CS 161 – Watermarking

13 November 2006

How can we mark data?

We want to protect data:

- Video, sound, music (Digimarc, Intertrust, etc)
- Programs (Collberg, Thomborson)
- Statistical data

Examples of "traditional" protection methods:

- False entries in biographical dictionaries
- Copyright notices
- Licensing agreements
- Secure coprocessors

Watermarking

Watermarking:
- include low level bit data that marks information
- Either on a per-copy basis or a per-provider basis

Example: temperature database
- slightly adjust temps to mark uniquely

- Store copies of info released
  - If reused, prove using similarities
- But what if adversary changes low-level info?

Can watermarking work for data?

- It is not clear how applicable watermarking will be for data
- A perfect technique (immune against strong tampering) is probably impossible
- But some watermarking techniques may be usable
- DMCA: removing watermarks is illegal
- In this talk, I survey watermarking techniques for photographic data.

Motivation

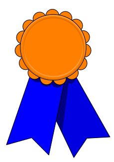
- Intellectual property is important for the Internet
- IP (images) are valuable
  - Costly to create high quality images
  - Users are attracted by good design
- Binary data is trivial to copy
- The web is a headache for copyright protection
- Many methods for free data exchange

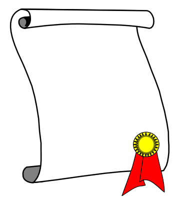
- Watermarking is seen as the white knight of copyright protection

Part 1: Making Image Watermarks

Secrets of a image watermarking salesman

- **This slide can transform you into an experienced watermarking salesman**
- Show two identical images
- Claim that one is watermarked
- Assert that it's robust against attacks
- Get signature on big $ contract





Companies to apply to

- Digimarc
- Bluespike
- MediaSec
- Signafy
- Signum (signumtech.com)
- ARIS (musicode.com)
- Intertrust
- But also some of the 2-3 letter companies
  - IBM
  - HP
  - NEC

Applications

- Copyright protection
  - Content owner embeds a secret watermark
  - Proof of ownership by disclosing the secret key
- Fingerprinting
  - Embed a serial number describing the recipient
  - Later we can detect which user copied the image
- Authentication
- Integrity verification
  - A fragile watermark assures integrity
- Content labeling
- Rights management
- Content protection

Watermarking concepts

Watermarks → visible / invisible

Verifiability → private / public

Watermarking technique → fragile / robust

Key → symmetric / asymmetric

Original necessary → yes / no

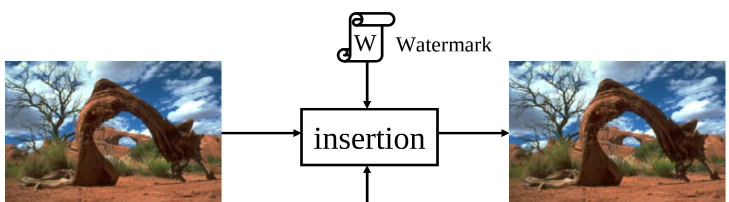
Visible watermarks

- Visible watermarks are used in special domains
  - Vatican library
  - Sw

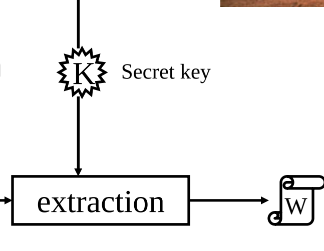
- Issues with visible watermarks
  - Content producer does not like to degrade the image
  - Customers don't appreciate them either
  - Visible watermarks are easier to remove
  - Easy to detect for people
  - But more difficult to detect automatically

The watermarking process (private wm)

