Digital Watermarking: An Introduction

Edward J. Delp

Purdue University
School of Electrical and Computer Engineering
Purdue Multimedia Testbed
Video and Image Processing Laboratory (*VIPER)*
West Lafayette, Indiana

email: ace@ecn.purdue.edu
http://www.ece.purdue.edu/~ace

Outline

- Provide an introduction to watermarking and data hiding and overview its use
- Describe how security techniques may/will impact multimedia systems

Multimedia Security

- “Everything” is digital these days - a copy of a digital media element is identical to the original
- How can an owner protect their content?
- Are images still “fossilized light”?
- What does all of this mean in terms of law?
- Does any security system really work or does it just make us feel good!

What Do We Want From a Security System?

- **Access Control**
- **Copy Control**

Þ

Playback Control

Record Control

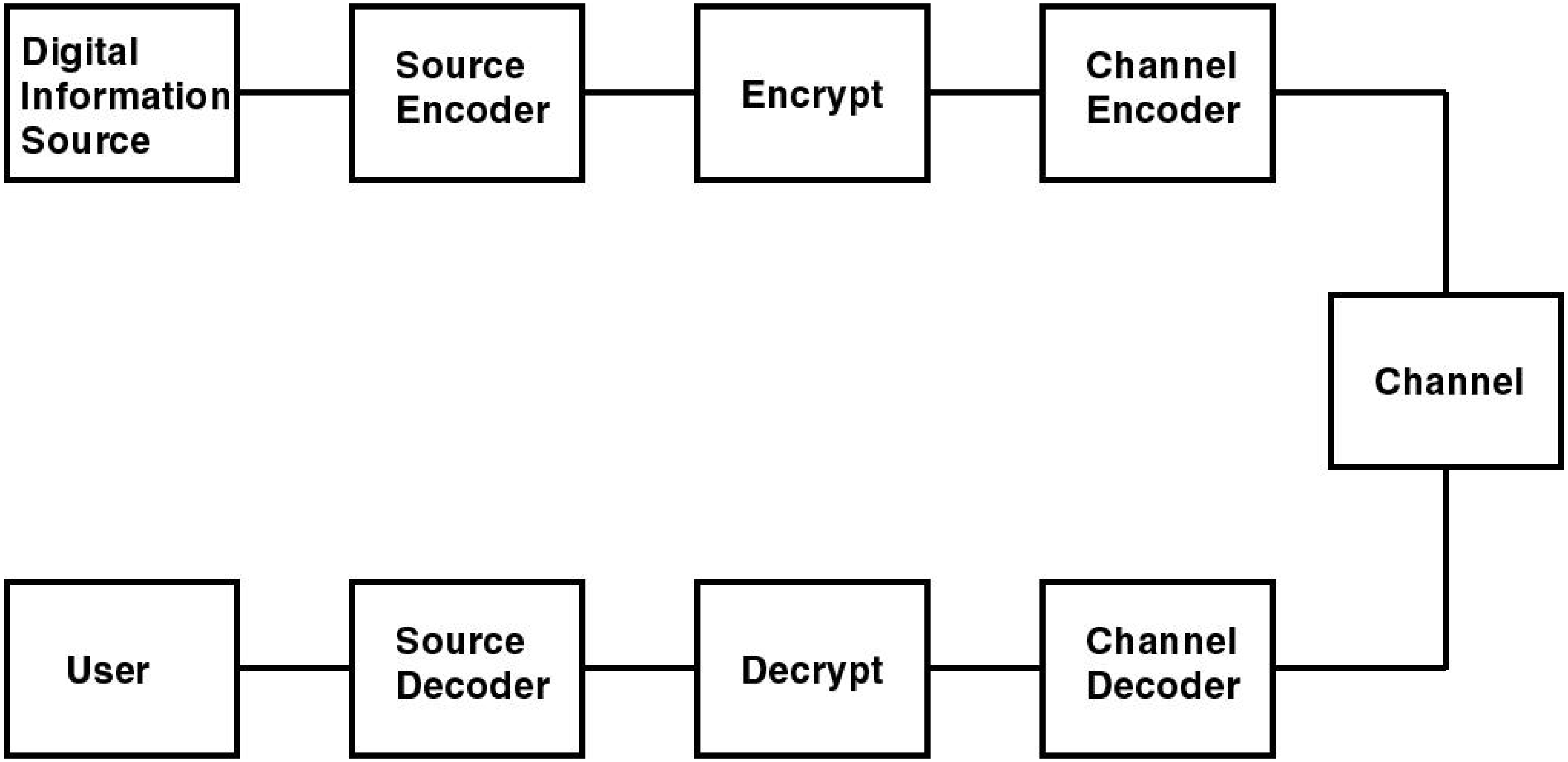
Generation Control

- **Auditing (fingerprinting)**
  - **Who did what and when?**

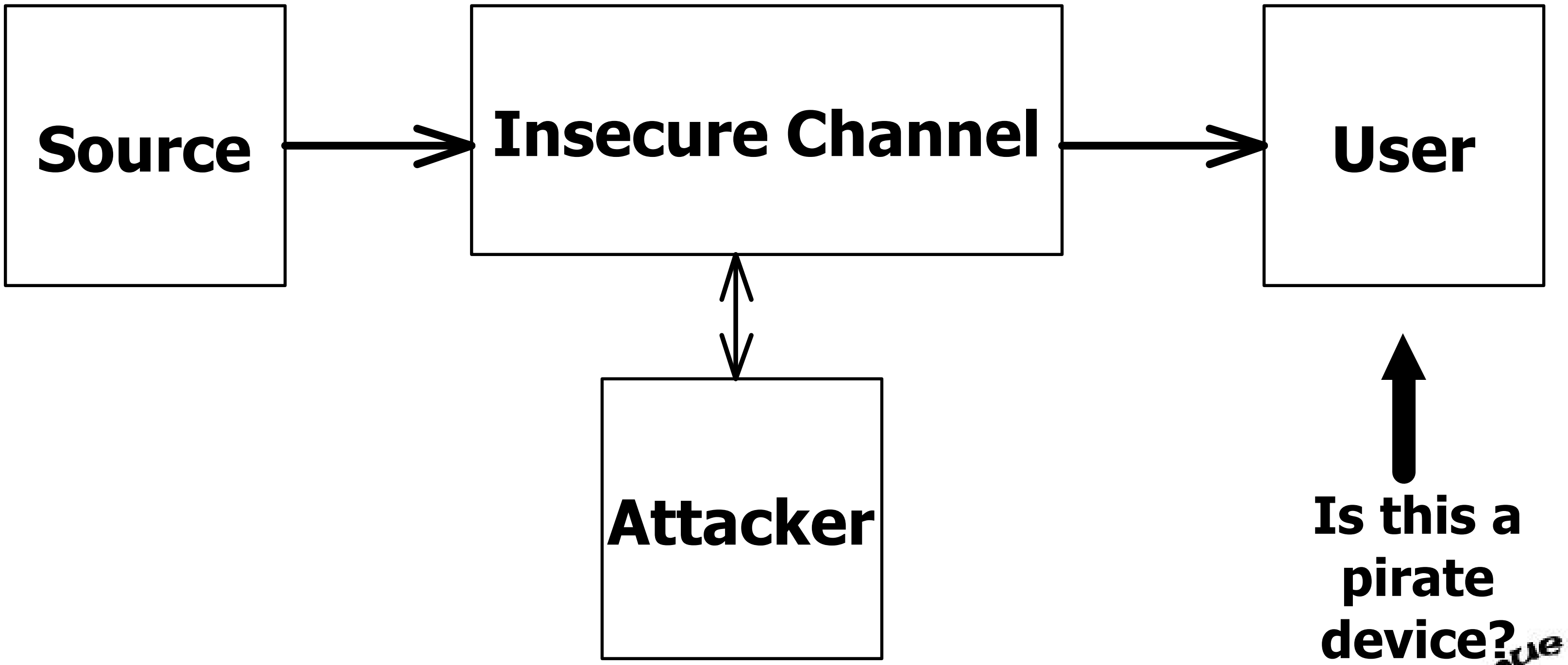
What Do Users Want?

- Time-shifting
- Space-shifting
- Single copy (back ups?)

Digital Communication System

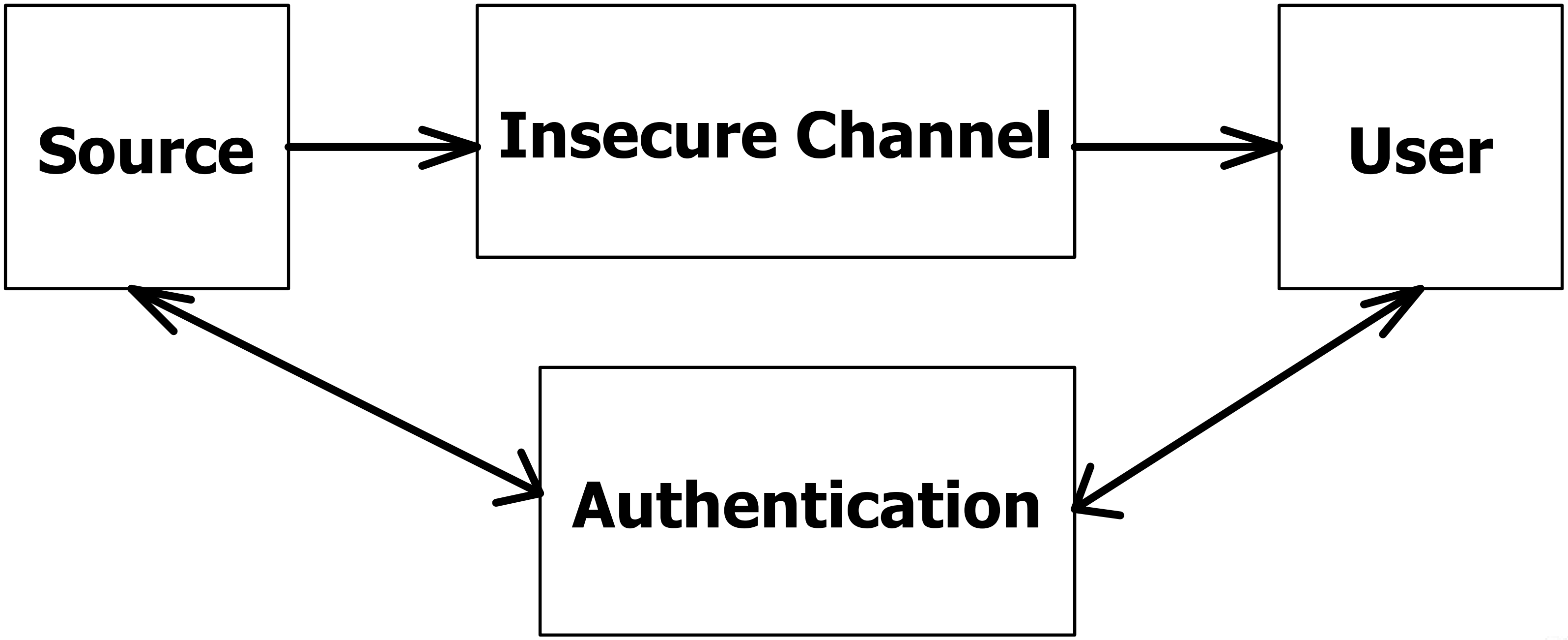


Typical Cryptography System: Trusted Users



Is this a pirate device?

Cryptography System: User Not Trusted



Media Elements

- Audio
- Video
- Documents (including HTML documents)
- Images
- Graphics
- Graphic or Scene Models
- Programs (executable code)

Multimedia Security - Tools Set

- Encryption
- Authentication
- Hashing
- Time-stamping
- Watermarking

Multimedia Security Applications

- **Privacy**
- **Forgery Detection Þ *watermarking***
- **Copyright Protection Þ *watermarking***
- **Proof of Purchase (non-deniable)**
- **Proof of Delivery (non-deniable)**
- **Intruder Detection**

What is Watermarking?

- The use of perceptually invisible authentication techniques
  - “controlled” distortion is introduced in a multimedia element
- Visible watermarks also exists

Watermarking Scenario

- Scenario
  - an owner places digital images on a network server and wants to “protect” the images
- Goals
  - verify the owner of a digital image
  - detect forgeries of an original image
  - identify illegal copies of the image
  - prevent unauthorized distribution

Where are Watermarks Used?

- **Watermarks have been used or proposed in:**
  - **digital cameras**
  - **DVD video**
  - **audio (SDMI)**
  - **broadcast video (in US - ATSC)**
    - **visible watermarks now used**
  - **“binding” mechanism**
  - **key distribution systems**
  - **preventing forgery of bank notes**

Usually as secondary security Þ conversion to “analog”

Steganography

Steganography - (*covered writing*) techniques used to hide information within other information to conceal the very existence of the message

Used much longer than cryptography

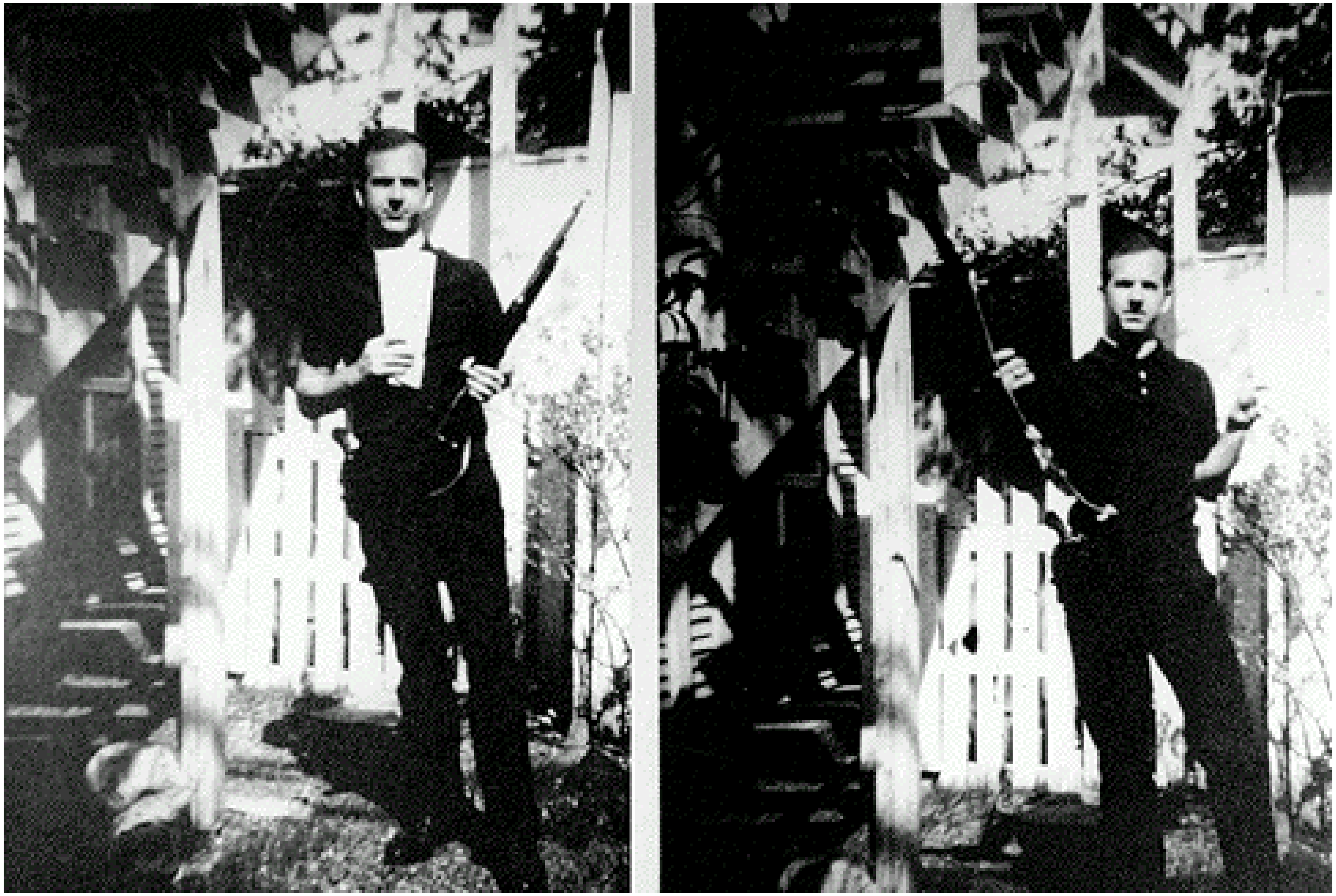
Different than crytography in that an illegal user may intercept the message

Why is Watermarking Important?

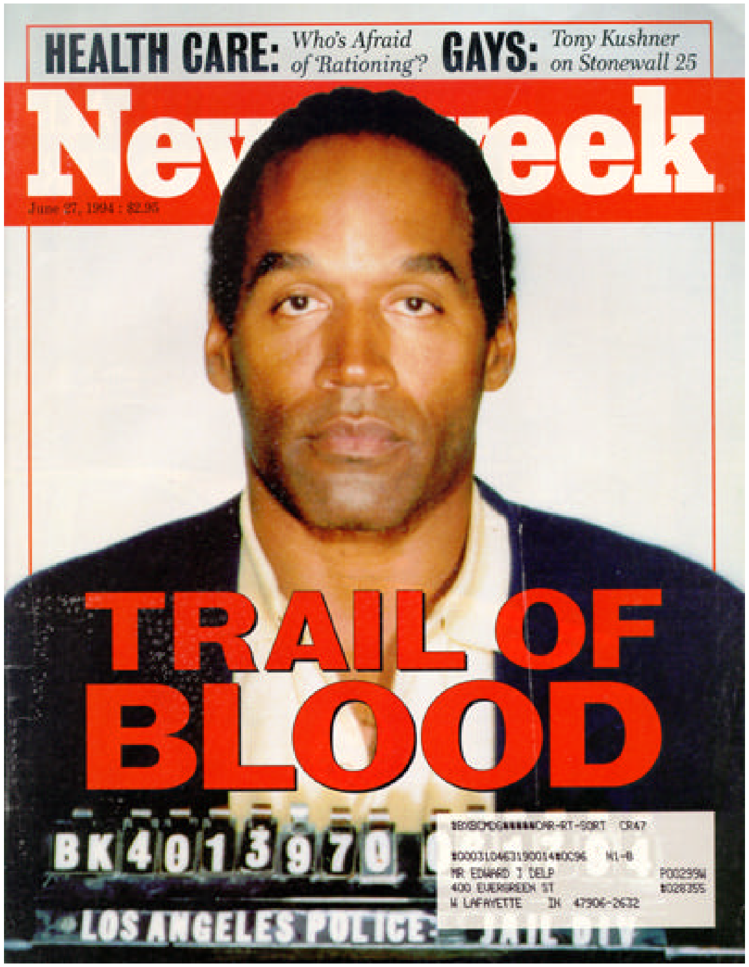
SPY
Nove
WHITE-HOT
POLITICS ISSUE
KENNEDY
The Torch Passes
PLUS:
THOSE SURPRISING GORE GIRLS!
Daryl Hannah as
Mrs. John F. Kennedy Jr.?

