Lecture-4

Information Hiding

- *Slides adapted from Spring 2011 offering of ENEE 408G and Fall 2013 offering of ENEE 631 in the ECE Department, University of Maryland, College Park by Profs. Min Wu (minwu@umd.edu) and Ray Liu (kjiliu@umd.edu)*

Drexel UNIVERSITY

Side Track: Binary Representation of Numbers

- Decimal representation (our everyday use)

$$11 = 1 \times 10 + 1$$

$$234 = 2 \times 10^2 + 3 \times 10 + 4$$

  - ✓ "Base 10": 10 choices of number per digit (0 ~ 9)
  - ✓ When counting exceeds 9, move up to next digit

- Binary representation (by computer/electronics)

$$(11)_{10} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2 + 1 = (1011)_2$$

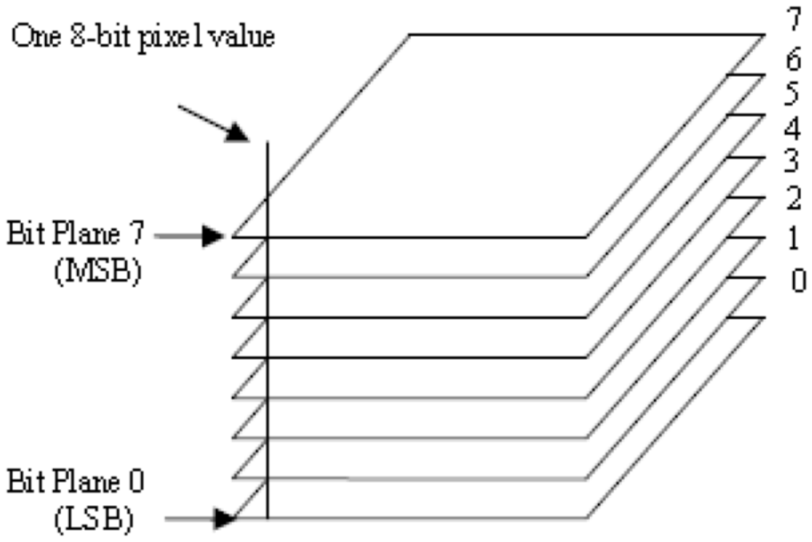
  - ✓ ONLY two choices of number per digit (0 and 1)
  - ✓ When counting exceeds 1, move up to next digit