- Embed a watermark

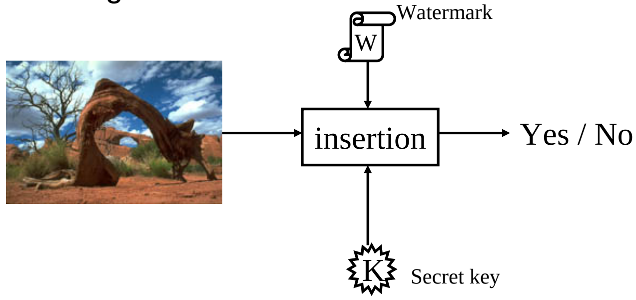


- Watermark extraction



Watermarking process II

- Detecting a watermark



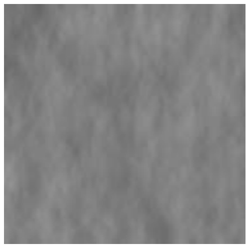
Requirements of invisible watermarks

- Robust against tampering (un- & intentional)
  - Various image transformations (RST)
  - Image compression
  - Color requantization
  - Non-linear transformations (print and scan)
- Non-perceptible, hard to detect
- Easy to use, exportable, etc.

- How can watermarking be possible?
  - The visual system has very strong "error correction"
  - An images contains a lot of redundancies
  - Small changes are undetected
  - People are used to low image quality (TV, newspaper images)

Example: The NEC watermark

- There is no perceptible difference between the original and watermarked image
- But the difference image looks interesting
- The watermark is present everywhere!


Early aproaches: Spatial Domain Embedding

- Original idea: LSB is insignificant
- JK-PGS (Jordan-Kutter pretty good signature)
  - The watermark was embedded directly in the LSB of the pixels of the blue plane in the spatial domain
  - For robustness, every possibility of rotation, translation, scale was searched
- Flaws
  - Blue plane is insignificant
  - Least significant bits are unimportant
  - Possible search space is huge
  - Not secure against, say, compression
- Tirkel, van Schnydel, and Osborne scheme
  - Embed m-sequences in the LSB of the spatial domain
  - But also not robust against tampering

Spatial Domain Embedding II

- Bender '95, Nikolaidis and Pitas '96
  - Randomly divide image into disjunct pixel set A and B
    - For most images, statistically,

$$\sum_{A\_pixels} pixel - \sum_{B\_pixels} pixel \approx 0$$

  - Insertion:
    - choose k small
    - A pixels: add k
    - B pixels: subtract k
    - Merge A and B to get watermarked image
  - Detection:
    - divide image again into A and B set

$$x = \sum_{A\_pixels} pixel - \sum_{B\_pixels} pixel$$

    - if x close to 0, then no watermark is present
    - if x close to N*k, then a watermark is present

Transformation Domain Encoding

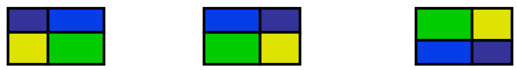
- An early goal was robustness against JPG compression
- Hence, design watermarking into JPG compression
- New ideas
  - Use strong error correction
  - Spread-spectrum encoding
  - Embed the mark in the perceptually important regions
  - Tradeoff robustness vs degradation (artifacts)
- Robustness against RST is essential
  - O'Ruanaidh uses Fourier-Mellin transform to achieve RST invariance
  - Reed-Solomon error correction
  - Spread-spectrum encoding
  - Strong error correction also gives JPG robustness
  - Does not need the original image for watermark extraction!

Signal Processing Primer

- The Fourier transform analyzes image frequencies

$$F(w) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} x_t \left(\cos\left(\frac{2\pi f}{N} t\right) + j \sin\left(\frac{2\pi f}{N} t\right)\right)$$

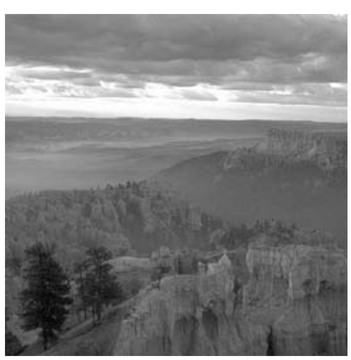
- Properties of the magnitude spectrum
  - Translation invariance