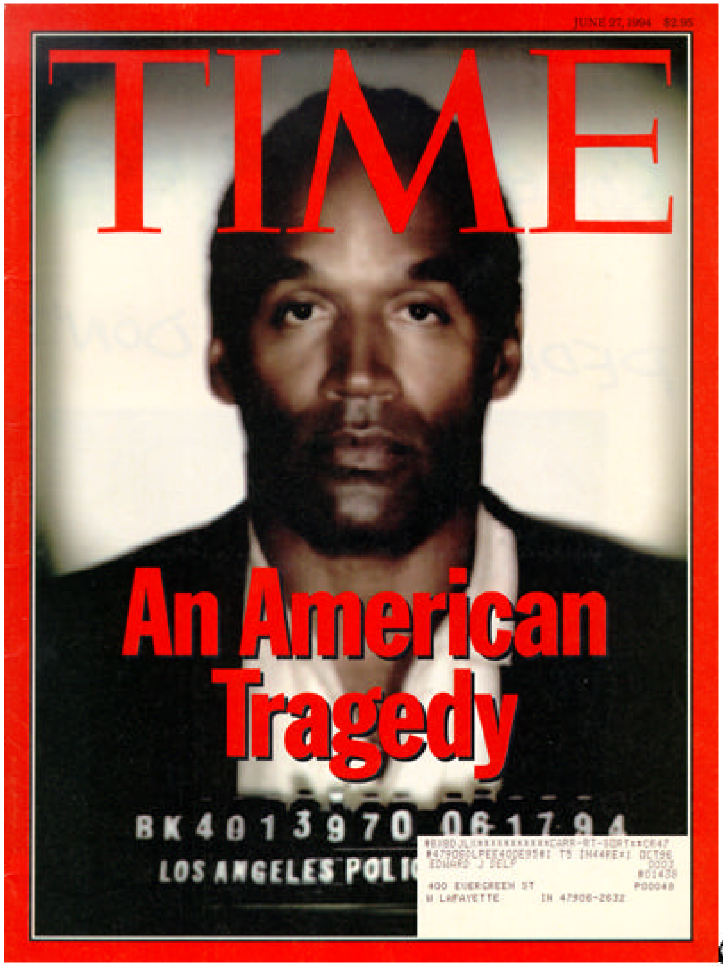
Why is Watermarking Important?

Why Watermarking is Important?

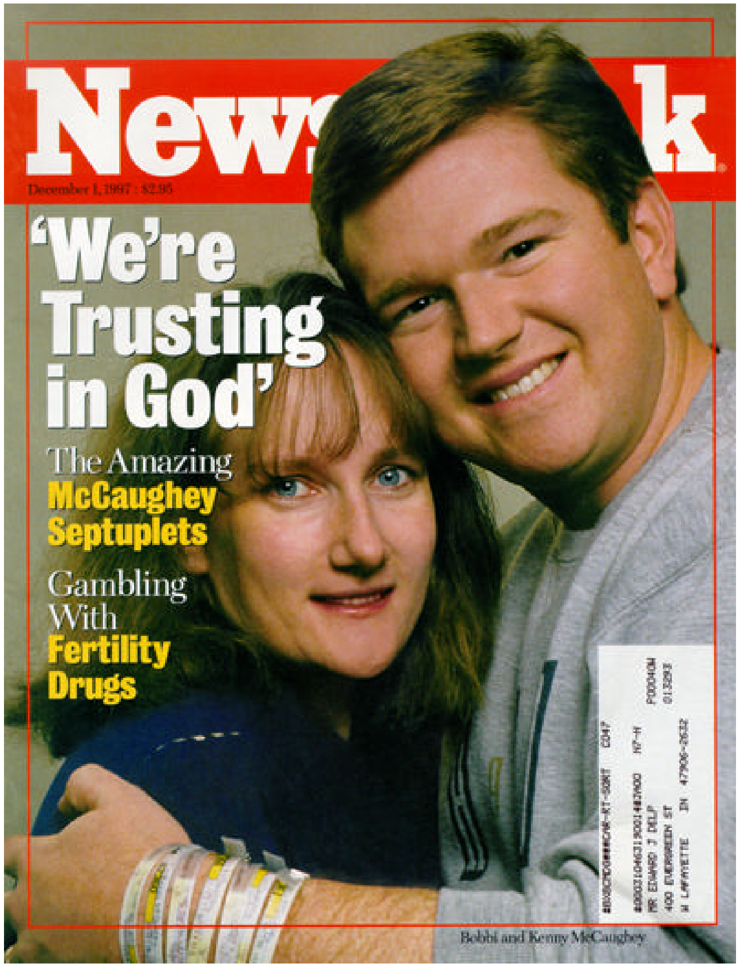


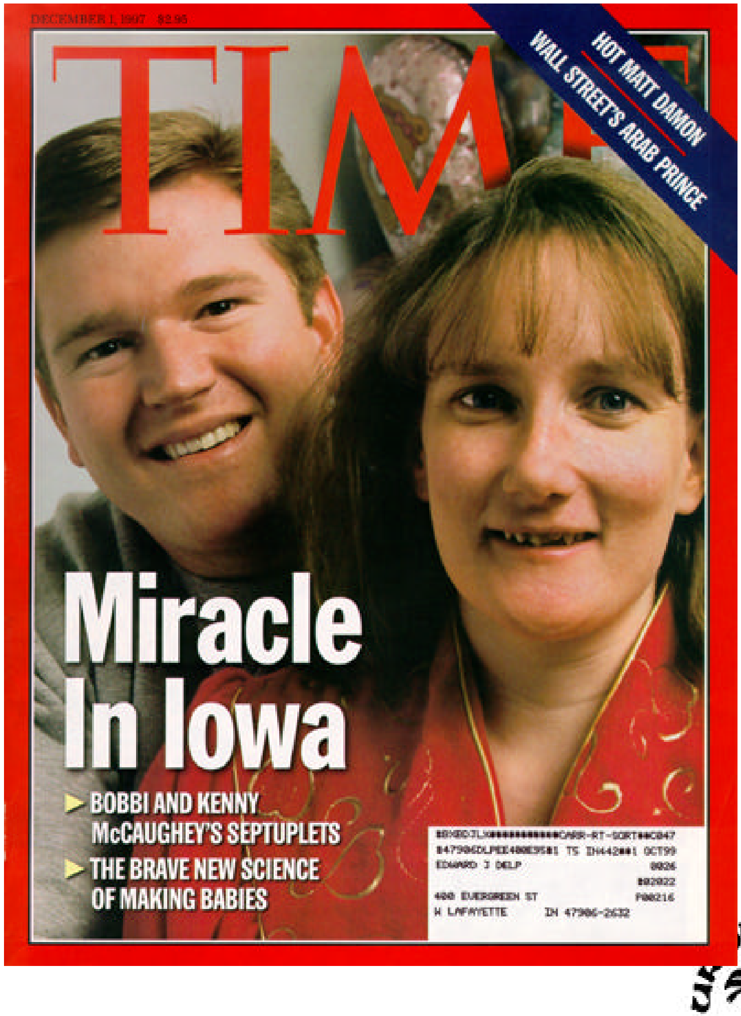
Why is Watermarking Important?


HEALTH CARE: Who's Afraid of 'Rationing'?
GAYS: Tony Kushner on Stonewall 25
Newsweek
June 27, 1994 : $2.95
TRAIL OF BLOOD
BK 4013970
MR EDWARD J DELP
400 EVERGREEN ST
W LAFAYETTE IN 47906-2632
LOS ANGELES POLICE - JAIL DIV

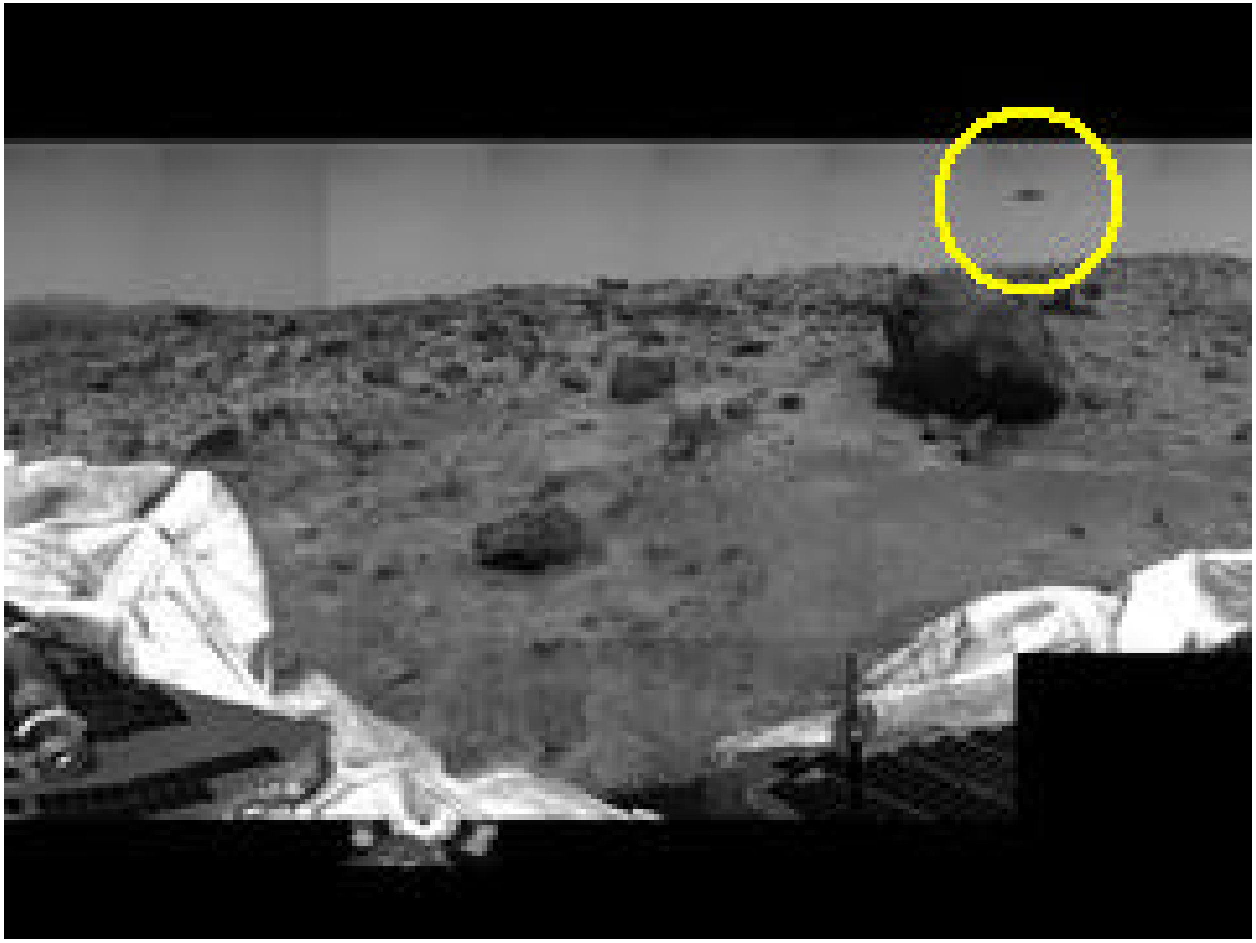


Why is Watermarking Important?

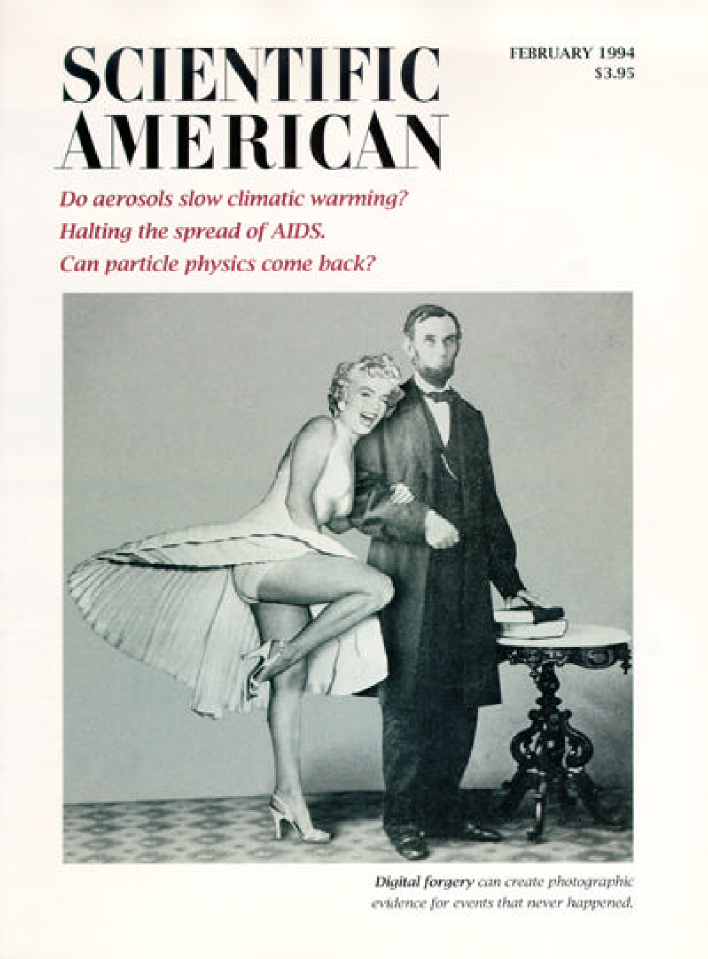

Newsweek
December 1, 1997 : $2.95
'We're Trusting in God'
The Amazing McCaughey Septuplets
Gambling With Fertility Drugs
MR EDWARD J DELP
400 EVERGREEN ST
W LAFAYETTE IN 47906-2632
Bobbi and Kenny McCaughey

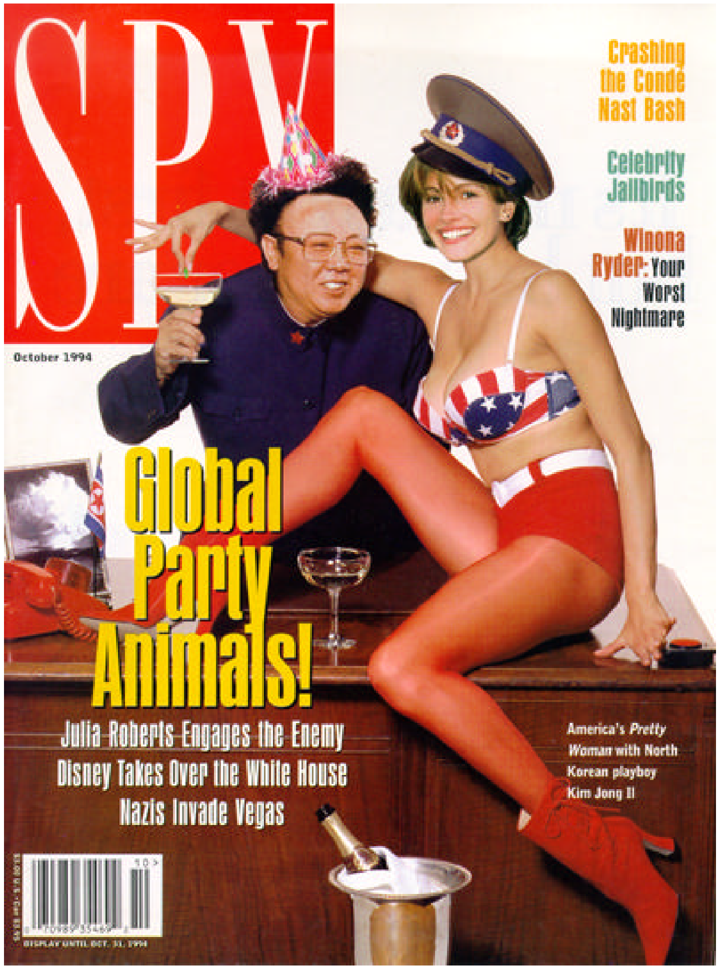

DECEMBER 1, 1997 $2.95
TIME
HOT MATT DAMON
WALL STREET'S ARAB PRINCE
Miracle In Iowa
▶ BOBBI AND KENNY McCAUGHEY'S SEPTUPLETS
▶ THE BRAVE NEW SCIENCE OF MAKING BABIES
EDWARD J DELP
400 EVERGREEN ST
W LAFAYETTE IN 47906-2632

Why is Watermarking Important?