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011

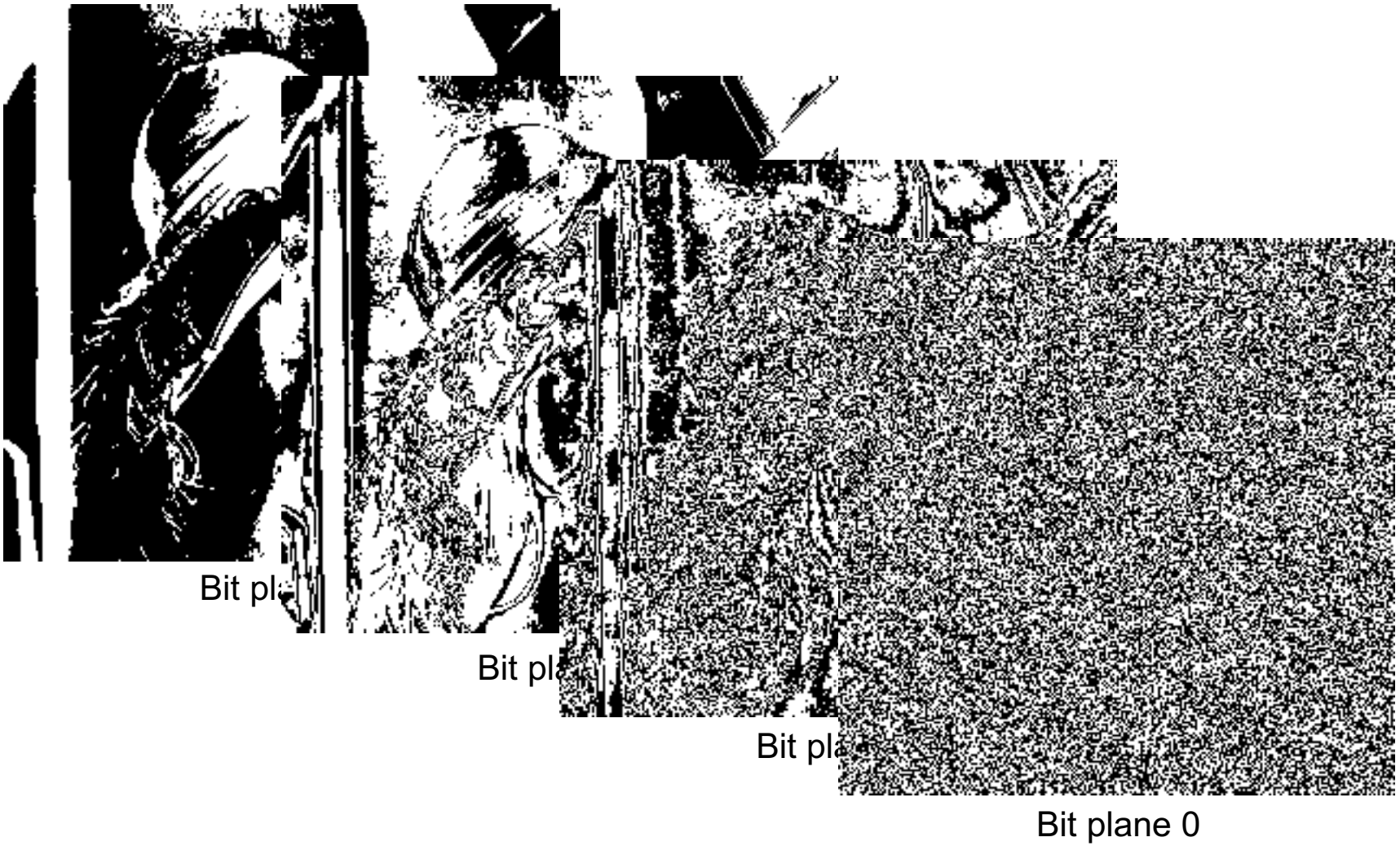
Binary Representation of Pixel Values

- A grayscale image with 256 = 2^8 shades of grays
  - Pixel values are 0 ~ 255 (darkest to brightest)
  - 8 bits can represent one pixel: 0000 0000 ~ 1111 1111

- View image as a stack of binary "bit planes"
  - Value of n^{th} bit plane equals n^{th} significant bit of pixel value
  - Equivalent to 8 binary images



Bit Plane Example



Changes to Pixel Values

- Most significant bit (MSB): look from the higher end of bits
  - Changing MSB of 8-bit pixel is equiv to $\pm 2^7 = 128$
  - Most perceptual significance

- Least significant bit (LSB): look from the lower end of bits
  - Changing LSB is equivalent to ± 1
  - Least perceptual significance

- Can exploit this to *hide information in images*

Example: Data Embedding by Replacing LSBs

Downloaded from `http://www.cl.cam.ac.uk/~fapp2/steganography/image_downgrading/`

Example: LSB Replacement (cont'd)

Replace LSB with Pentagon's MSB

Using Higher LSB Bitplanes for Embedding

……		
11	1011	
10	1010	
9	1001	
8	1000	← smaller distortion
7	0111	← original pixel value
6	0110	
5	0101	← replace 2nd LSB
4	0100	
……		

LSB embedding is very fragile

- may not survive mild lossy encoding

Issues to consider:

- *how much distortion by embedding?*
- *how much resilience to minor changes?*

Example: LSB Replacement of Higher Bitplanes



Replace 6 LSBs with Pentagon's 6 MSBs

Review: Pixel Depth

8 bits / pixel

4 bits / pixel

2 bits / pixel

- "Contour" artifacts for low pixel depth at gradual transition areas
- Human eyes distinguish about 50 gray levels => 5~6 bits/pixel

Drexel UNIVERSITY

Watermarking

- Insert hidden information into signal to achieve security goal
- Fragile watermarking
  - Watermark easily destroyed by any signal processing
  - Useful for tamper detection
- Semi-fragile watermarking
  - Watermark robust to certain signal processing operations (compression,
  - Useful for tamper detection
- Robust watermarking
  - Watermark difficult to distort or remove using any signal processing
  - Useful for identifying source of information, tracing info leaks, etc.