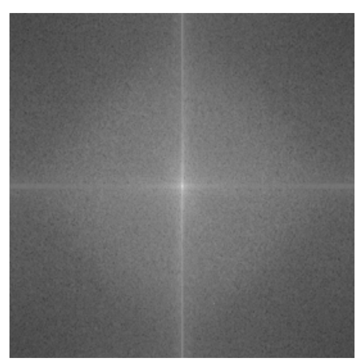


  - Rotation of the image translates to a rotation in the Fourier domain
  - Scaling results in “zoom in”
- The inverse Fourier transform returns the original image

Examples of the Fourier transform

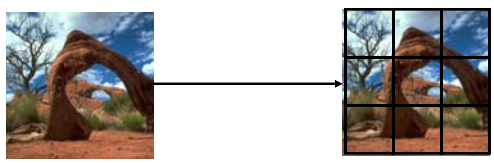
- Fourier transform of a photograph





Example: Robustness to cropping

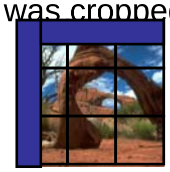
- Let's use the Fourier transform to construct a scheme which is robust against cropping
- Tile the image with small blocks of watermarks
  - For each block, we compute the Fourier transform
  - The watermark is embedded in the Fourier domain (each block)
  - Then we compute the inverse transform



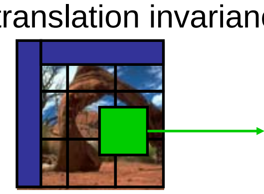
Each block is handled individually

Example: Robustness to cropping II

- The image was cropped



- On detection, any block will reveal the correct watermark - we win! (translation invariance)



Correct watermark

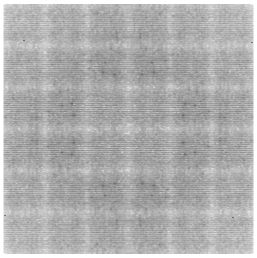
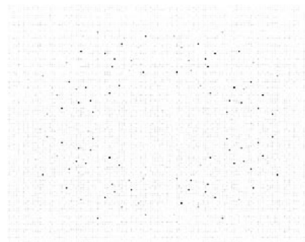
Part 2: Attacking Watermarks

Problems of Watermarking

- Copyright protection is big business - many attackers
- Internet spans continents and countries seamlessly
- Digital information is easy to copy
- Hackers are knowledgeable, creative, have lots of time, and are numerous
- Many attack opportunities
  - Few inventors, many attackers
  - Inventors despair after 3 years
- Human factors:
  - The default user does not understand watermarking
  - Human vision system is very robust to noise in images
  - Used to low quality in images (TV, strong JPEG compression)

How could we hope to attack?

- Detectable regularities let us believe that watermark removal is possible
- Example: Regularities of the FFT for Digimarc

- Empirical evidence has shown that schemes were not robust against tampering

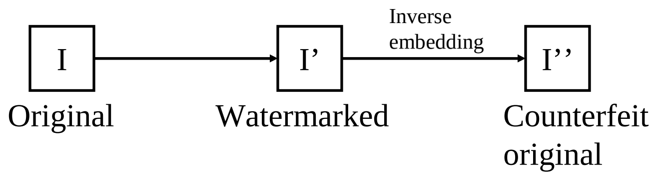
Attack classification

- System architecture failures
- Signal diminishment
- Image detector failure
- Court of law attacks

- Note: to illustrate the attacks, Alice is our content creator/owner, Bob is another “good” person and Mallory is the attacker.

System architecture failures I

- Protocol attack: Counterfeit original attack
  - Attack against schemes which use the original image



- Human factor
  - Unfamiliarity with watermarking
- User interface
  - Having the watermarking tool built into the same program as the image manipulation tools is asking for trouble
  - Should shield the user from error

System architecture failures II

- Implementation weaknesses
  - Digimarc uses public watermarks for authentication of creator and copyright protection
  - Image creator id only has 2-key password
  - Very easy to blackmail another user
  - Debugging tools to change software behavior
- Web crawler limitations
  - Refuse connection to crawlers
  - Spoofing, logins, payments
  - All the image detector failures we will discuss later

