends in database

- ❑ New trends in database are based on “Video-google” techniques.
- ❑ The main idea is to copy the text-retrieval model to video search.
- ❑ The main techniques :
 - Introduction to bag-of-words, bag-of-features (faster than RANSAC).
 - Use distance between descriptors (k -means) vs partitioning of the high dimension space.
 - Use inverse-document technique for the query.

Outline

- Introduction
- Watermarking
- Fingerprinting
- Applications
 - UGC Filtering
 - Pirate seat localization
- Conclusion & future work

Youtube statistics (*)

- ❑ Uploaded videos per day in March 2008: 200 000.
- ❑ The average video length: 2 minutes 46.17 seconds.
- ↳ 384 days of contents were uploaded every day in March 2008.
- ❑ Amateur contents (unambiguously user-generated): 80.3%.
- ❑ Professional contents: 14.7%.
- ❑ Commercial contents: 4.7%.
- ❑ Percentage of video probably in violation of copyright: 12%.

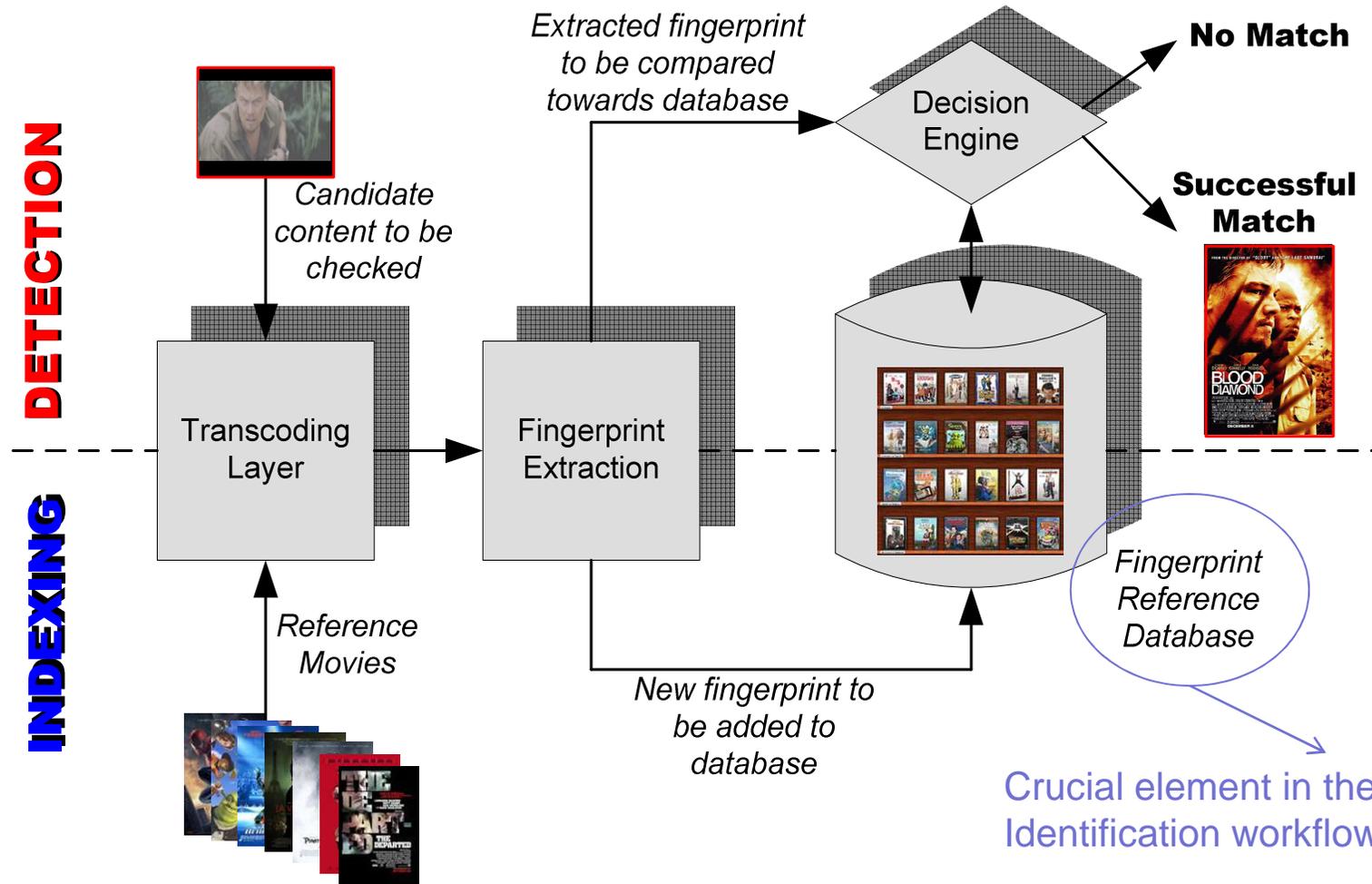
If we consider that some uploaded videos are removed immediately by YouTube, how many copyrighted contents are really uploaded every day?