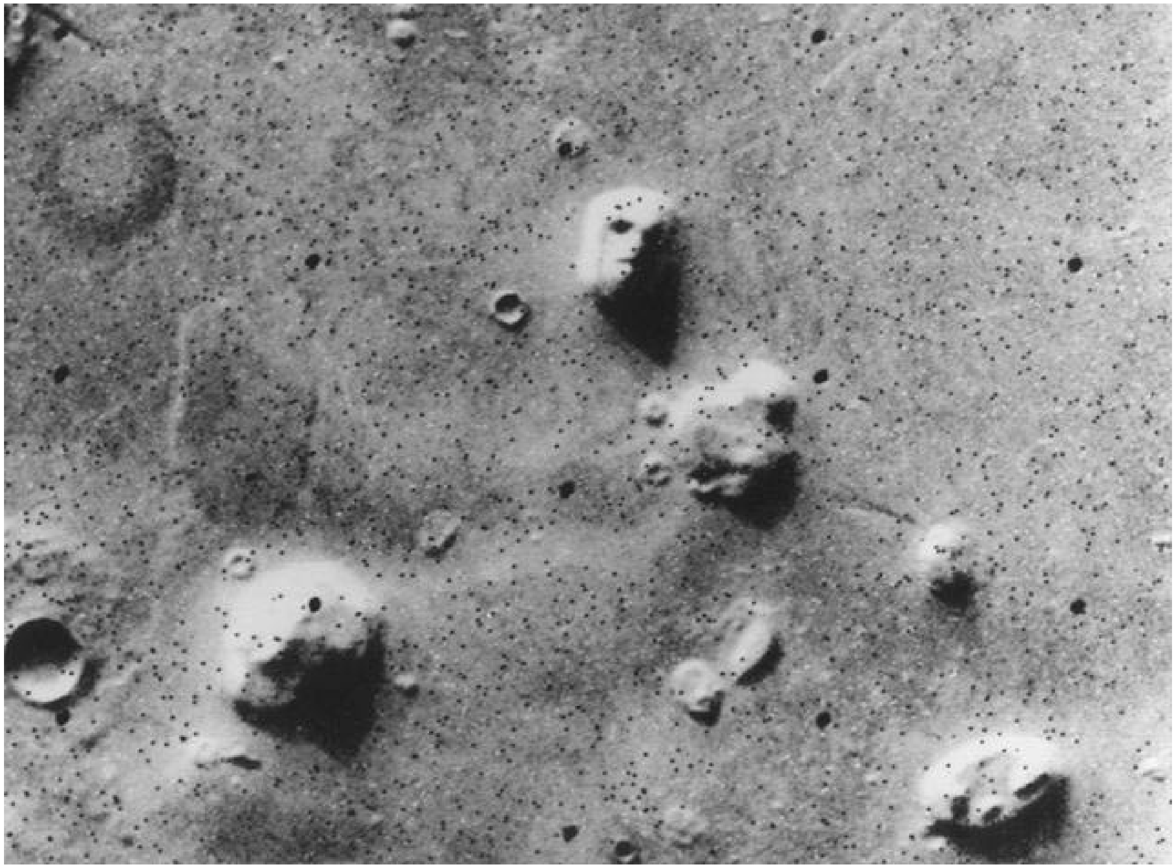


Why is Watermarking Important?


SCIENTIFIC AMERICAN
FEBRUARY 1994
$3.95
Do aerosols slow climatic warming?
Halting the spread of AIDS.
Can particle physics come back?
Digital forgery can create photographic evidence for events that never happened.


SPY
October 1994
Crashing the Conde Nast Bash
Celebrity Jailbirds
Winona Ryder: Your Worst Nightmare
Global Party Animals!
Julia Roberts Engages the Enemy
Disney Takes Over the White House
Nazis Invade Vegas
America's Pretty Woman with North Korean playboy Kim Jong Il
DISPLAY UNTIL OCT. 31, 1994

Why is Watermarking Important?

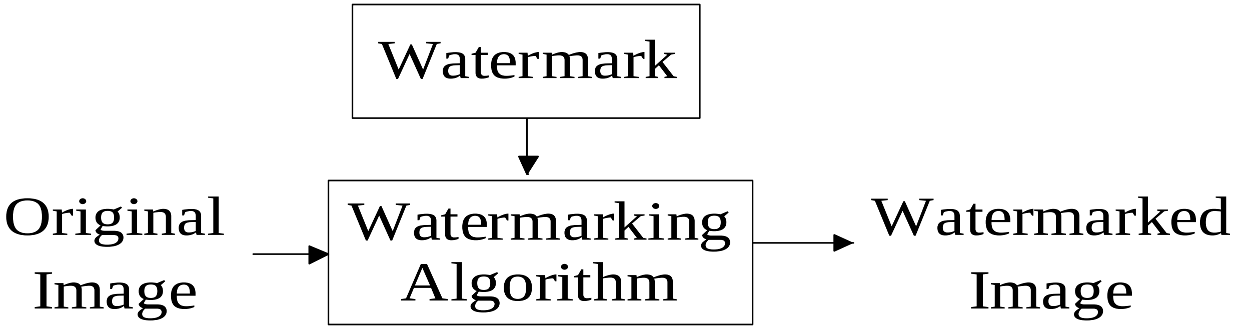


Watermarking

- The use of perceptually invisible authentication techniques is one form of watermarking
  - distortion is introduced in the data
- Other forms include visible watermarks

A Review of Watermarking Techniques

- **Spatial watermarking**
- **Sub-band (wavelet) watermarking**
- **DCT coefficient modulation**
- **Visible watermarks**


Watermark

Original Image

Watermarking Algorithm

Watermarked Image

Components of a Watermarking Technique

- The watermark, W
  - each owner has a unique watermark
- The marking algorithm
  - incorporates the watermark into the image
- Verification algorithm
  - an authentication procedure (determines the integrity / ownership of the image)

Watermark Detection

- The tradeoff of detectability vs. visibility (host signal interference)
- Do you need the original image for detection?
  - If not Þ blind detection
- What about the “key?”
  - private or public?
- These are very important when with video

Main Principles

- **Transparency - the watermark is not visible in the image under typical viewing conditions**

- **Robustness to attacks - the watermark can still be detected after the image has undergone linear and/or nonlinear operations (this may *not* be a good property - *fragile watermarks*)**

- **Capacity - the technique is capable of allowing multiple watermarks to be inserted into the image with each watermark being independently verifiable**

Fragile Watermarks

- Changes to image easily detected and localized
- Used for authentication, rather than copy detection

Fragile Watermarks

- Are we asking too much from a robust watermark?
  - it is a very interesting signal processing/signal detection problem but will lead to a solution or an arms race
- Fragile watermarks do not have to worry about this!

Attacks

- Compression
- Filtering
- Printing and rescanning
- Geometric attacks - cropping, resampling, rotation
- Collusion - spatial and temporal
- Conversion to analog

Current Research Issues

- Theoretical Issues
  - – capacity and performance bounds
  - – models of the watermarking/detection process
- Robust Watermarks
  - – linear vs. nonlinear
  - – scaling and other geometric attacks
  - – watermarking analog representations of content
  - – new detection schemes
  - – what should be embedded (watermark structure)

Fixed-length DCT Watermark

- **W is a sequence of random numbers**
  - **bipolar binary sequence, or N(0,1)**
- **X_D and Y_D are DCT of X and Y**
- **a = scaling factor:**

$$Y_D(i) = X_D(i)(1 + aW)$$

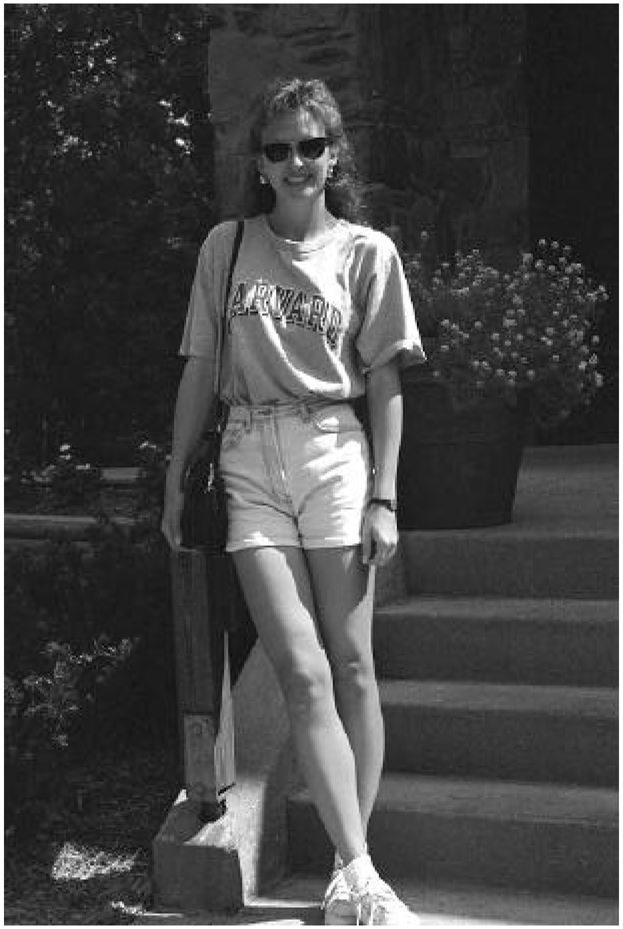
DCT Watermark

- W^* is the extracted version of the watermark
- Verification:

$$S(W, W^*) = \frac{W \cdot W^*}{\sqrt{W^* \cdot W^*}}$$

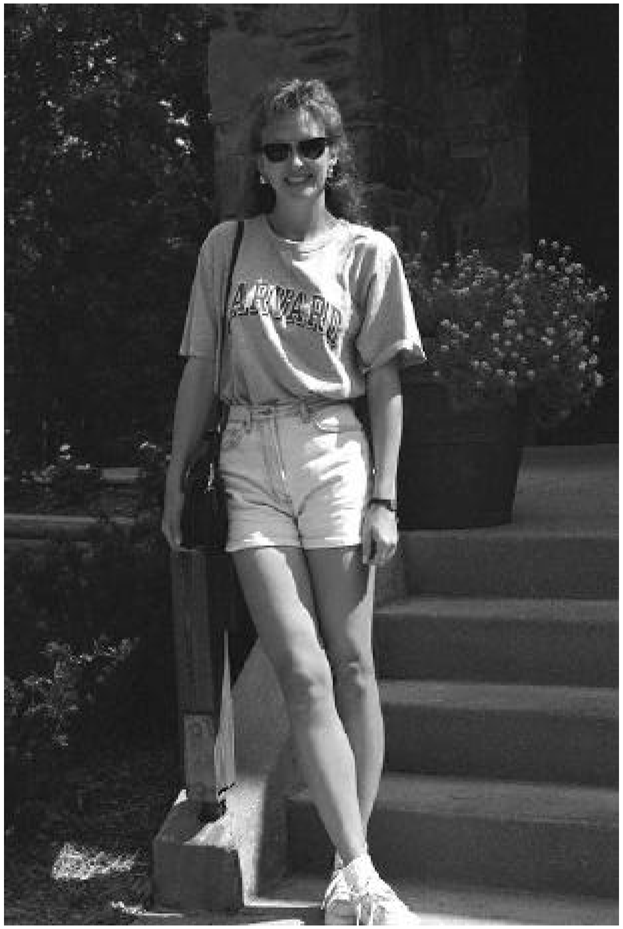
- T = user-defined threshold
- If S > T, image is authentic

Original Image



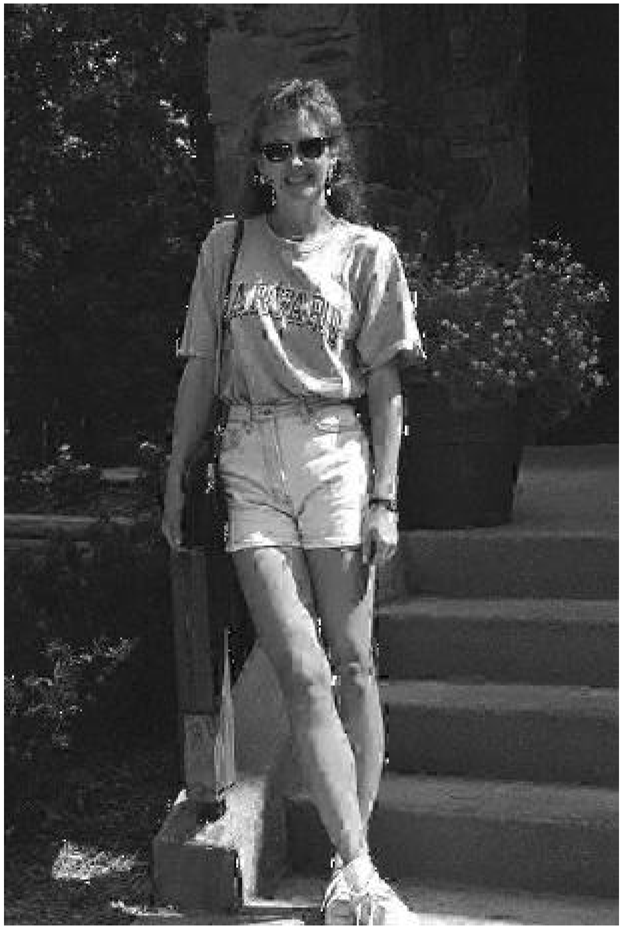
Fixed-length DCT Watermark

a = 0.1



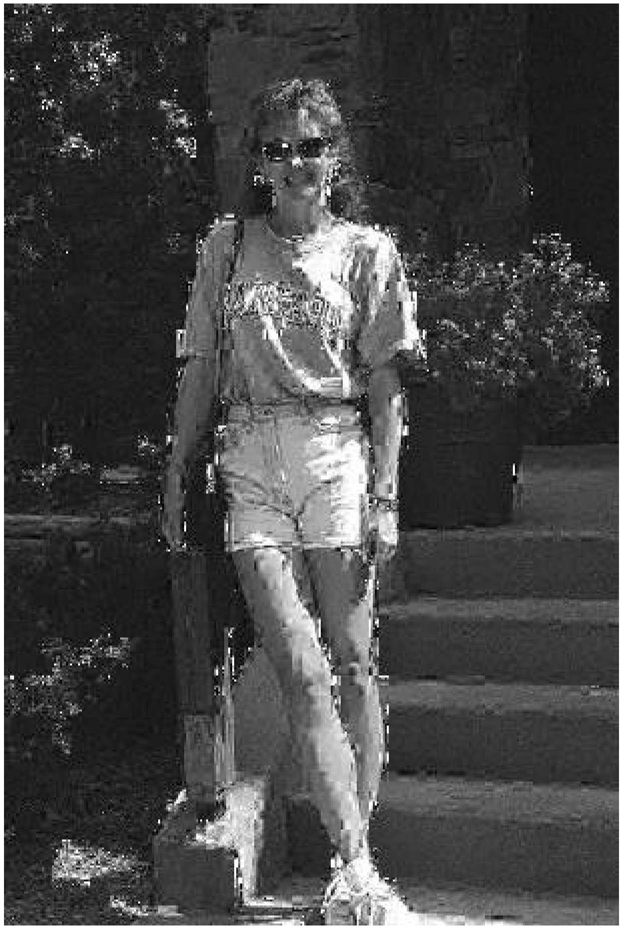
Fixed-length DCT Watermark

a = 0.5



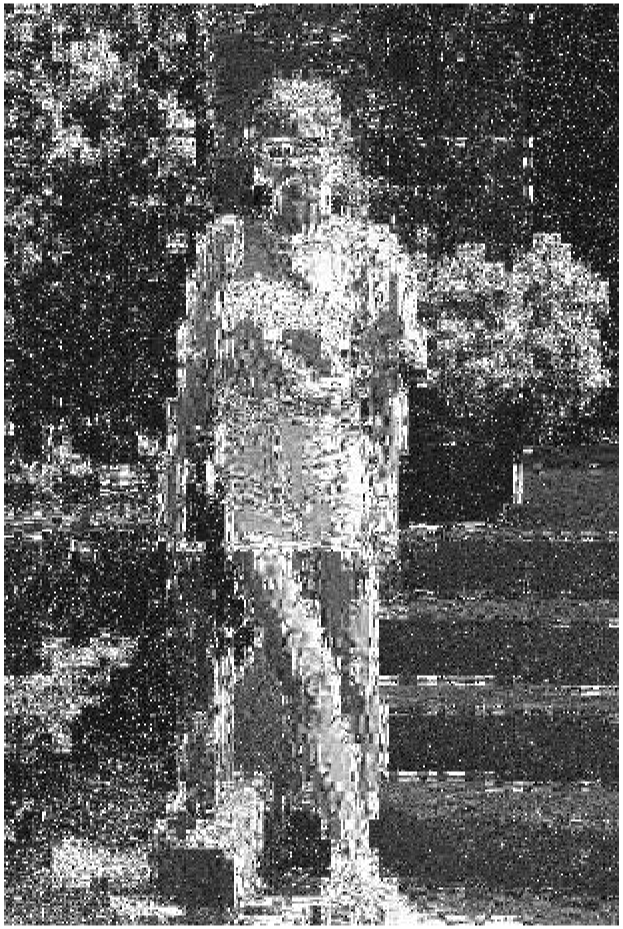
Fixed-length DCT Watermark

a = 1.0



Fixed-length DCT Watermark

a = 5.0



Levels of Transparency

1. **Spatial Watermarks**
   - **watermark embedded in lower order bit planes**
   - **mark imperceptible, or indistinguishable from noise**

2. **Transform Domain Watermarks**
   - **watermark sequence added to transform coefficients**
   - **transforms based on human visual system**
   - **DCT, Wavelet**

Levels of Transparency

3. Image Adaptive (IA) Watermarks

- **transform based**
  - **IA-W : 9-7 biorthogonal wavelet**
  - **IA-DCT : Discrete Cosine Transform**
- **different transform coefficients can tolerate different amounts of change, before changes are noticed**
- **amounts determined by formal visual models**
- **amplitude of watermark adjusted according to these amounts**
- **most robust of the three levels**

Visual Models

- Used extensively in image compression
- Determine how much a transform coefficient can imperceptibly change

- "Just noticeable difference" (*JND*) value determines:
  - – quantization step size
  - – watermark amplitude

- Models developed for DCT, Wavelet

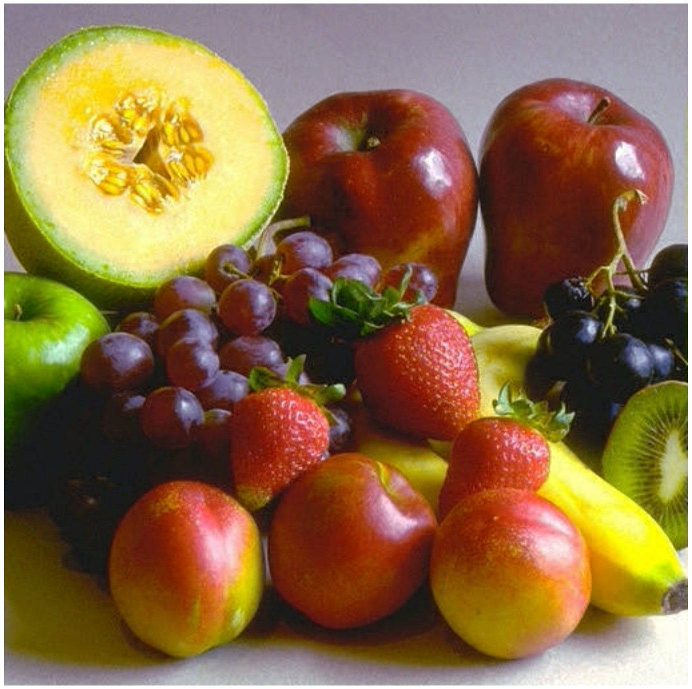
Image Adaptive Watermarks

- *JND* scales the watermark before insertion into image

$$Y_{u,v} = \begin{cases} X_{u,v} + J_{u,v} W_{u,v}, & X_{u,v} > J_{u,v} \\ X_{u,v}, & otherwise \end{cases}$$