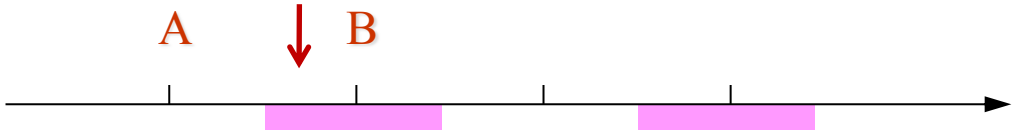
Tampering Detection by Pixel-domain Fragile Wmk

Downloaded from ICIP'97 CD-ROM paper by Yeung-Mintzer

Improve Robustness in Embedding

- Introduce quantization to embedding process
  - Make features being odd/even multiple of Q

even "0"
odd "1"



feature value	2kQ	(2k+1)Q	(2k+2)Q	(2k+3)Q
odd-even mapping	0	1	0	1

- Tradeoff between embedding distortion and robustness

  Larger Q => Higher resilience to minor changes

  => Higher average changes required to embed data

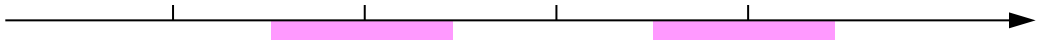
Recall: Embedding Basics – Two Simple Tries

Data Hiding: To put secondary data in host signal

(1) Replace LSB

(2) Round a pixel value to closest even or odd numbers

even "0"
odd "1"



pixel value	98	99	100	101
odd-even mapping	0	1	0	1

- *Both equivalent to reduce effective pixel depth for representing host image*
- *Detection scheme is same as LSB, but embedding brings less distortion in the quantized case, esp. for higher LSB bitplane*

\+ Simple embedding;

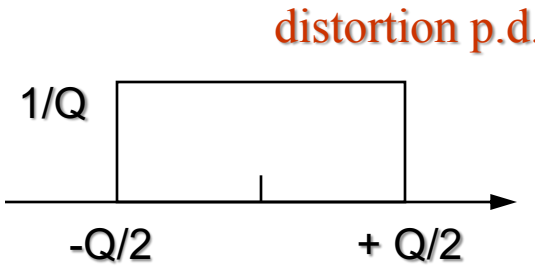
– Fragile to even minor changes

Distortion from Quantization-based Embedding

- Uniform quantization with step size Q

  (Assume source's distribution within each interval is approx. constant)

  $MSE = Q^2 / 12$

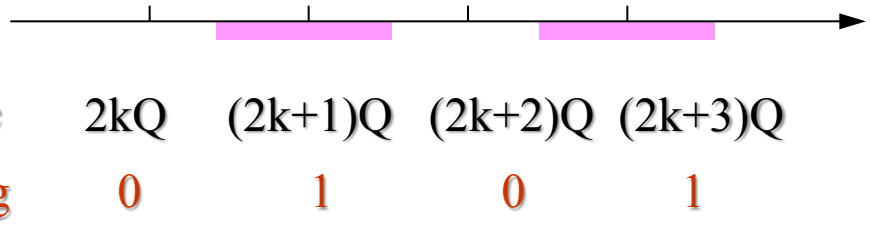


- Odd-even embedding with quantization

  $MSE = ½ * (Q^2/12) + ½ * (7Q^2/12)$

  $= Q^2 / 3$

  - *MSE equiv. to quantize with 2Q step size!*
  - *"Predistort" via quantization to gain resilience [-Q/2, Q/2]* (conditional)

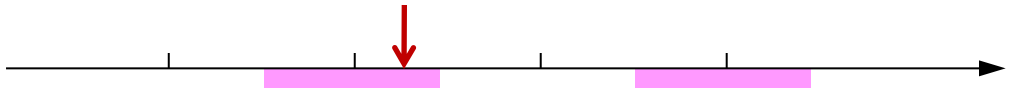


Security Problem

- Least significant bit plane changed to hidden information
- Very easy to discover hidden information by examining bit planes
  - Signal visually detectable
  - Common LSB amongst many images from the same source
- Simple watermark attack
  - Identify that LSB is watermarked
  - Extract & save LSB from watermarked image (save watermark)
  - Modify image
  - Re-insert watermark into tampered image

Fight Against Forging Tamper-Detection Watermark?