Signal diminishment

- Adding noise
- Lossy compression
- Image averaging, powerful against fingerprinting

- Users are usually happy with a low quality level (Jpeg, TV)

Watermark detector failure

- Most of these attacks prevent the watermark detector to synchronize with the watermark
- Jitter attack
- Distortion attack (StirMark) which simulates printing/rescanning
- Bandwidth limitation (mosaic attack)
  - Watermarking cannot handle small images
  - Split images in small pieces (e.g. 100×100)
- Java applets/ActiveX controls
  - Image displayed with Java applet automatically or after certain actions of the user
  - Can even be de-scrambled ‘on the fly’
- Unanticipated collisions

Early experiment – jitter attack

SysCoP

```
skytale:SysCoP$ imageread_demo syscop_watermarked.ppm
Key:

No certificate file.
--------------------------------------------------
A valid watermark found - estimated correction
percentage is : 100
Retrieved Secret Label (string) : SysCoP(TM)
```

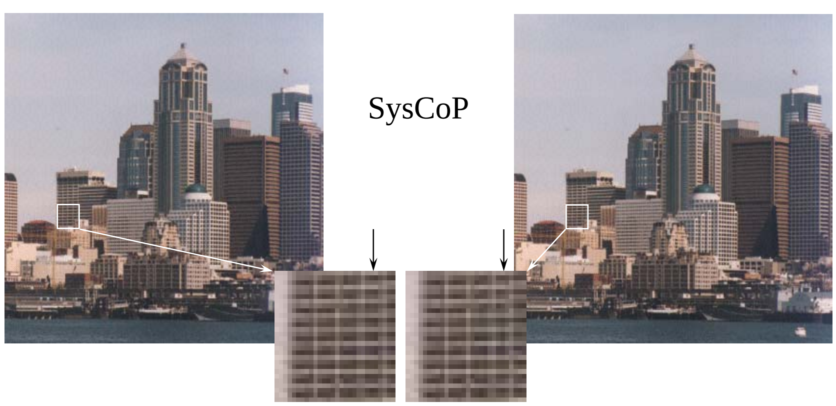
```
skytale:SysCoP$ imageread_demo syscop_jitter.ppm
Key:

No certificate file.
--------------------------------------------------
Cannon find valid watermark - failed.
Image syscop_jitter.ppm has been tampered or has
not been watermarked.
```