- $\boldsymbol{X_{u,v}}$ = original transform coefficient
- $\boldsymbol{J_{u,v}}$ = JND value for particular coefficient
- $\boldsymbol{W_{u,v}}$ = watermark element
- $\boldsymbol{Y_{u,v}}$ = marked coefficient

Image Adaptive Watermarks (DCT)



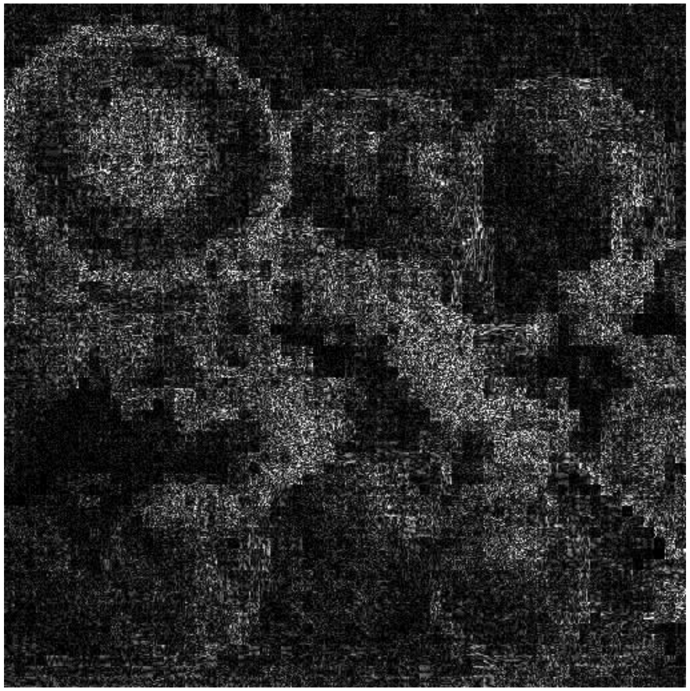


Image Adaptive Watermarks (DCT)

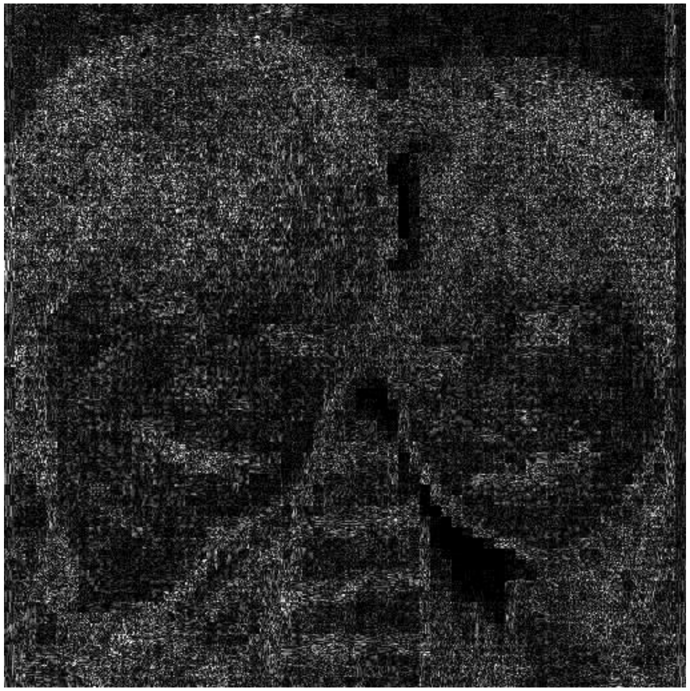
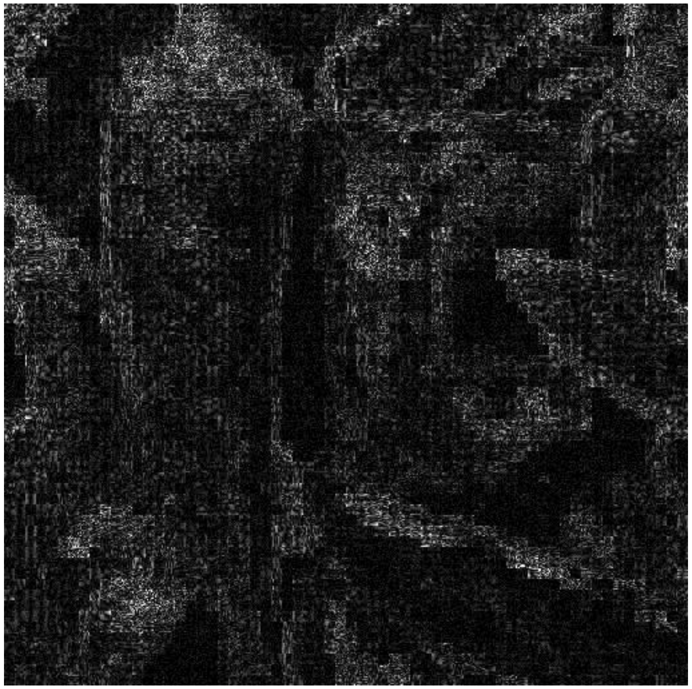


Image Adaptive Watermarks (DCT)



IBM ATTACK

- **Description**
  - ***X*** **~ original,** ***Y*₁** **~ image marked with** ***W*₁**

$$Y_1 = X + W_1$$

  - **Create counterfeit original,** $\boldsymbol{X_F}$

$$X_F = Y_1 - W_2$$

  - $\boldsymbol{Y_1}$ **now appears to be a marked version of** $\boldsymbol{X_F}$

$$\Rightarrow Y_1 = X_F + W_2$$

IBM Attack

- **What is the original, X or X_F ?**
- **Attacker can claim X_F is true original**

Thwarting the IBM Attack

- **Time stamp original image, X**

- **Let time stamp certificate (S) of X be part of W**
  - **W now depends on X, owner name and creation date**
  - **only one legitimate W per original image**
  - **attacker needs correct time stamp of X_F before generating X_F**

Steganography

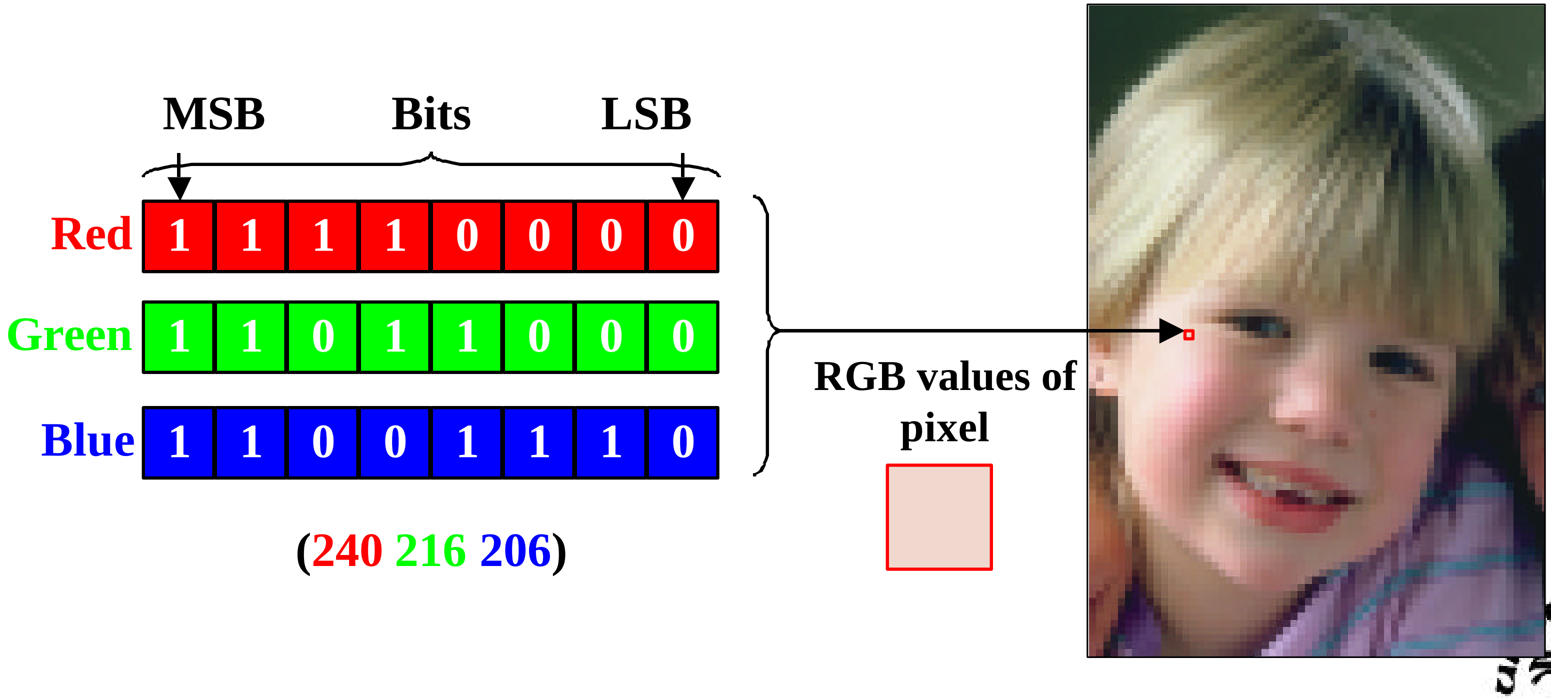
Introduction

Digital Color Images

- Visual data is organized as array of pixel values in the spatial domain
- Many different image formats
- In personal computing:
  - RGB (Red, Green, Blue) color space is common
  - Each color component assigned an integer value between 0 to 255, or 8 bits
  - (0,0,0) corresponds to darkest black; (255,255,255) corresponds to brightest white

24-Bit Color Images

- In images with 24 bits/pixel, there is a Red, Green, and Blue value for each pixel location



MSB Bits LSB

Red	1	1	1	1	0	0	0	0
Green	1	1	0	1	1	0	0	0
Blue	1	1	0	0	1	1	1	0

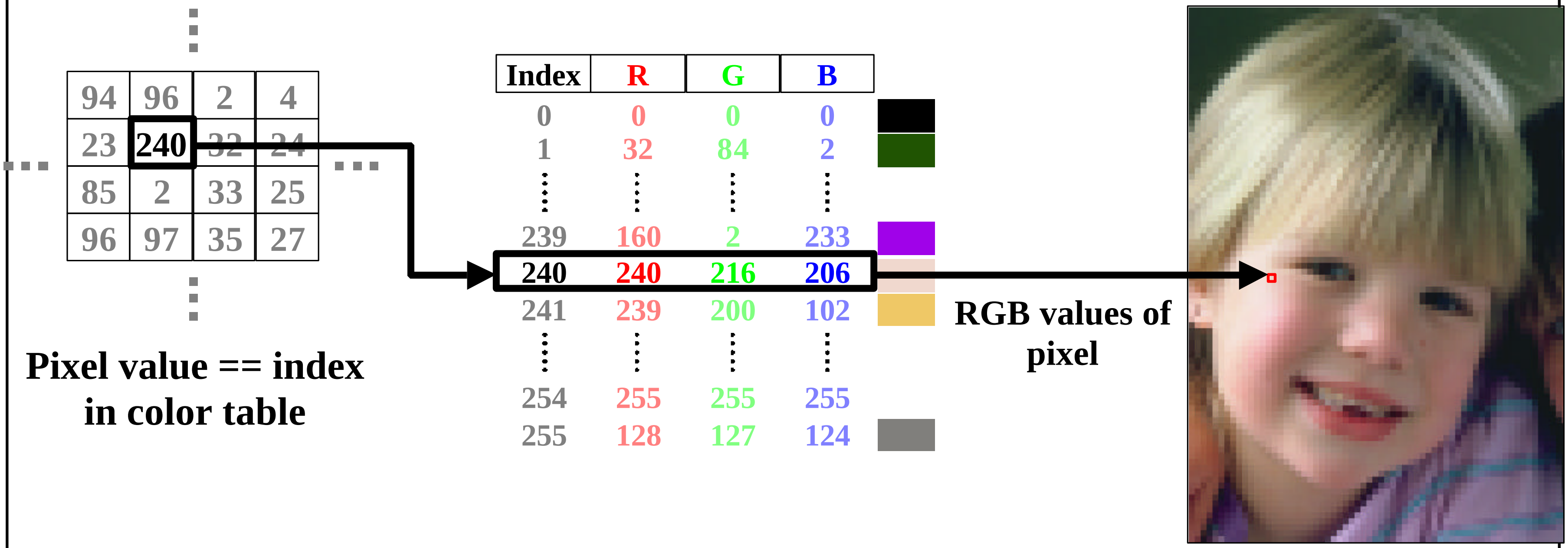
(240 216 206)

RGB values of pixel

8-Bit Color Images

- In images with 8 bits/pixel, pixel values represent indices in a color lookup table or palette
  - Each entry in palette has Red, Green, Blue values
  - Size (# of entries) of palette == maximum number of colors used in the image
  - Size can be up to 256 entries for 8 bits/pixel
  - Pixel values do not correspond to RGB values

8-Bit Color Images



Array of Pixel Values

94	96	2	4
23	**240**	32	24
85	2	33	25
96	97	35	27

Pixel value == index in color table

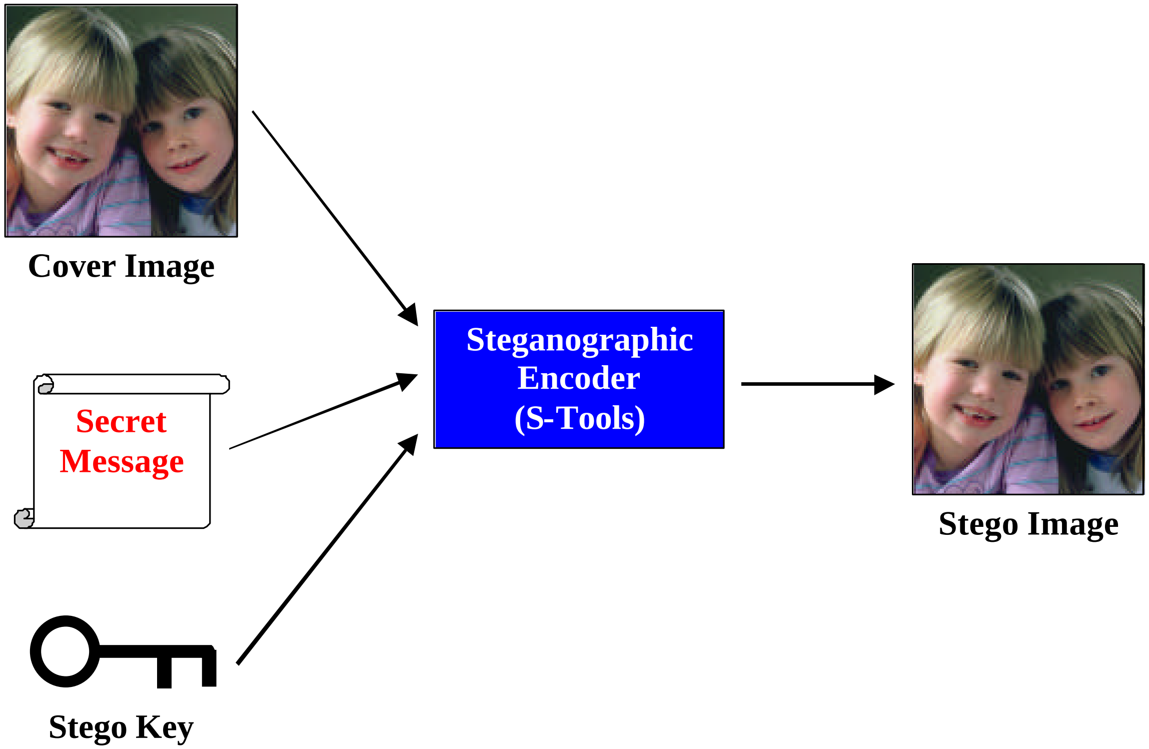
Color Lookup Table

Index	R	G	B	
0	0	0	0	⬛
1	32	84	2	🟩
⋮	⋮	⋮	⋮	
239	160	2	233	🟪
240	**240**	**216**	**206**	⬜
241	239	200	102	🟨
⋮	⋮	⋮	⋮	
254	255	255	255	
255	128	127	124	⬜

RGB values of pixel

Digital Image Steganography

- Objective: covertly embed data inside images
- Steganography: “Covered Writing” (Greek)



Cover Image

Secret Message

Stego Key

Steganographic Encoder (S-Tools)

Stego Image

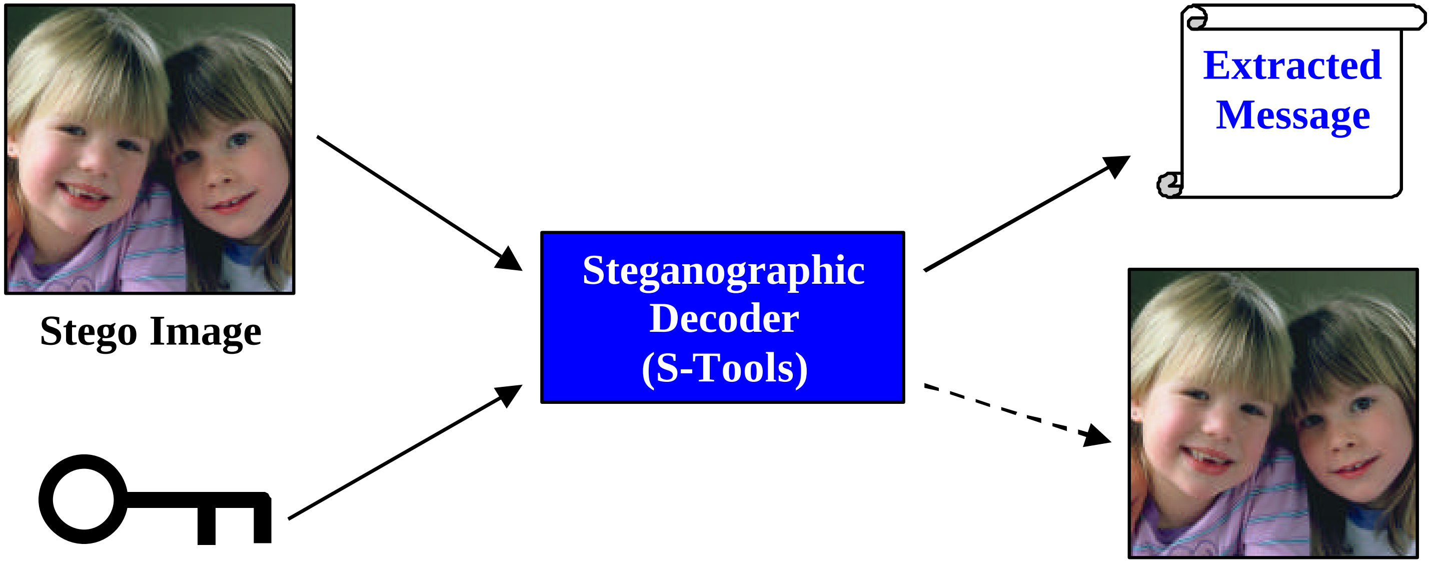
Digital Image Steganography

- Cover Image = Original image
- Message = Data (files) to be hidden
- Stego Image = Altered image containing hidden data
- Stego Key = Secret needed to embed or recover message