↳ Thus UGC sites need methods to detect copyrighted content.

(*) <http://ksudigg.wetpaint.com/page/YouTube+Statistics?t=anon>



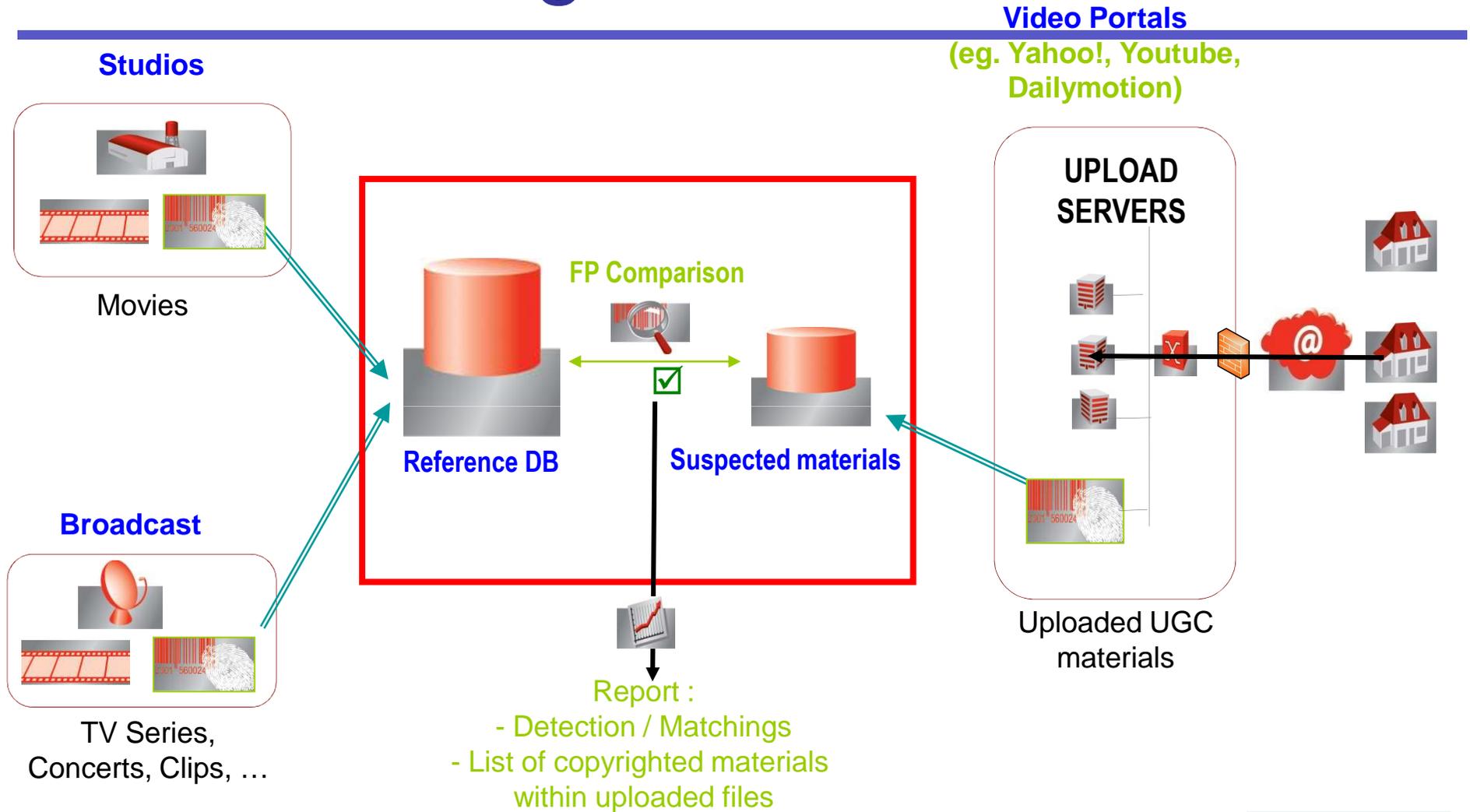
Content Identification Context



Crucial element in the Content Identification workflow



UGC Filtering



UGC filtering: conclusion

- ❑ Fingerprinting is mainly designed to identify contents.
- ❑ UGC filtering application requires
 - ❖ a fast detection module
 - ❖ 0 false positive
 - ❖ hit detection rate between 90 and 100%
 - ❖ scalability.
- ❑ We can not dissociate fingerprint generation from the fingerprint database.



Outline

- Introduction
- Watermarking
- Fingerprinting
- Applications
 - UGC Filtering
 - **Pirate seat localization**
- Conclusion & future work