- If using LSB to embed a fragile watermark for tampering detection, adversary can alter image but retain LSB



feature value		2kQ	(2k+1)Q	(2k+2)Q	(2k+3)Q	
odd-even mapping		0	1	0	1	
lookup table mapping	…	0	1	1	0	…

[Solution 1] Add uncertainty to the embedding mapping

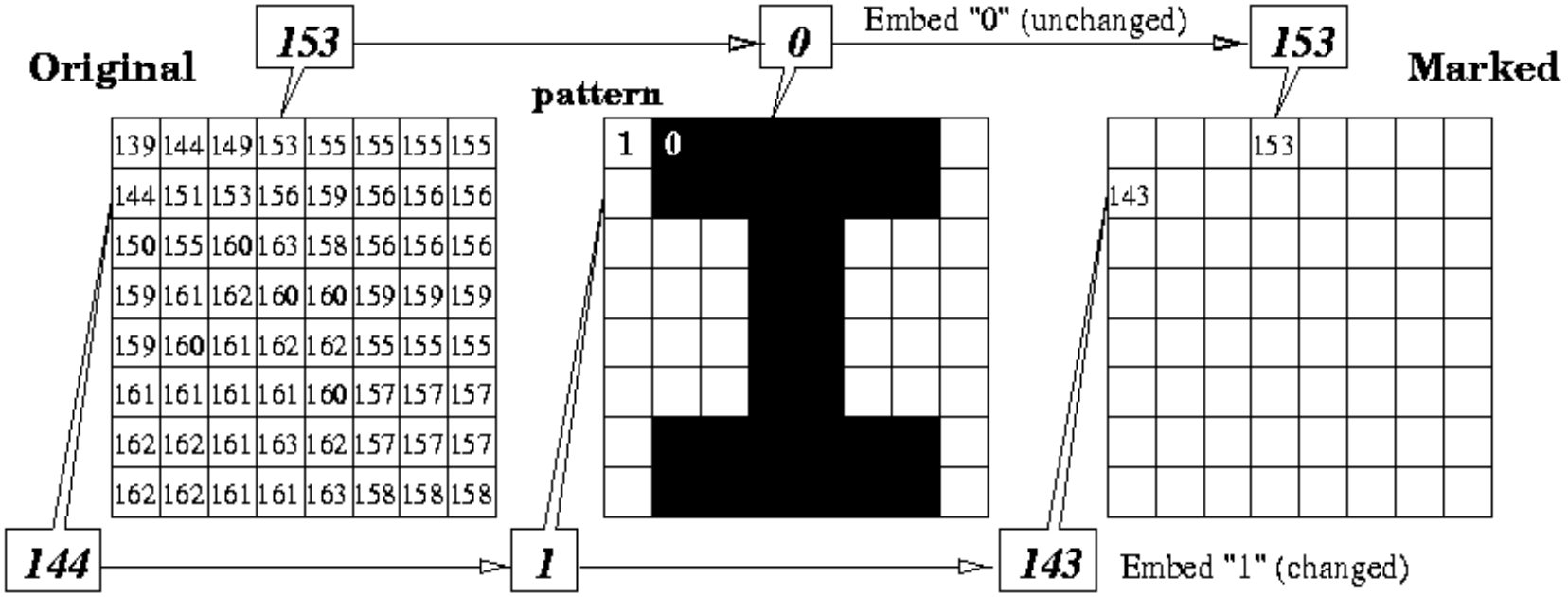
– through a random look-up table with controlled run length