- Cover Image often innocuous to avoid suspicion
- Stego Image appears identical to cover image under causal observation
- Maximum size of message (capacity) depends on steganography technique and cover image

Digital Image Steganography

- Secret Message can be recovered by decoder
  - If stego image has not been altered, extracted message will be identical to the secret message



Stego Image

Stego Key

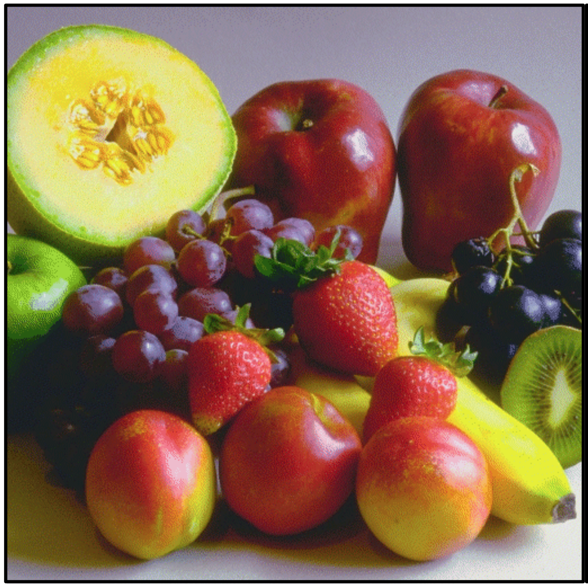
Steganographic Decoder (S-Tools)

Extracted Message

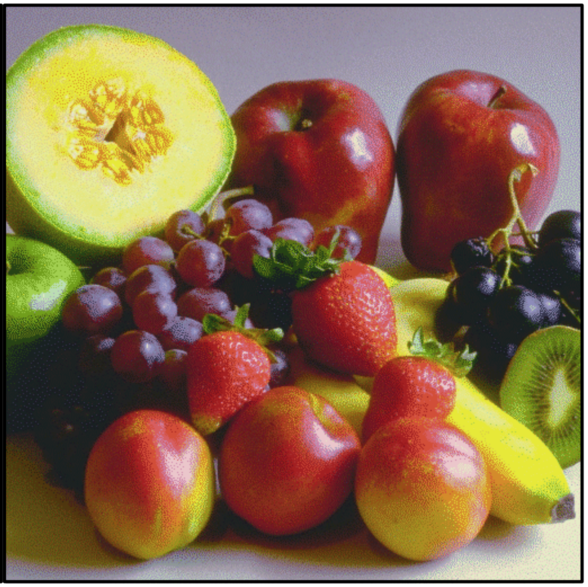
Cover Image

Digital Image Steganography

- Example cover and stego images produced by S-Tools 4.0
  - Message: `This is a test message demonstrating the S-Tools 4.0 steganography software.`
  - Stego key: **`STEGO`**



Cover Image
(8 bits/pixel)



Stego Image
(8 bits/pixel)

Steganalysis

- Examination of suspect images:
  - Visual inspection (color artifacts, graininess, excess noise)
  - Stochastic or statistical analysis
  - Histogram analysis
- Extracting an embedded message without knowledge of stego key
- Deducing the stego key

Overview of S-Tools Software

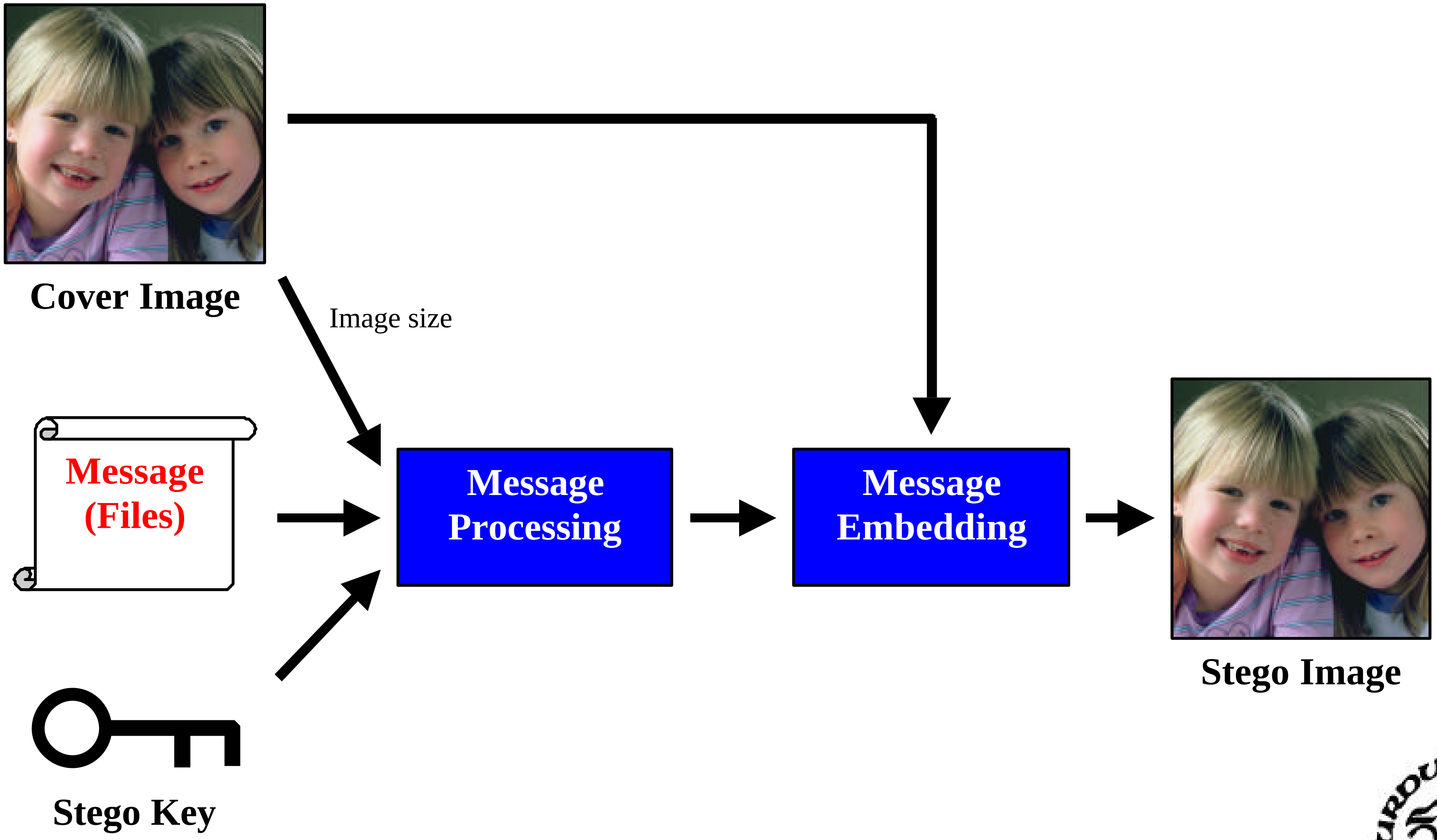
S-Tools 4.0 Overview

- Created in 1996 by Andrew Brown
  - Not compatible with previous versions of S-Tools
  - Software handles both encoding and decoding

- Capacity is 3 bits/pixel with ~128 bits overhead

- Different methods of embedding for
  - 24-bit images (BMP)
  - 8-bit images (BMP, GIF)
  - Does not support JPEG, TIFF, PNG images
  - Can also hide data in audio (WAV) files

S-Tools 4.0 Encoding Process



Cover Image

Image size

Message (Files)

Stego Key

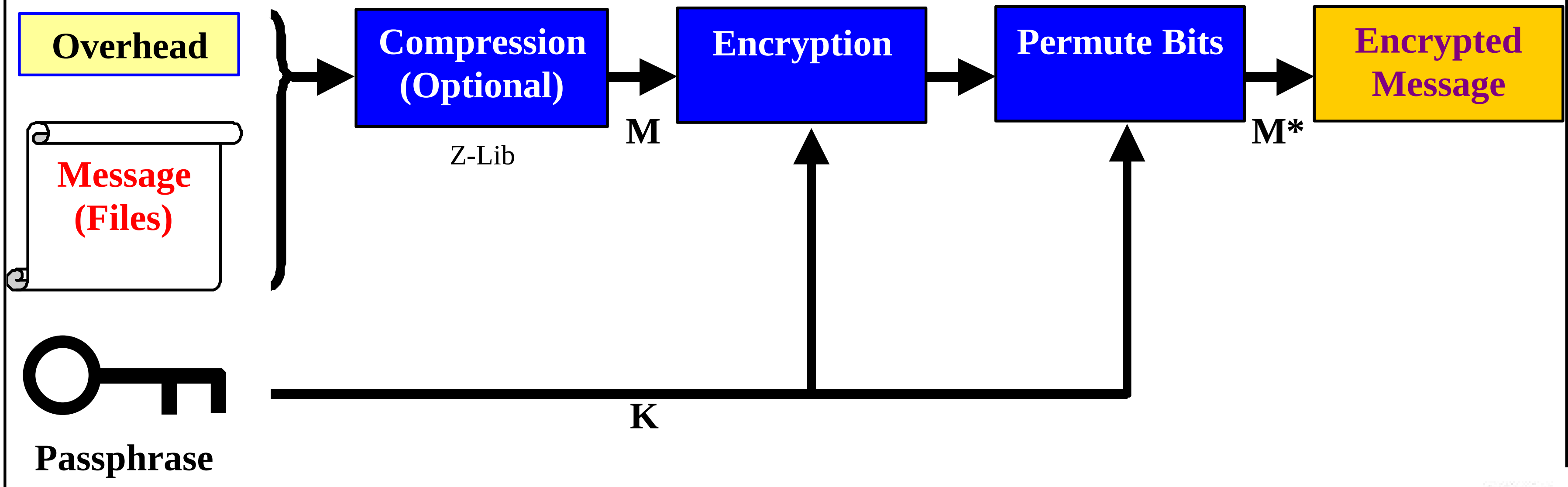
Message Processing

Message Embedding

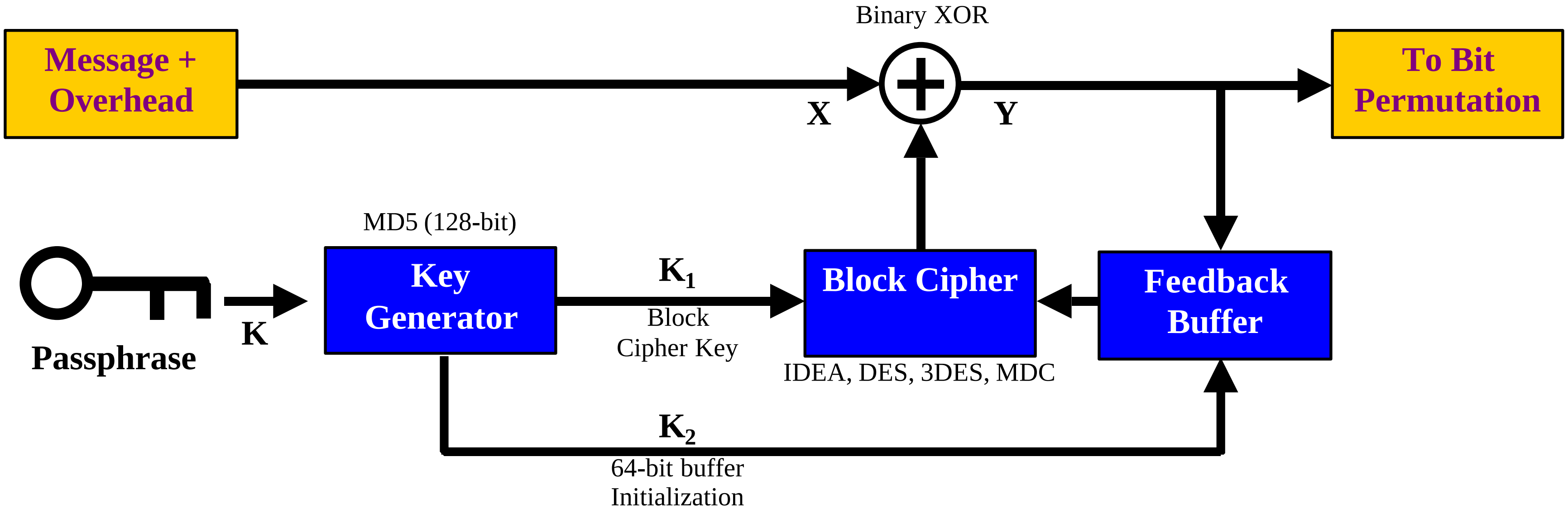
Stego Image

Message Processing

- Goal: Encrypt (scramble) the message



Message Encryption Details



- Cipher Feedback Mode (CFB)
- User can select block cipher (defaults to IDEA)

Bit Permutation Details

- Message bits are shuffled randomly
  - The decoder can reverse shuffling process with knowledge of the passphrase
  - Complicates message recovery for an attacker
  - Permutation performed on the entire message, not 64-bit blocks

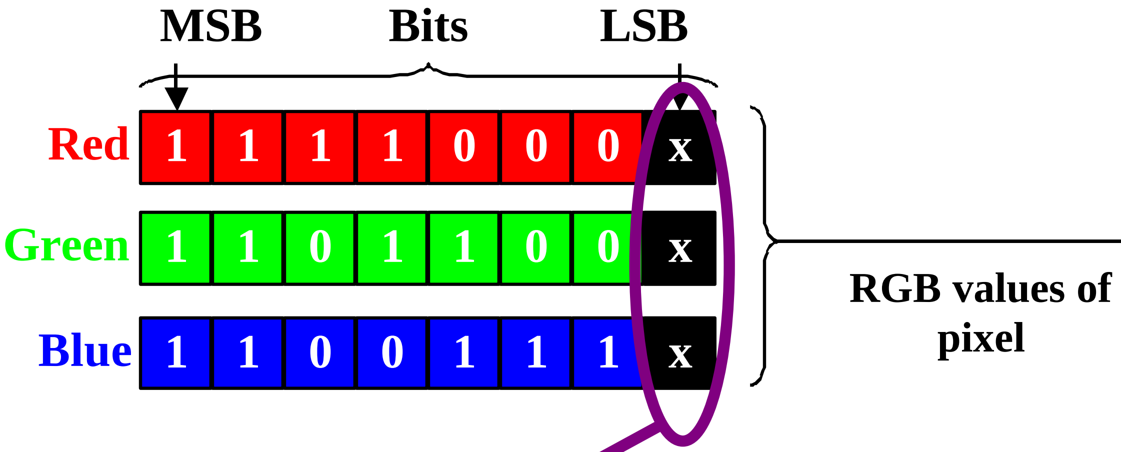
Message Embedding

- Goal: Embed the encrypted message into cover image by altering pixel values
  - The altered image becomes the stego image

- Embedding process differs for 24-bit images and 8-bit images

Message Embedding in 24-Bit Images

- One encrypted message bit is embedded in the LSB of the Red, Green, and Blue values of each pixel



Red	1	1	1	1	0	0	0	x
Green	1	1	0	1	1	0	0	x
Blue	1	1	0	0	1	1	1	x

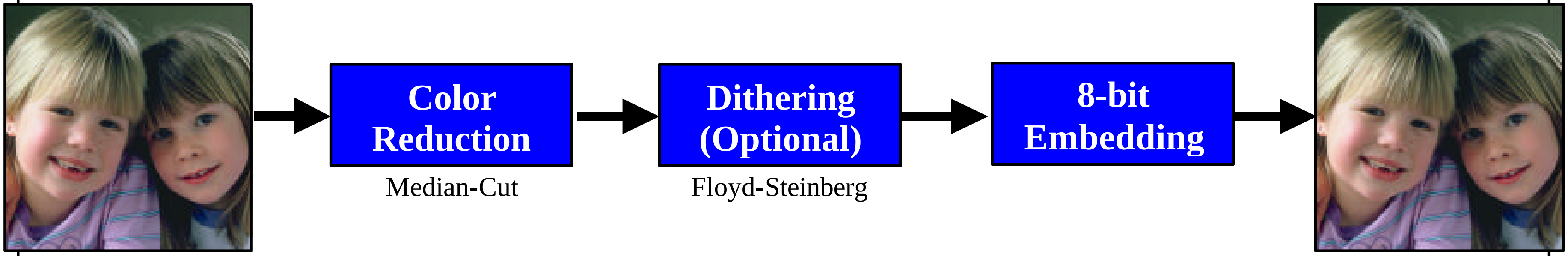
These bits are changed to that of the encrypted message

Embedding in 8-Bit Images

- **Problem: Pixel values represent index in color table, not intensity values**
  - **In general, changing color indices renders image useless**

- **Strategy: Reduce number of colors in the image, then embed message in reduced color image**
  - **User also has choice to convert 8-bit image to 24-bits and using 24-bit embedding**

Embedding in 8-Bit Images



Color Reduction

Median-Cut

Dithering (Optional)

Floyd-Steinberg

8-bit Embedding

Cover Image

Stego Image

- Color reduction reduces colors in image
- Dithering can improve visual quality
- 8-bit embedding inserts encrypted message

Color Reduction

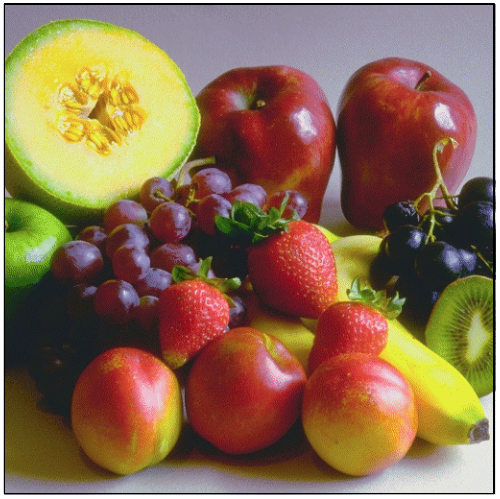
- **Median Cut Algorithm described in**

  P. S. Heckbert, “Color image quantization for frame buffer display”, *Computer Graphics*, vol. 16, no. 3, pp. 297-303, 1982.