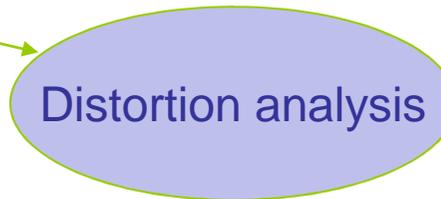
Problem statement



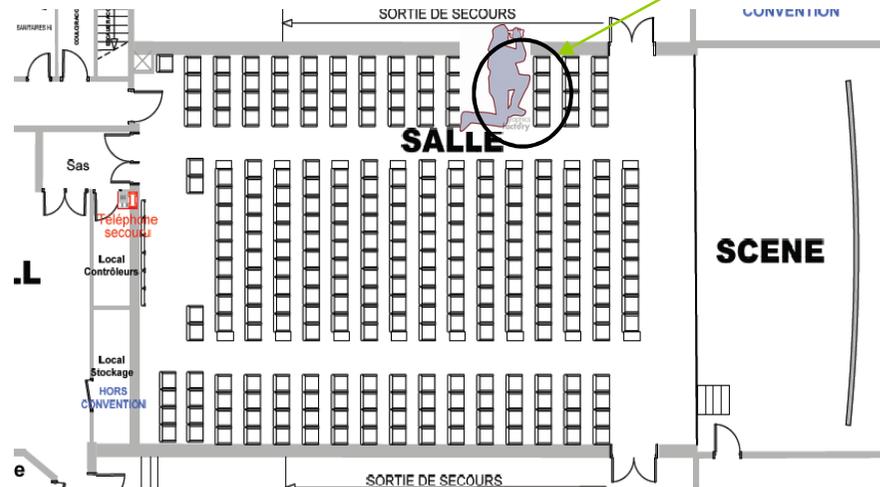
reference video



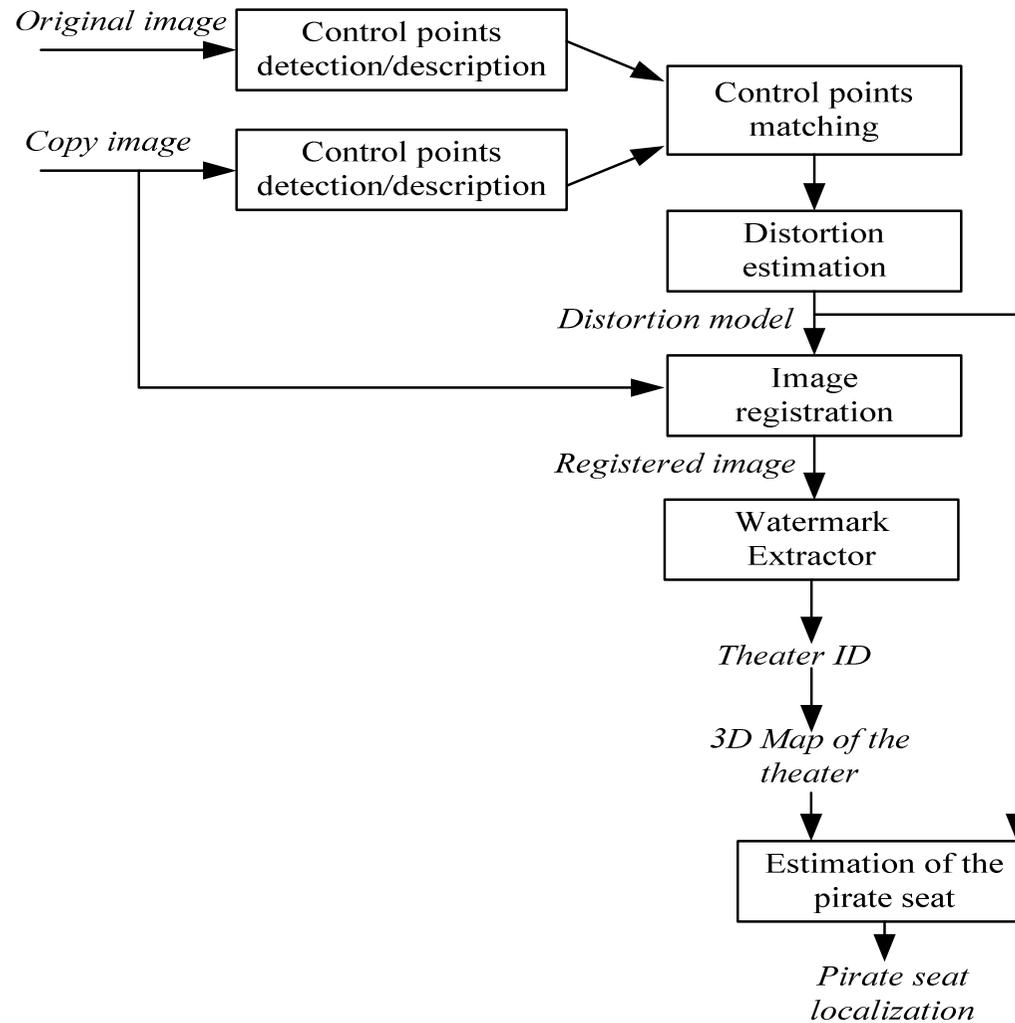
camcord copy



The pirate was here!



Pirate seat localization: investigation process (*)



(*) Chupeau, B., Massoudi, A. and Lefèbre, F, "In-theater piracy: Finding where the pirate was", SPIE'2008.

Temporally mapped original & copy frames



Original sequence (*) – frame #2760



Camcorder copy – frame #1459

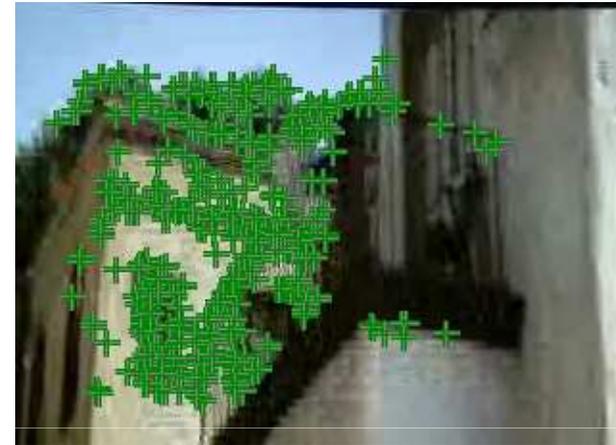
(*) ASC-DCI, Standard Evaluation Material (StEM), <http://www.theworx-digital.com/stem.html>