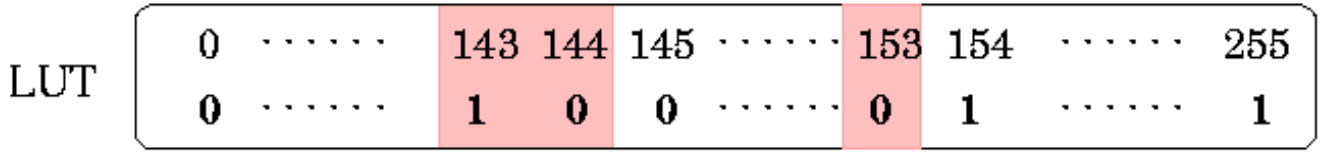
[Solution 2] Make watermark securely depend on host content

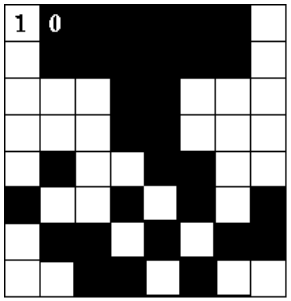
E.g. embed a robust/content-base hash of host image

wmk = randomized_function [host image]

Pixel-domain Table-lookup Embedding *(Yeung-Mintzer ICIP'97)*







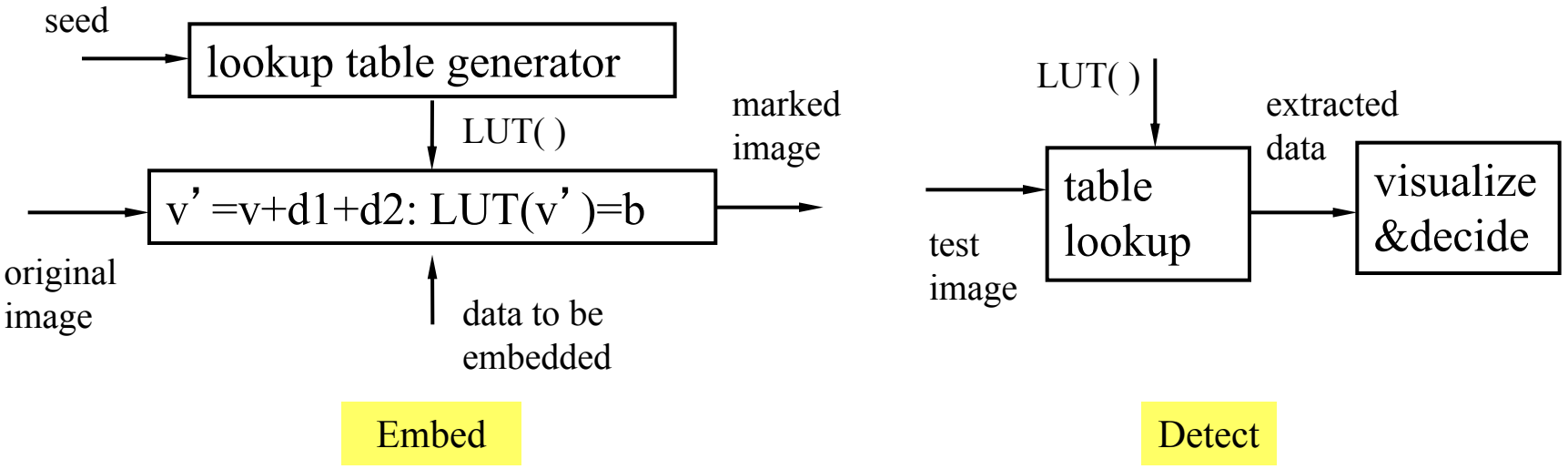
extracted wmk
from altered image

– Simple to implement; be able to localize alteration

Drexel UNIVERSITY

Yeung's Fragile Watermark for Tampering Detection

- Basic idea:
  - enforce certain relationship to embed data
  - minimize distortion: nearest neighbour, constrained runs
  - diffuse error incurred to surrounding pixels