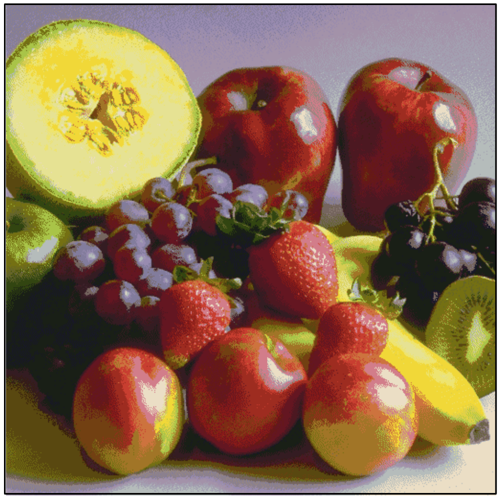
  – **Used in S-Tools to reduce the number of colors in the 8-bit image from 256 to 32**

  – **Color reduction can be noticeable, usually reducing visual quality of image**

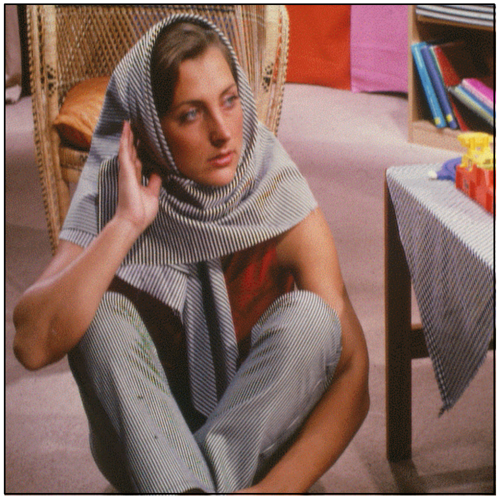
Color Reduction Examples



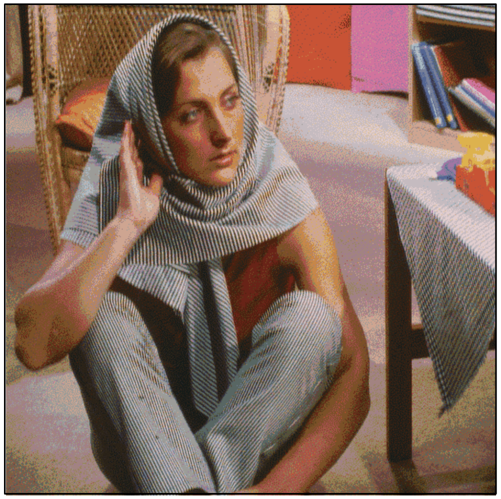
256 color “Fruit”



32 color “Fruit”



256 color “Barbara”



32 color “Barbara”

Dithering

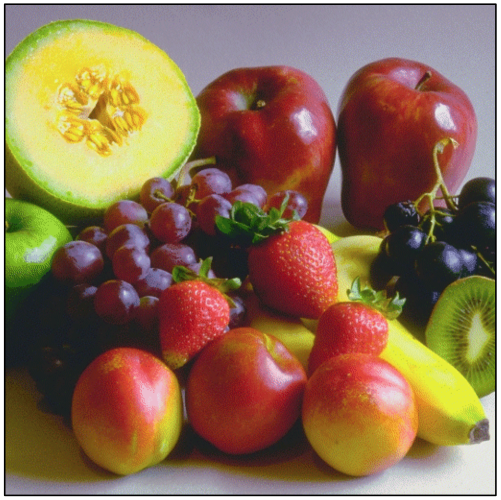
- Dithering process improves visual quality by reducing "contouring" effects

  R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grayscale," *Proceedings of the Society for Information Display*, vol. 17, no. 2, 1976, pp. 75-77.

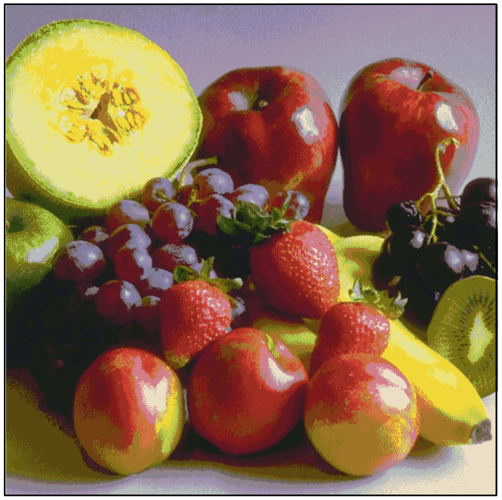
  – Floyd-Steinberg dithering does not change color table or increase the colors used

- User can disable dithering if desired (default is enabled)

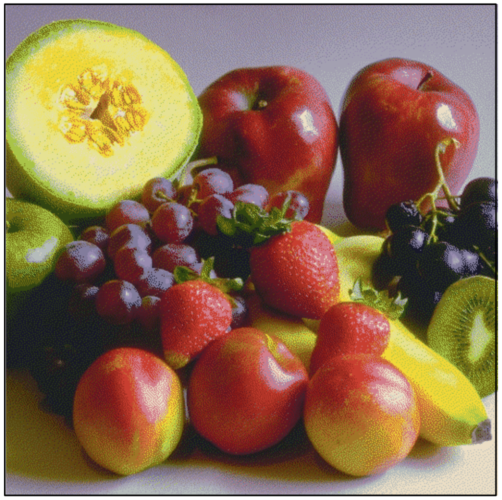
Dithering Examples



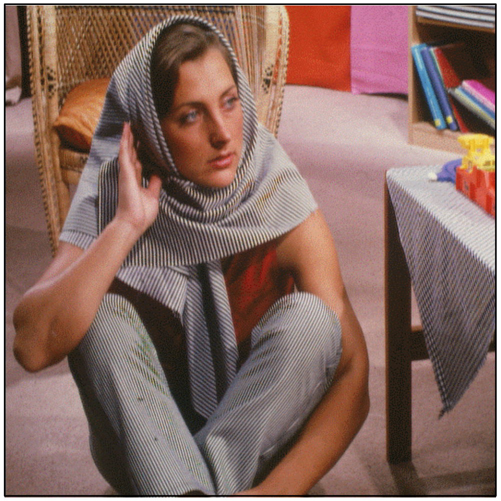
256 color “Fruit”



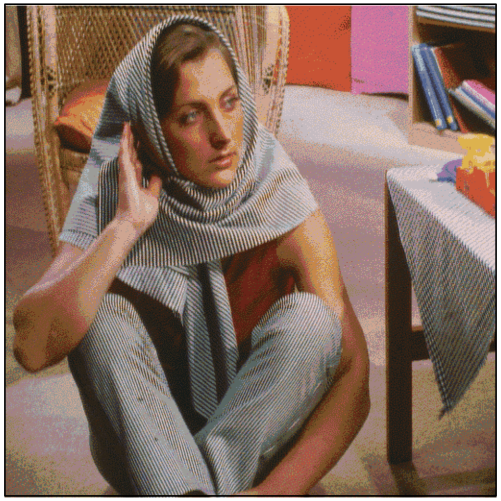
32 color “Fruit”



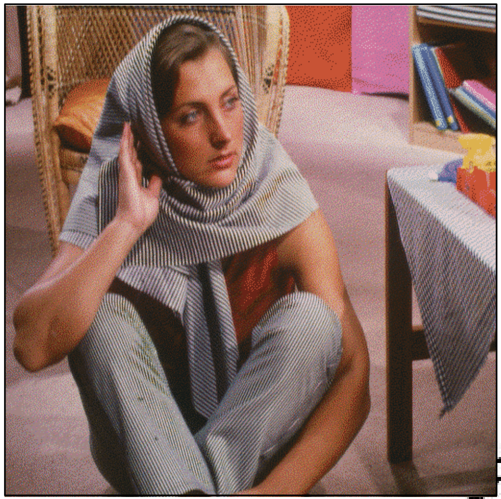
32 color “Fruit” (dithered)



256 color “Barbara”



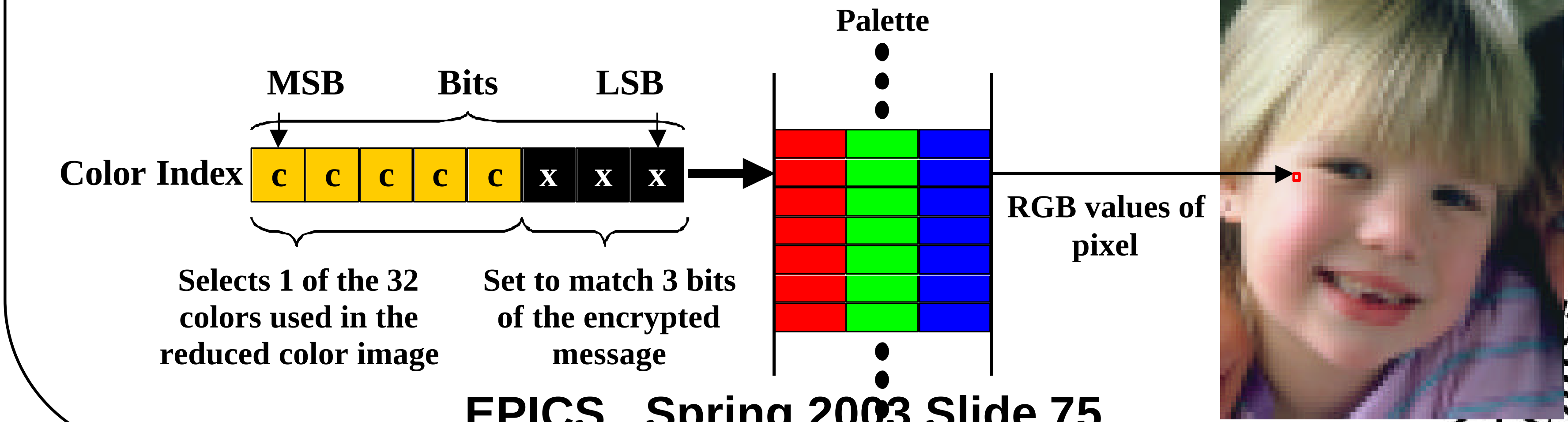
32 color “Barbara”



32 color “Barbara” (dithered)

Message Embedding in 8-Bit Images

- Message is embedded in least 3-LSBs of each pixel
  - – A 256 entry color table is constructed for stego image, but every 8 color entries is similar (differing LSB)
    - Thus, the lowest 3-LSBs can be changed to any combination and image will appear similar
    - Palette entries may be shuffled after embedding to a


Palette
MSB
Bits
LSB
Color Index
c
c
c
c
c
x
x
x
RGB values of pixel
Selects 1 of the 32 colors used in the reduced color image
Set to match 3 bits of the encrypted message

Summary

- Steps: Message Processing, Message Embedding
  - Message Processing encrypts messages
  - Message Embedding inserts the encrypted message into cover image
- Message embedding differs for 8-bit and 24-bit cover images
  - 24-bit: Straightforward LSB embedding
  - 8-bit: Color reduction, dithering, 3-LSB embedding
- Decoding is straightforward

Attacking S-Tools

Cover Image Issues

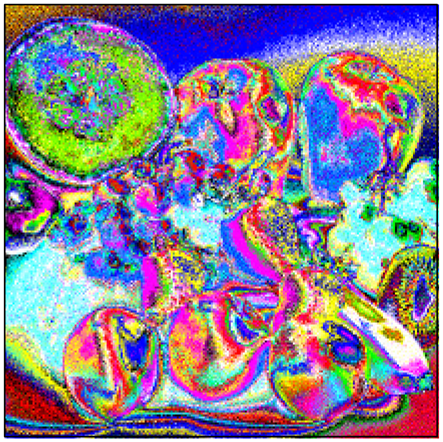
- Cover image can strongly influence steganalysis
- Poor cover images can produce stego images that can be suspect by visual inspection

- Poor cover images include:
  - Images with large monochromatic regions
  - Images with smooth transitions in color
  - Images with too few colors
  - 8-bit images with too many colors
  - Computer generated images
  - Decompressed JPEG images [Fridrich]

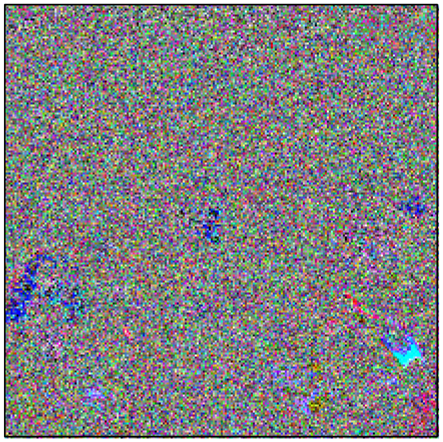
Is LSB Independent of Image?

Original Fruit (24 bpp)

Stego Image (24 bpp)



LSB Original Fruit



LSB Stego Image

Statistical Steganalysis vs. LSB

- Attacks exploiting Pair of Values
  - Chi-squared comparison between suspect image and image with pair-wise color values [Westfeld]
  - Raw Quick Pairs technique [Fridrich]
- Examining smoothness of groups of pixels (RS) [Fridrich]

Other Attacks

- **Palette Analysis [Johnson]**
  - **S-Tools creates “clusters” of similar colors in palette for 8-bit images because of color reduction**
  - **Attack examines palette for color clustering**

Conclusions

- S-Tools steganography uses least-significant bit (LSB) embedding for embedding message
- For many cover images, LSB is not independent of the image content
- Many methods for inspecting suspect images
- Determining message without stego key or the stego key itself may be difficult

References

- **S-Tools 4.0 software by Andrew Brown (a.brown@nexor.co.uk, 1996)**
  - **Z-LIB compression component by Jean-loup Gailly and Mark Adler**
  - **CRYPTLIB encryption component by Peter Gutmann, Eric Young and Colin Plumb**

- **P. S. Heckbert, “Color image quantization for frame buffer display”, *Computer Graphics*, vol. 16, no. 3, pp. 297-303, 1982.**
- **R. Floyd and L. Steinberg, “An adaptive algorithm for spatial grayscale,” *Proceedings of the Society for Information Display*, vol. 17, no. 2, 1976, pp. 75-77.**

Steganalysis References

- J. Fridrich, M. Goljan, "Practical steganalysis of digital images—State of the art," *Proc. SPIE Security and Watermarking of Multimedia Contents IV*, vol. 4675, San Jose, CA, January 2002.
- J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compatibility," *Proc. SPIE Multimedia Systems and Applications IV*, Denver, CO, August 20-24, 2001.
- J. Fridrich, R. Du, and L. Meng, "Steganalysis of LSB encoding in color images," *Proc. IEEE International Conference on Multimedia and Expo,* New York, NY, July 30-August 2, 2000.
- N. Johnson and S. Jajodia, "Steganalysis of images created using current steganography software," *Lecture Notes in Computer Science*, vol. 1525, Springer-Verlag, Berlin, 1998, pp. 273-289.
- A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems: Breaking the steganographic utilities EzStego, Jsteg, Steganos, and S-Tools—and some lessons learned," *Proc. Workshop on Information Hiding*, Dresden, Germany, September 29-October 1, 1999.
- R. Chandramouli, "A mathematical approach to steganalysis," *Proc. SPIE Security and Watermarking of Multimedia Contents IV*, vol. 4675, San Jose, CA, January 2002, pp.14-25.