Detected control points



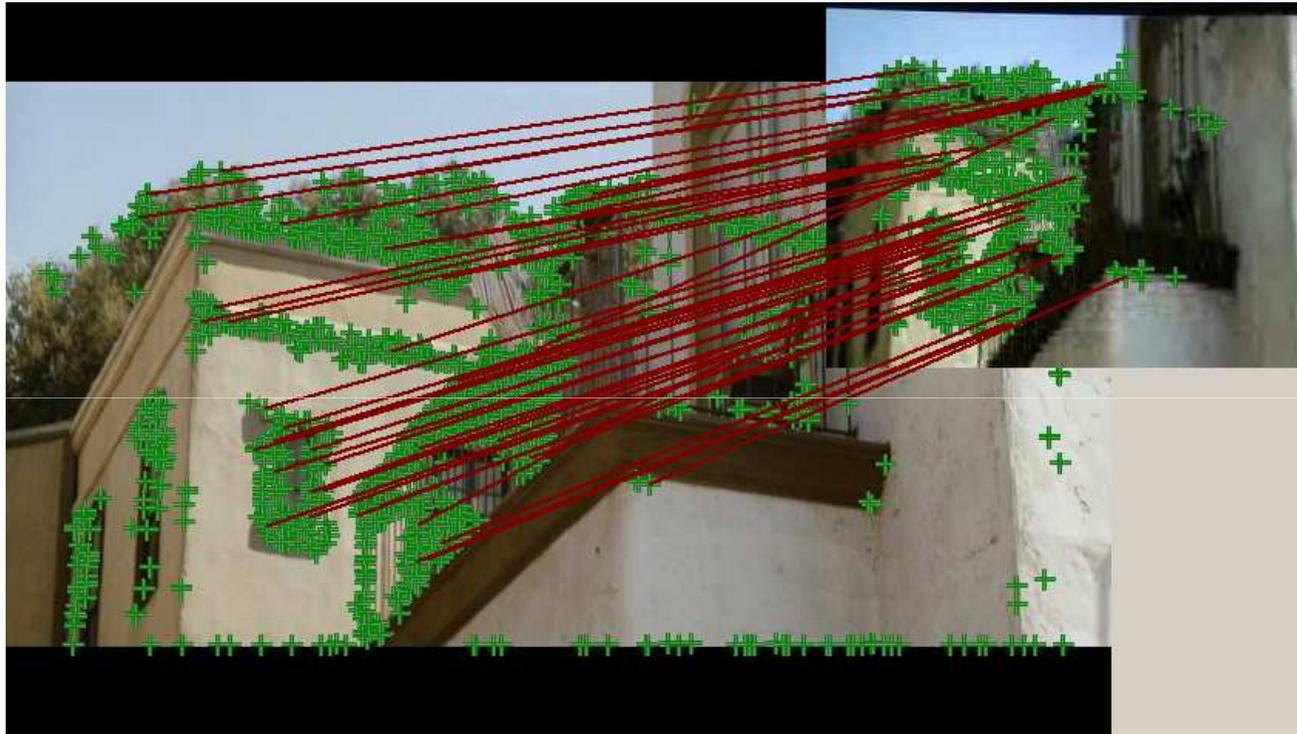
1734 control points in original frame



523 control points in copy frame



Matched control points



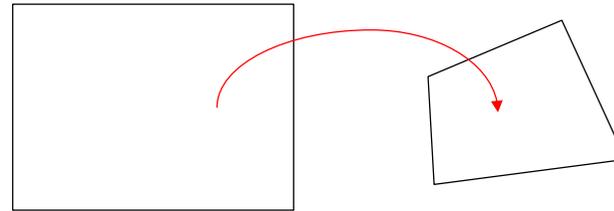
45 pairs of matched control points after filtering



Distortion model estimation

- ❑ 8-parameter homographic model
- ❑ Able to describe distortions due to camcorder capture

$$\begin{cases} x' = \frac{h_0x + h_1y + h_2}{h_6x + h_7y + 1} \\ y' = \frac{h_3x + h_4y + h_5}{h_6x + h_7y + 1} \end{cases}$$



Robust estimation method (least median squares)



Registration



Registered pirate frame with estimated homographic model:



Difference between original and registered copy frames



Results: Compensation of synthetic distortion



Original



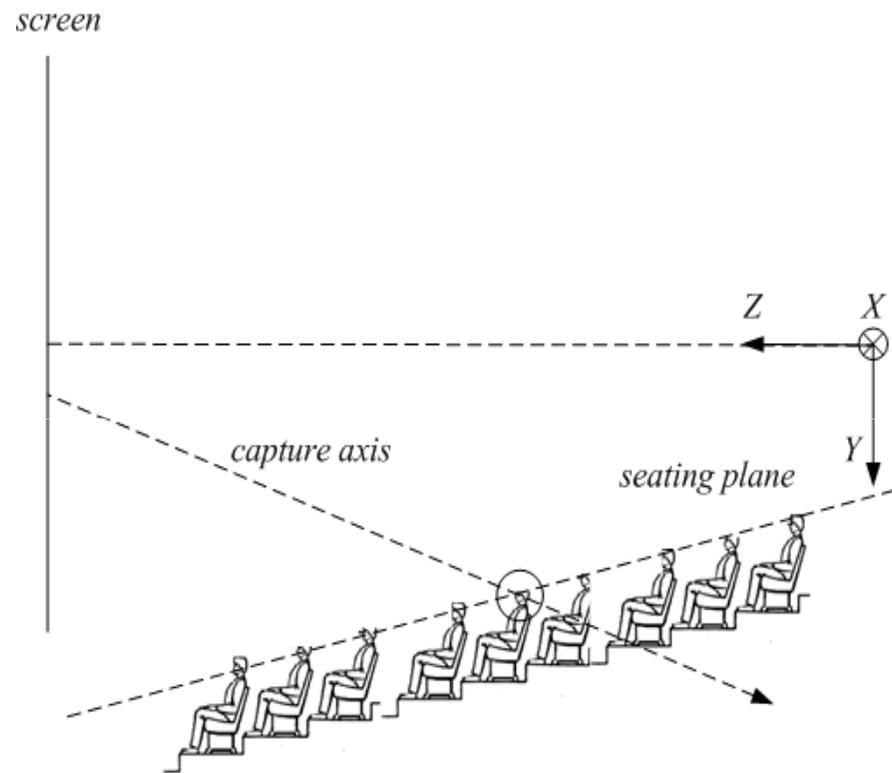
Synthetic distortion