Jitter attack example

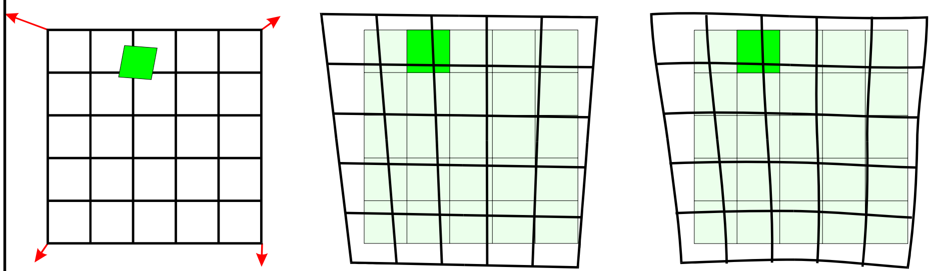
Watermarked

Jitter added

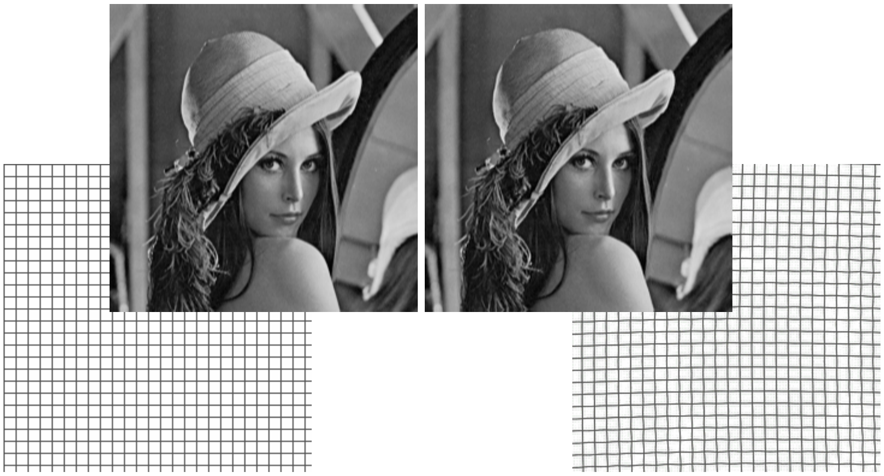


StirMark

- Apply minor geometric distortion
  - Stretching, shearing, shifting and rotation
  - Simulate printing/scanning process



StirMark example



PictureMarc 1.51, SysCoP (Demo), JK_PGS, SureSign, EIKONAmark (Pitas), NEC (Cox *et al.*)

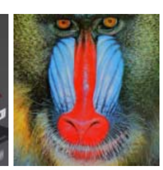
StirMark – Benchmark

- Images
  - *Glasses*, *Lena*, *Mandrill*, *Benz*, *Girl* (M. Kutter)
- Transformations
  - Scaling (0.5, 0.75, 0.9, 1.1, 1.5, 2)
  - Cropping (1%, 2%, 5%, 10%, 15%, 20%, 25%, 50%)
  - Rotation & cropping (–2°, –1°, –0.5°, 0.5°, 1°, 2°)
  - Rotation & scaling (–2°, –1°, –0.5°, 0.5°, 1°, 2°)
  - JPEG compression (90, 85, 80, 75, 60, 50, 25, 15, 10, 5)
  - Gaussian & median filter
  - StirMark's geometrical distortions

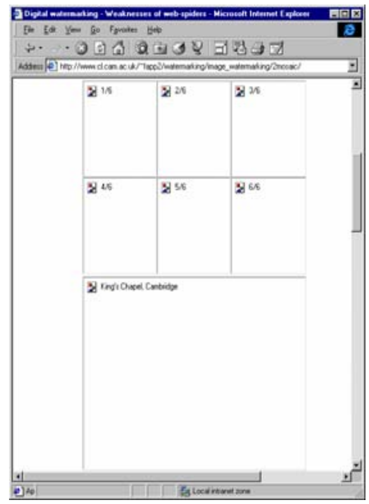
StirMark – Benchmark's results

	Digimarc 1.51	**SureSign 3.0 Demo**	**EikonaMark 3.01**	**JK_PGS 1.0 (Sun)**	**Giovanni 1.1.0.2**	**SysCoP 1.0R1**
GIF Conversion	20.00	20.00	20.00	20.00	12.00	16.00
Scaling	14.00	20.00	0.00	0.00	12.67	0.00
Cropping	20.00	20.00	0.00	8.00	3.00	0.00
Rotation & cropping	16.00	11.33	0.00	0.00	2.00	0.00
Rotation & scaling	16.67	12.00	0.00	0.67	2.00	0.00
JPEG	11.20	14.40	18.00	9.20	2.40	11.60
Filtering	20.00	20.00	20.00	20.00	12.00	16.00
Horizontal flip	20.00	20.00	0.00	0.00	0.00	0.00
StirMark 1.0	16.00	16.00	0.00	0.00	0.00	0.00
StirMark 2.2	0.00	0.00	0.00	0.00	0.00	0.00
	15.39	**15.37**	**5.80**	**5.79**	**4.61**	**4.36**



‘Mosaic’ attack



Legal attacks

- Server in another country “Internet is global but the law isn’t!”
- There are about 250 countries, 250 different laws
- General problem: web servers do not issue “receipts”
- Will law enforcement start to download content?

Can we apply to data?

- Non-problem:
  - Data does not always come in 2-dimensional form
  - Users may use a subset of data
- Problems:
  - Bit-rate for dispensing data
  - Averaging or modification of data
  - Retention of information to prove that data was taken.