“Anti-Watermarking” Techniques

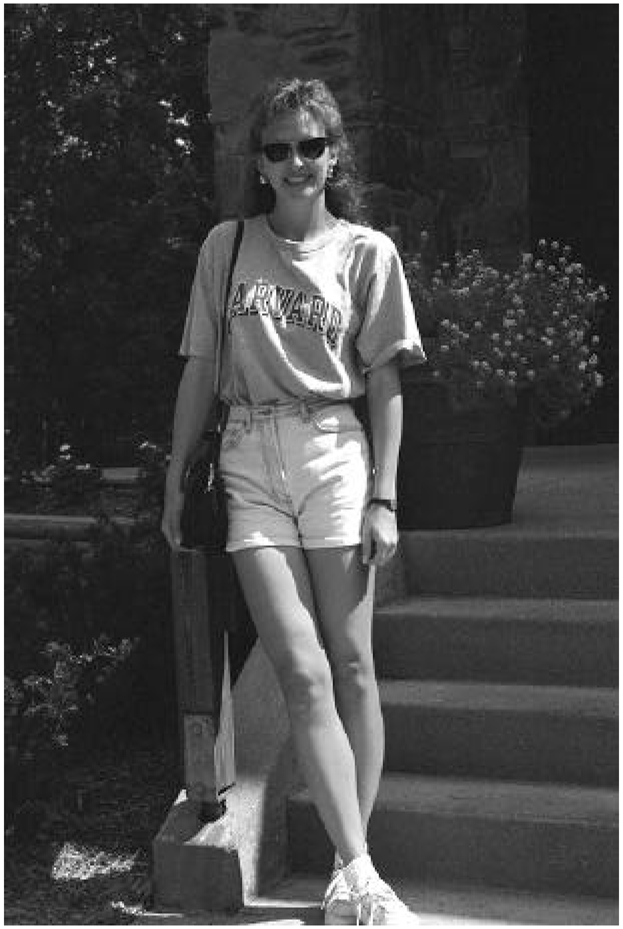
- UnSign software
  - – http://altern.org/watermark/

- StirMark
  - – http://www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/stirmark/

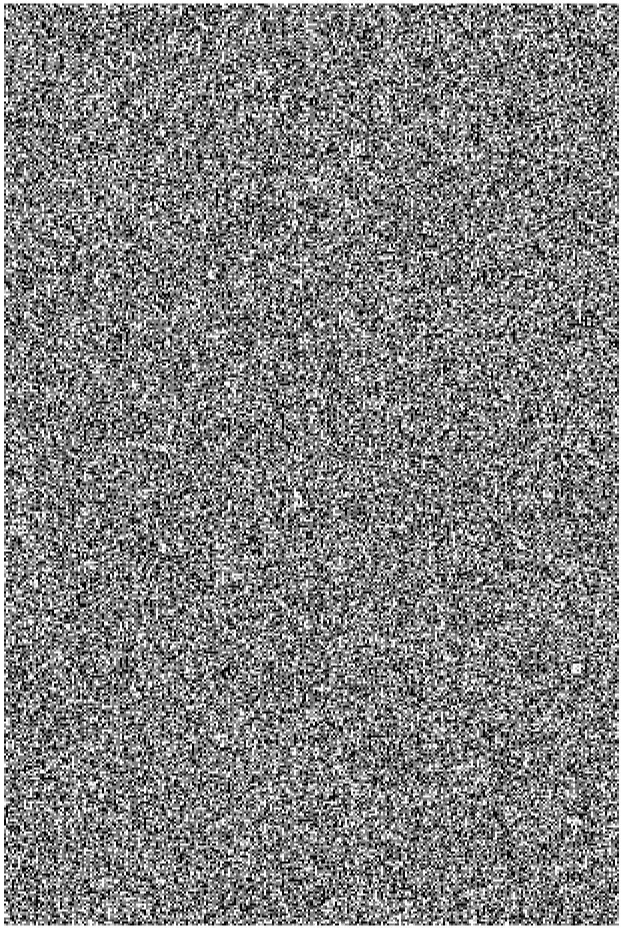
Research at Purdue

- Fragile and semi-fragile watermarks for forensic imaging
  - are fragile watermarks better than hashing?
- Extending concept of robust image adaptive watermarks to video (with Chris Podilchuk)
  - is there a temporal masking model that works?

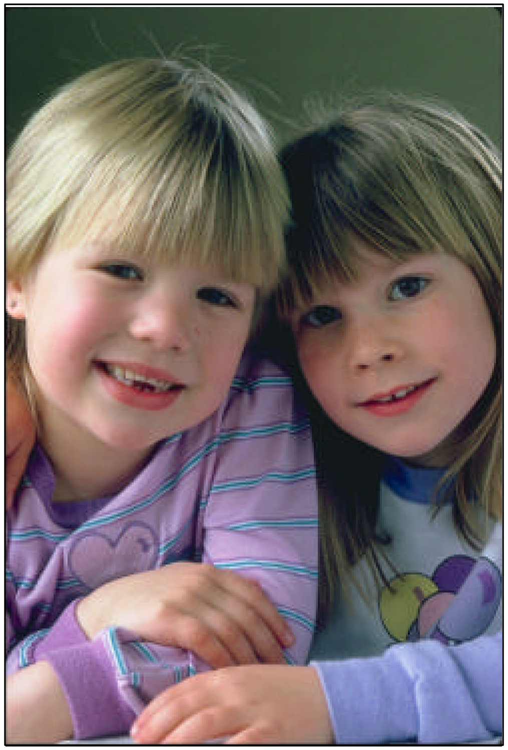
VW2D Watermarked Image



VW2D Difference Image

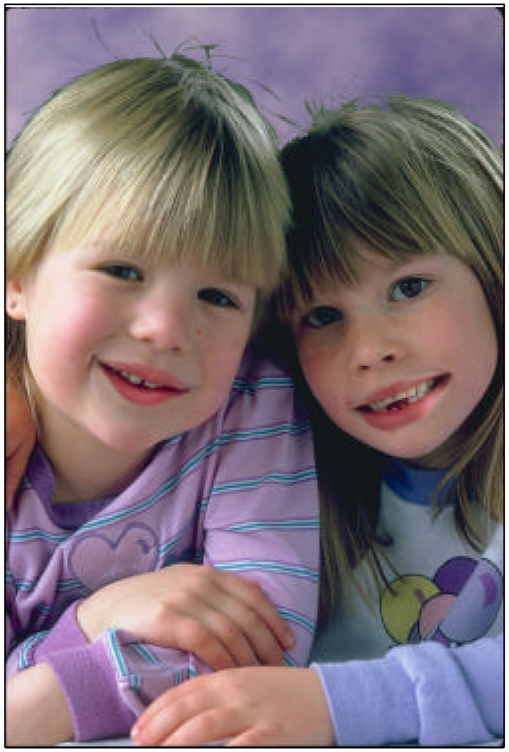


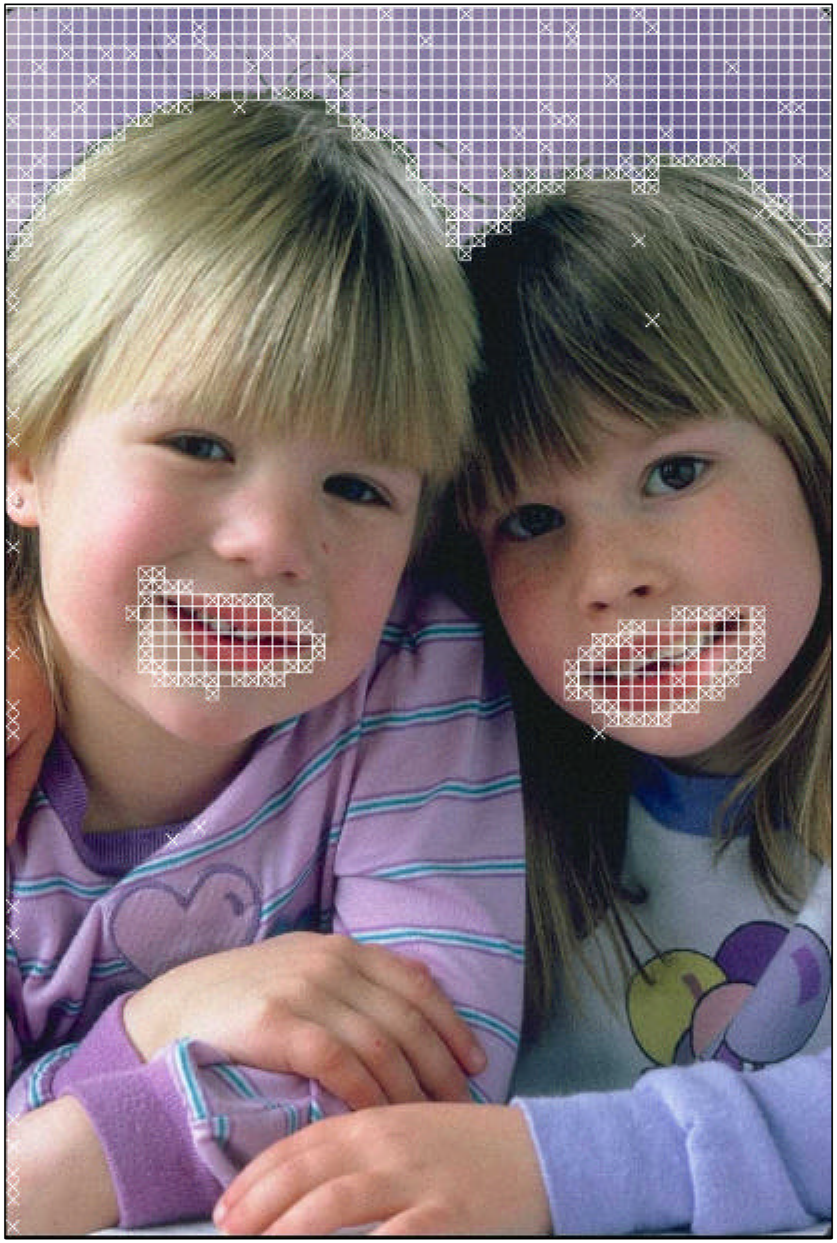
Results - Girls



¬ **Original “Girls”**

Altered “Girls” ®





Video Watermarking Issues

- A video sequence cannot simply be treated as an ordered collection of images:
  - – visibility issues in the use of “still” image watermarks
  - – visibility issues in stop frames
  - – human perception of motion is not accounted for in visual models for still images
  - – embedding the same watermark in all the frames of a video sequence is not secure, an attacker can correlate across the entire sequence to estimate the watermark (temporal collusion)

Video Watermarking Issues

- – embedding completely different watermarks in successive frames of a video sequence is not secure
- – successive video frames are highly correlated, an attacker can exploit this to estimate and remove a watermark
- – the techniques for compressing video do not necessarily encode each frame of the sequence identically
- – the synchronization of the audio with the video sequence may be a consideration for watermark protection

Video Watermarking

- Use still image approaches
  - may have problem in MPEG with B and P frames
- Hash parts of the compressed video stream
- Techniques could be used to prevent multiple viewing, copying, and editing (e. g. inserts)
- Can the watermark survive the conversion back to an analog signal?

Video Watermarking

- Cannot trivially extend image watermarking techniques
  - additional attacks are possible
  - computationally very expensive

- Unique attacks on video watermarks
  - frame shuffling / insertion
  - inter-frame collusion

Watermarking of Compressed Video

Digital Video Compression

- Uncompressed digital video:

Video	Dimensions	Bits/pixel	Frames/sec	Bits/sec
CIF	352 x 288	12	24	29,196,288
CCIR601	720 x 480	16	30	165,888,000
HDTV	1920 x 1080	20	60	2,488,320,000

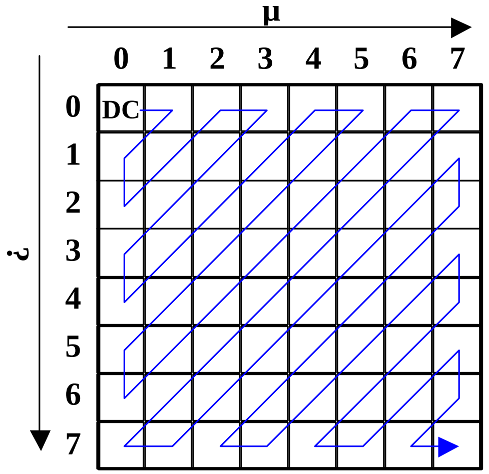
- Requires lots of network bandwidth or storage space

Digital Video Compression

- Motion Compensated Block DCT Coding
  - MPEG-1, MPEG-2, MPEG-4, H.261, H.263
- Remove spatial and temporal redundancy
  - Discrete Cosine Transform
  - Block-based motion compensation
- Quantization
- Entropy coding

Spatial Image Coding (DCT)

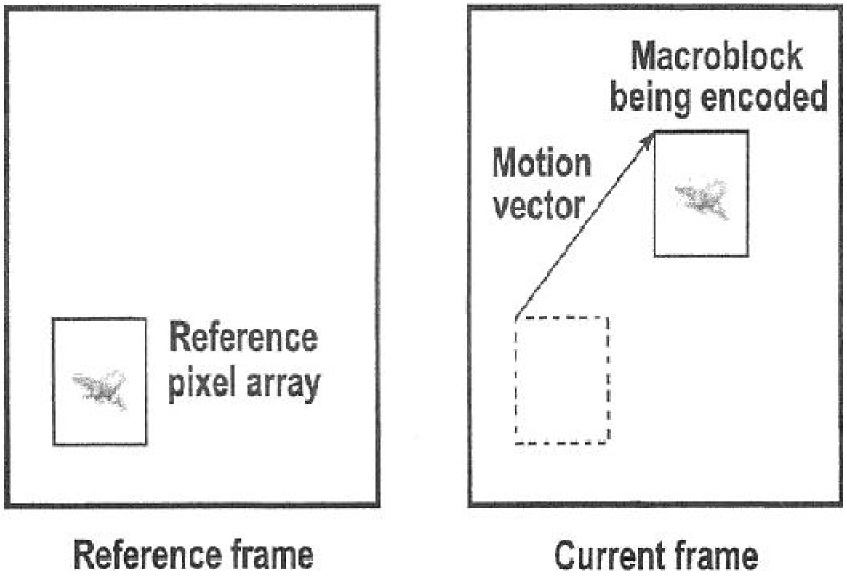
- Very similar to JPEG
- After DCT and quantization, many coefficients are zero
- Non-zero coefficients are coded:
  - Location (zig-zag)
  - Quantization index/value
- Zero coefficients not coded



Temporal Prediction

- Successive frames of video typically similar
  - Differences often in areas of motion

- Strategy: Predict contents of frame by indicating displacement of blocks from previously decoded frame(s)

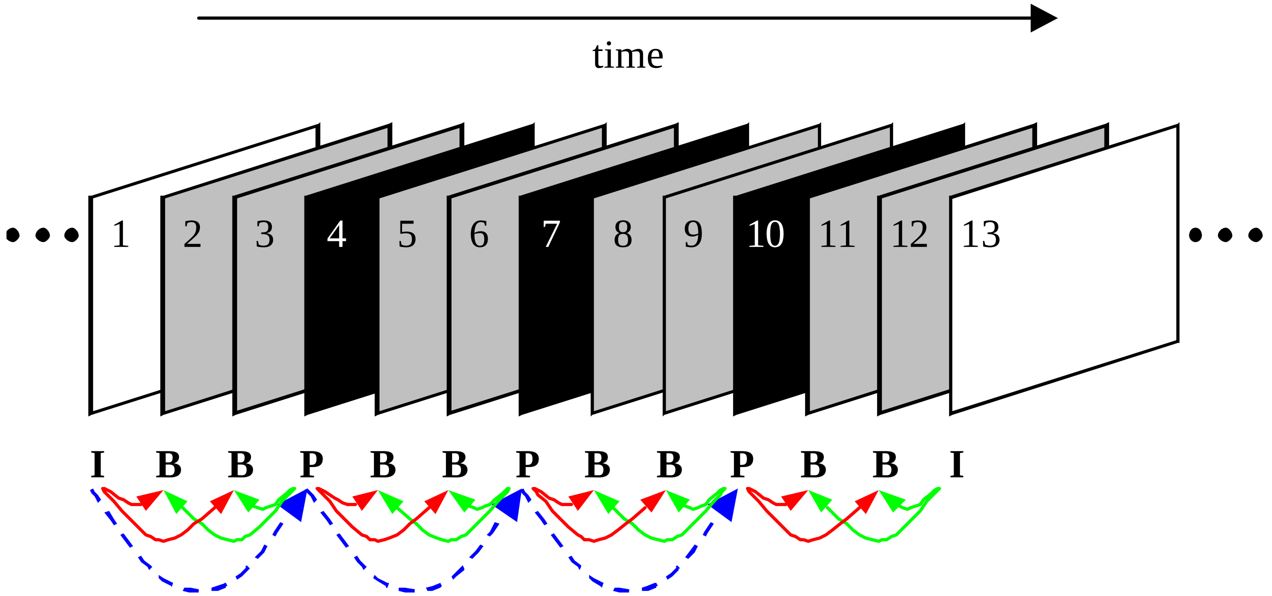


Temporal Prediction

- Encoding (Motion Estimation):
  - – Computationally expensive search
  - – For each block in predicted frame, must find most suitable match in reference frame

- Decoding (Motion Compensation):
  - – Relatively simple block copying

Temporal Prediction



time

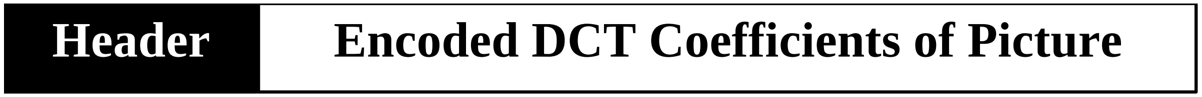
Beginning of stream
(Decoded First)

End of stream
(Decoded Later)

1	4	2	3	7	5	6	10	8	9	13	11	12

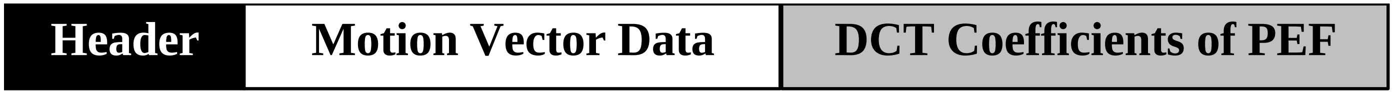
Compressed Video Data

- I-Pictures: “JPEG”, no motion vectors



Header | Encoded DCT Coefficients of Picture

- P/B-Pictures: Motion vector + DCT of residual



Header | Motion Vector Data | DCT Coefficients of PEF

- Other data in compressed video stream:
  - Synchronization and multiplexing
  - Audio

Watermarking Compressed Video

- “Classical” approach:
  - Decompress the original video
  - Insert watermark
  - Compress the watermarked video

- Disadvantages:
  - Computationally expensive
  - Compression can damage watermark
  - Must insert watermark with excess strength
  - Watermark embedder does not know compression parameters
  - Re-compression can degrade video further

Watermarking Compressed Video

- **Compressed-domain approach**
  - **Partially decode the compressed video**
  - **Insert watermark by altering syntactic elements of video (such as DCT coeffs)**
  - **Re-assemble the compressed video stream**

- **No motion estimation during re-assembly**
- **Watermark embedder ensures that alterations to video stream preserve decodeability**

Watermarking Compressed Video

- Advantages
  - Lower computational cost
  - Watermark embedder can make informed decisions based on compression parameters
  - Do not need to embed with excess strength

- Disadvantages:
  - Must parse compressed video data during watermark embedding
  - Watermark insertion is constrained by allowable syntax / semantics of compressed video stream

Watermarking Issues

- General watermarking issues:
  - Capacity, robustness, perceptibility, security
  - Synchronization
  - Attacks
  - Computational complexity

- Compressed domain issues:
  - Drift compensation
  - Preservation of data rate

Drift Compensation

- Drift occurs when a predictor is modified without adjusting the residual

Original Signal

3	6	4	2	2	2	4	4

Predictive (Differential) Coding

3	+3	-2	-2	0	0	2	0

Residual

Original Coded Signal	3	+3	-2	-2	0	0	2	0
Watermark			+1			+1		
Watermarked Signal	3	+3	-1	-2	0	1	2	0
Reconstructed Watermarked Signal	3	6	5	3	3	4	6	6