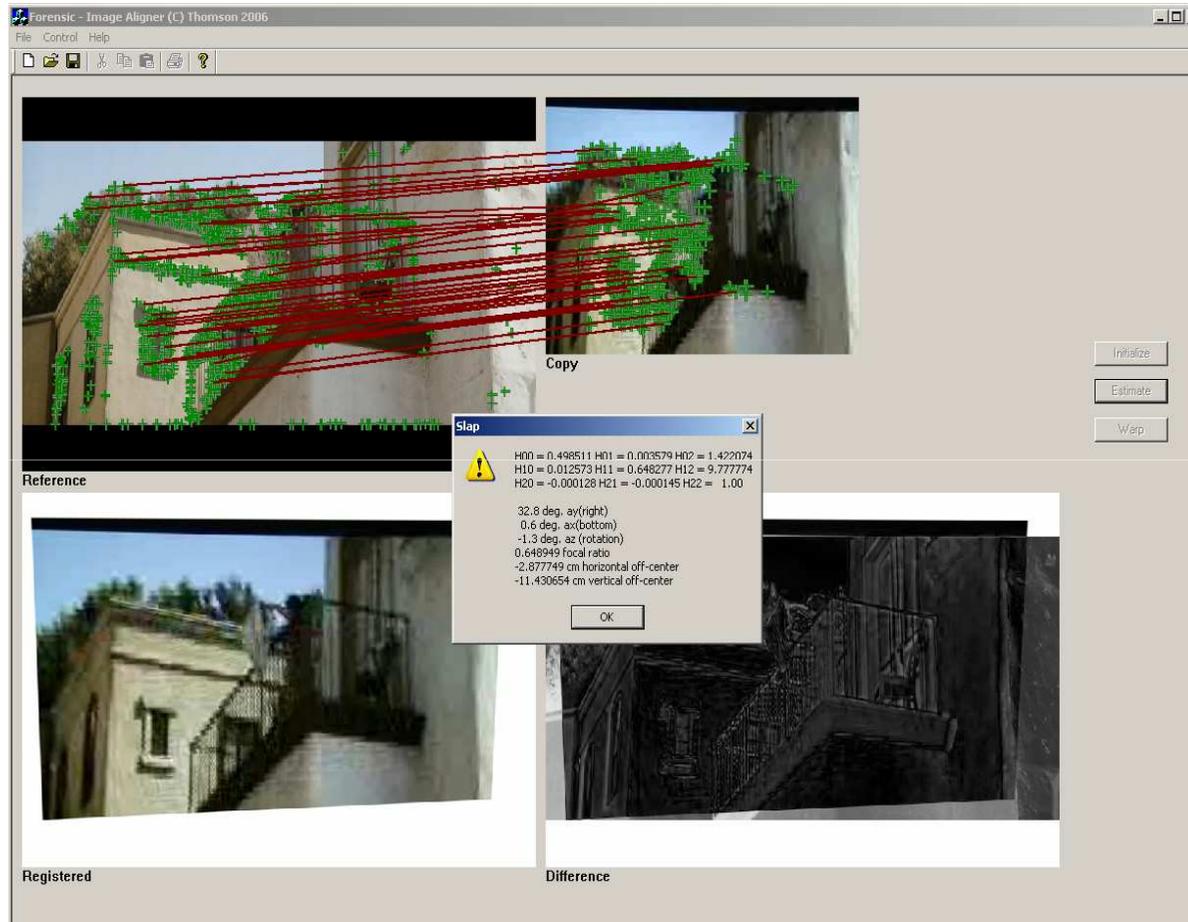
Registered



Estimation of the pirate seat: intersection with theater seating



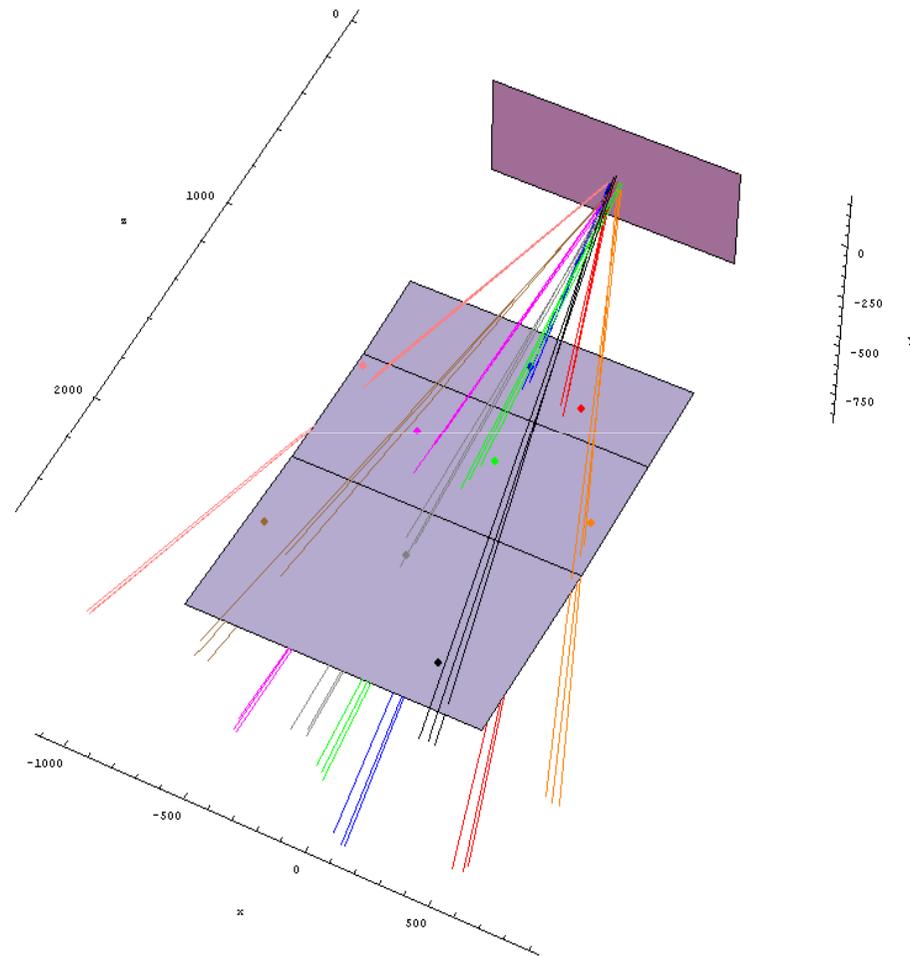
Estimation of the pirate seat: numerical estimation