d1: diffused error

LUT Embedding: Distortion/Security/Robustness

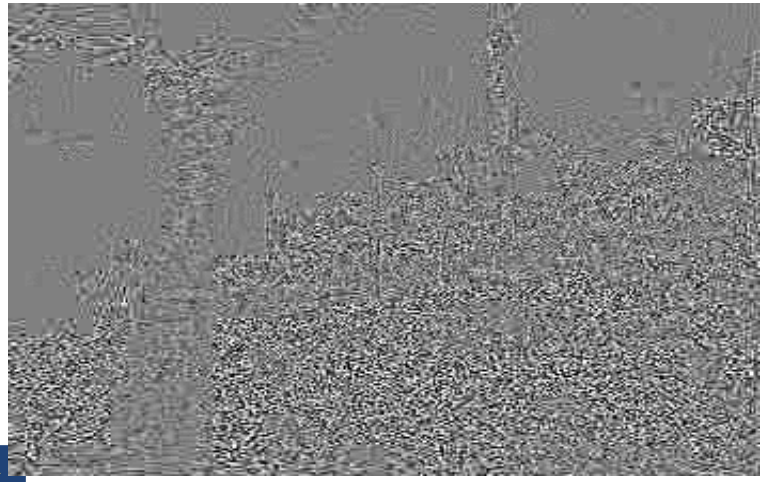
- What's new compared with odd-even embedding?
  - Mapping from feature to embedded bit is less predictable
  - Adjacent intervals may be mapped to the same bit value
- How much security gained with proprietary LUT? =>
  - Proprietary LUT brings uncertainty and makes it difficult for attackers to embed specific data at his/her will
- How much MSE introduced by embedding? =>
  - Larger than odd-even embedding
- How much resilience gained? =>
  - Moving away by Q/2 step may not trigger detection error
    - *Due to possible continuous run in LUT*

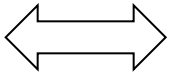
Ref: M. Wu: "Joint Security and Robustness Enhancement for Quantization Based Embedding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, no. 8, pp.831-841, August 2003. (see ICIP'03 for shorter conf. version)

From Fragile/Semi-Fragile to Robust Watermark

- Applications of fragile/semi-fragile watermark
  - Tampering detection
  - Secret communications => "Steganography" (covert writing)
  - Convey side info. seamlessly
- Situations demanding higher robustness
  - Protect ownership (copyright label), prevent leak (digital fingerprint)
  - Desired robustness against compression, filtering, etc.
- How to make it robust?
  - Use "quantization"; Use error correcting coding
  - Borrow theories from signal detection & telecommunications
    - ***"Spread Spectrum Watermark":*** *use "noise" as watermark and add it to the host signal for improved invisibility and robustness*

Example: Robust Watermark via "Noise"





10011010 …

© Copyright …

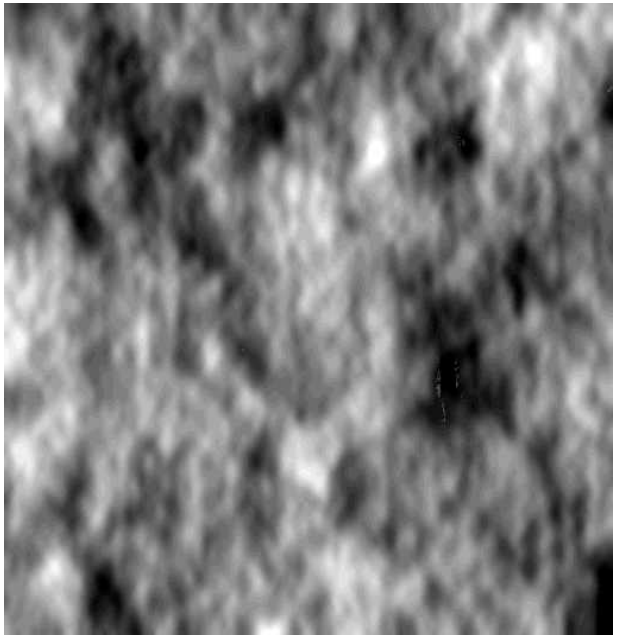
- *Embedding domain tailored to media characteristics & application requirement*

Drexel UNIVERSITY

Watermarking Example by Cox et al.

Original

Cox
whole image DCT
Embed in 1000 largest coeff.



Difference between marked & original