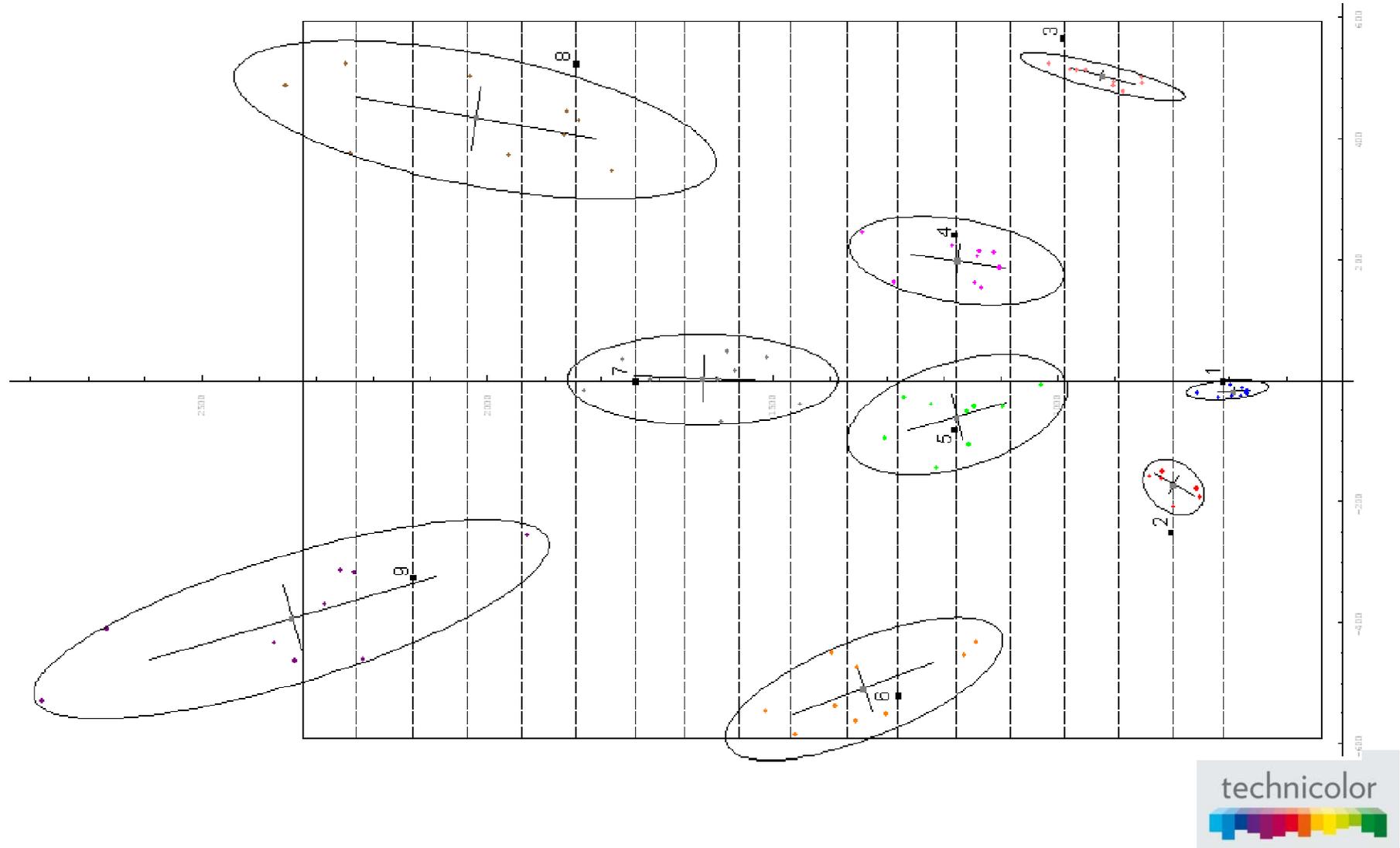
Screenshot of estimation software



Ground truth experiments



Results: top view of location estimates



Pirate seat localization: Conclusion

- ❑ Pirate localization from image distortion analysis is feasible, with acceptable accuracy
 - capture from projection booth vs. from seating area
 - divide seating area into several zones

Conclusion

- ❑ Fingerprinting
 - does not modify the content
 - enables robust identification of both watermarked and unwatermarked media content.
- ❑ Watermarking
 - Modifies the content
 - traces the origin of a leakage if the media is watermarked.
- ❑ UGC and peer-to-peer platforms come with new challenges for fingerprinting and watermarking technology, particularly robustness to strong distortions, collision-free, scalability, and detection speed.
- ❑ Fingerprinting, combined with watermarking, allows pirate seat localization.