Drift Compensation

- Must compensate for drift error during watermarking

Original Signal

3	6	4	2	2	2	4	4

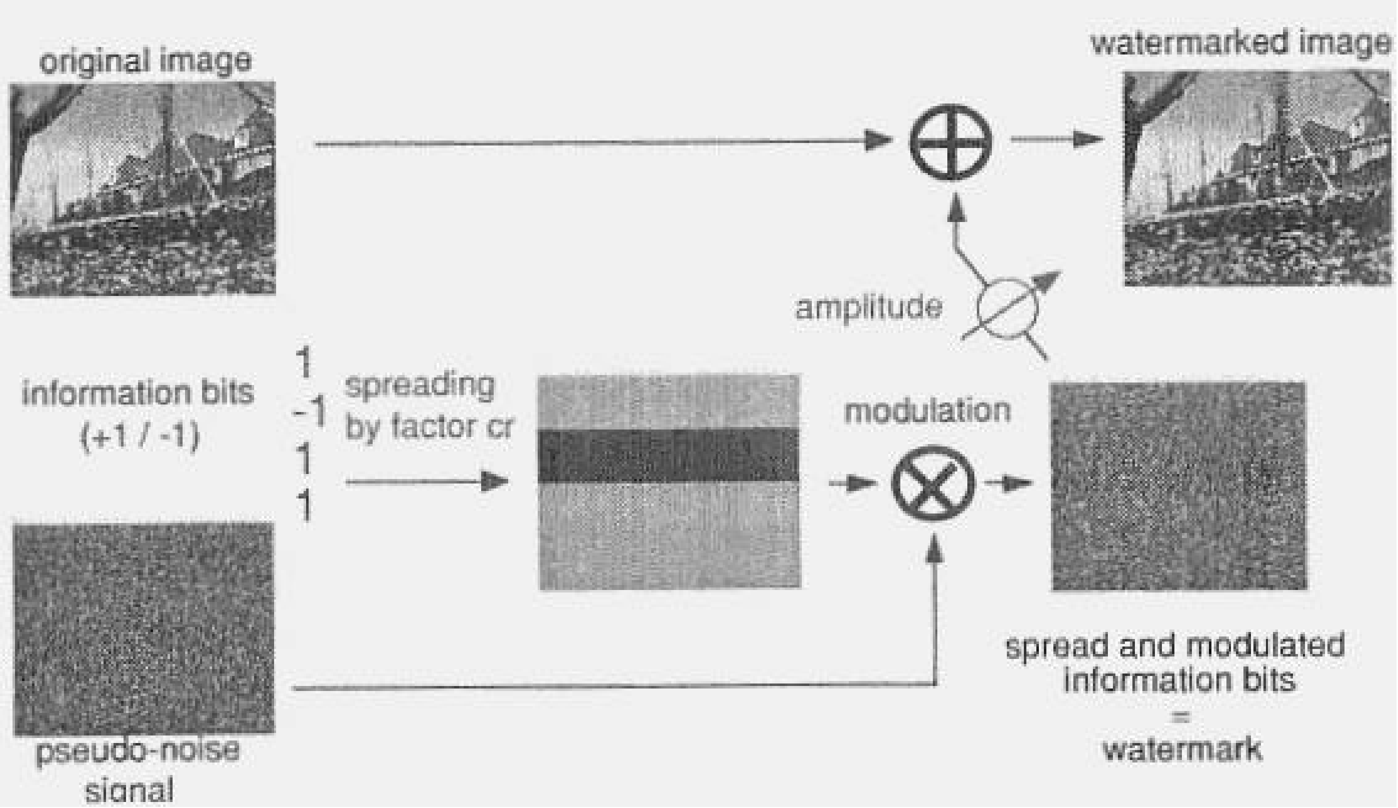
Original Coded Signal	3	+3	-2	-2	0	0	2	0
Watermark			+1			+1		
Drift Compensation Signal				-1			-1	
Watermarked Signal	3	+3	-1	-3	0	1	1	0
Reconstructed Watermarked Signal	3	6	5	2	2	3	4	4

Preservation of Compressed Data Rate

- Watermark signal is often “noisy” pseudo-random number sequence
- “Noisy” sequences are difficult to encode, causing the data rate of the watermarked video to increase, sometimes substantially

- Compressed domain watermark embedders must control the data rate

Hartung’s Technique



Hartung’s Technique

- Bitstream is parsed to obtain encoded DCT coefficients and motion vectors
  - Motion vectors are not watermarked
  - Spread spectrum signal is inserted into the DCT coefficients
- Drift compensation

Hartung's Technique

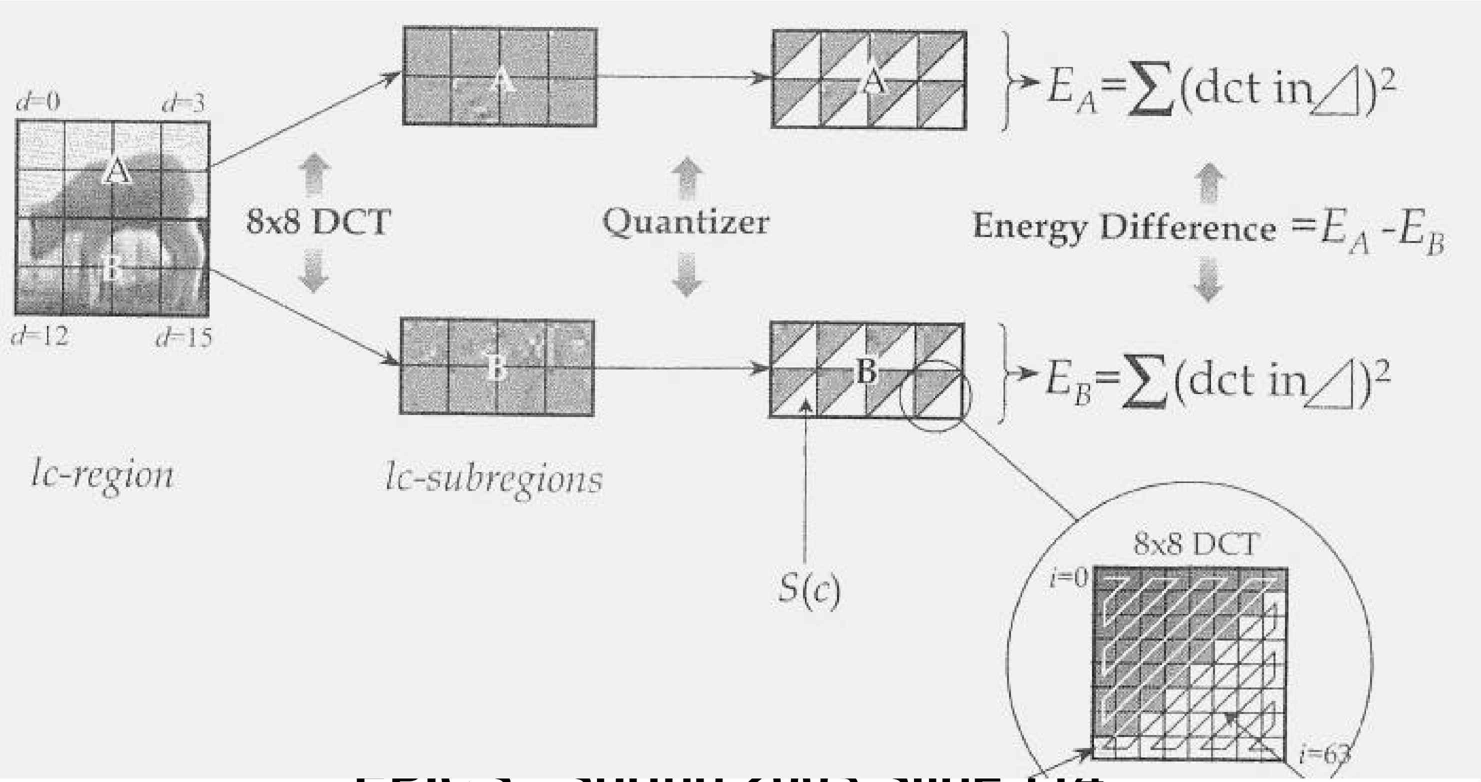
- Not all coefficients are watermarked:
  - Zero coefficients
  - If embedding watermark increases data rate
- Evaluated for high data rate video (4-12 Mbits/s)
- Spread-spectrum watermark fairly robust against signal processing attacks

Hartung's Technique

- Problems:
  - Not many coefficients are actually watermarked because of rate constraints (10% or so)
  - Thus, watermark may be more vulnerable to removal
  - Method of bit-rate control may not be applicable for low-rate video

Langelaar's Technique

- **Watermark key determines block groupings**



d=0
d=3
d=12
d=15
A
B
lc-region
8x8 DCT
lc-subregions
Quantizer
$E_A=\sum(\text{dct in}\triangle)^2$
$E_B=\sum(\text{dct in}\triangle)^2$
Energy Difference $=E_A-E_B$
$S(c)$
8x8 DCT
i=0
i=63

Langelaar's Technique

- Low computational cost
- Original technique only watermarked I-pictures; has been extended for P/B-pictures
  - Not very robust against attack, particularly against transcoding
  - Technique may have problems with low-rate video
- Evaluated using high rate video (4-8 Mbits/sec)

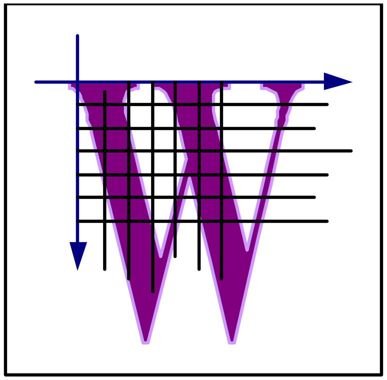
Conclusions

- Overview digital video compression
- Overview issues when watermarking compressed video
- Reviewed two different means for watermarking compressed video

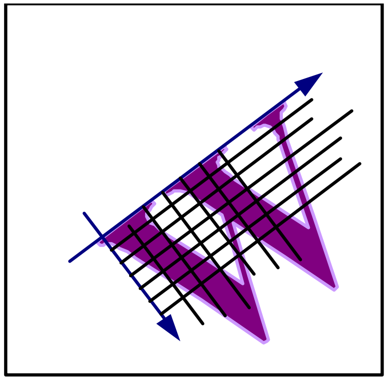
Temporal Synchronization in Video Watermarking

Synchronization

- Synchronization is necessary for reliable detection of many watermarks
- The detector establishes correspondence between the coordinates of the watermarked signal and the watermark

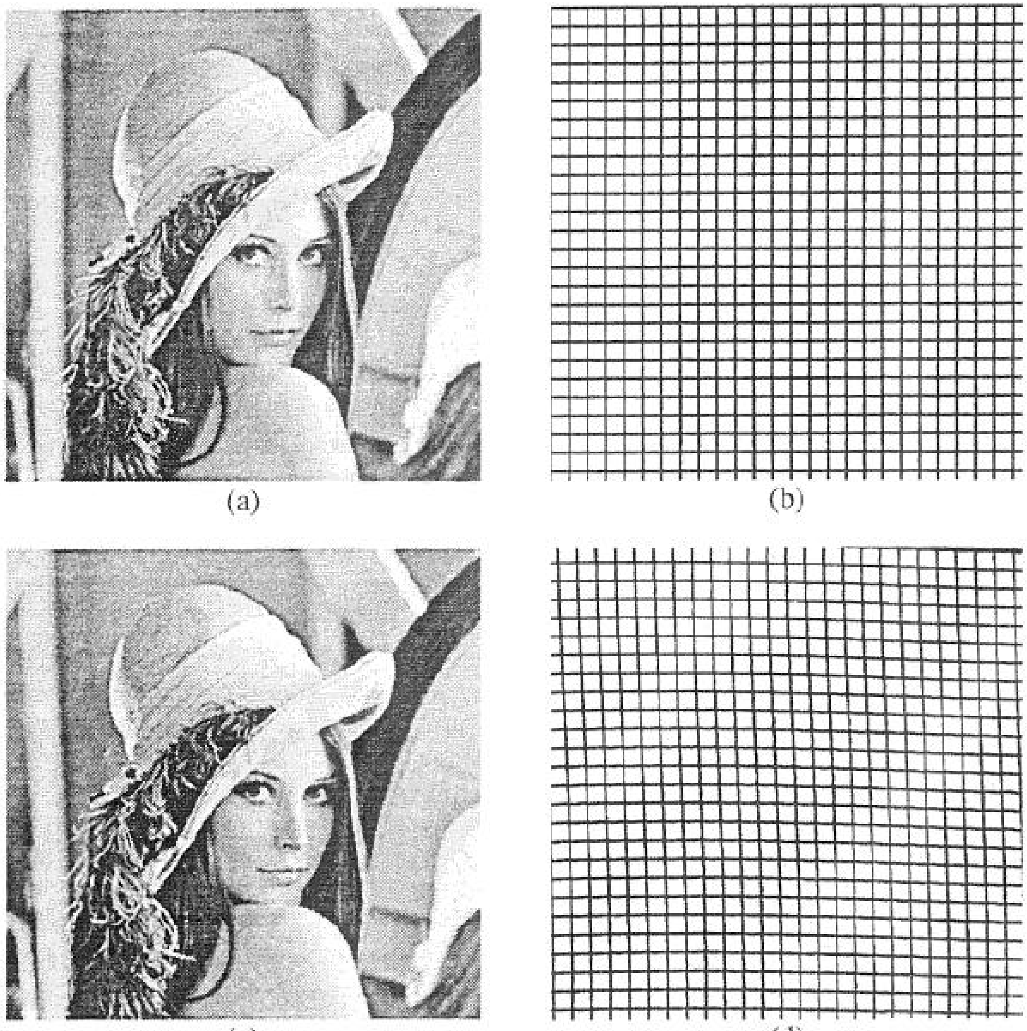


Watermarked



Attacked

Synchronization Attack Example



(a) (b)

(c) (d)

Synchronization

- Spatial synchronization
  - Attacks: Scaling, rotation, translation, warping
- Temporal synchronization
  - Initial synchronization
  - Re-synchronization after bit/decoding errors
  - Attacks: Frame deletion, insertion, transposition, averaging

Synchronization Techniques

- Fast synchronization techniques desired
  - Avoid computationally expensive searches
  - Real-time video applications

- Techniques
  - Sliding correlator
  - Templates

Sliding Correlators

$$\sum_x \sum_y \sum_t W(x, y, t) Y(x - x_0, y - y_0, t - t_0)$$

- Correlate over all spatial and temporal shifts x_0, y_0, t_0
- In general, very expensive search over a large search space

Synchronization Techniques

- Templates
  - Additional signal embedded into video for fast synchronization
  - Disadvantages:
    - Templates must be easily detectable, and thus vulnerable to attack and removal
    - Template embedding adds distortion in the watermarked video, affecting perceptual quality

A Synchronization Framework

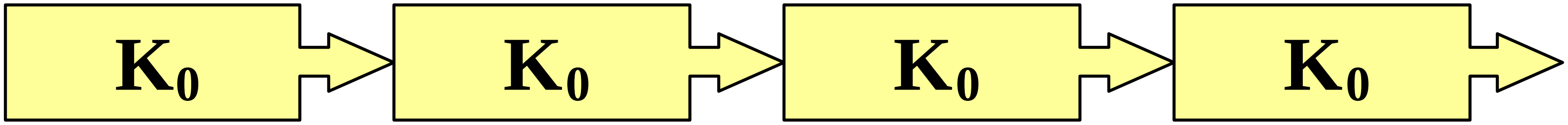
- Assume a symmetric video watermarking technique is used
- Embedder uses key K(t) to create the watermark embedded in frame t
  - If detector can deduce K(t): Synchronization achieved
  - If detector cannot deduce K(t): Synchronization lost
- K(t) often corresponds to watermark signal generator state

Temporal Redundancy

- Temporal redundancy: The degree by which the watermark signal can be deduced given the watermark signal in the past

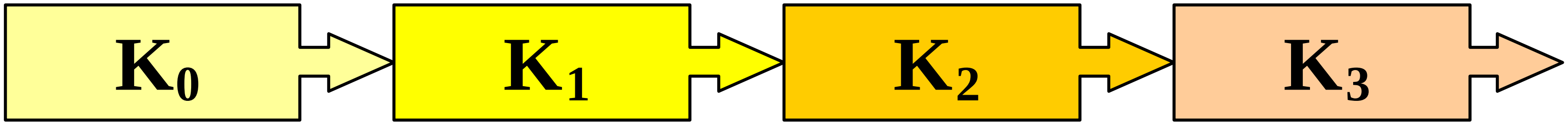
- Synchronization search can be reduced by increasing temporal redundancy of watermark
  - May be a security trade-off, as increased redundancy could make watermark more vulnerable to attack

Temporal Redundancy



- Time-invariant watermark
  - Very high temporal redundancy
  - Temporal synchronization is trivial (not needed)
  - Low security

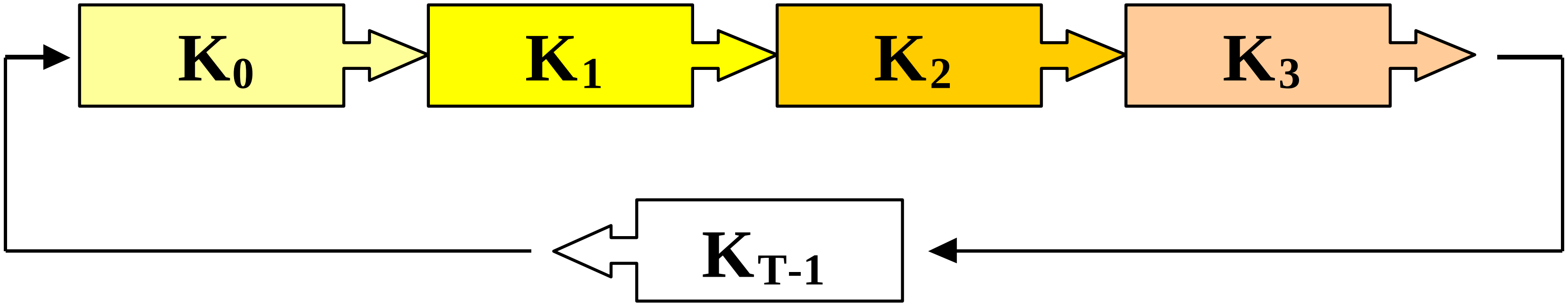
Temporal Redundancy



K_0 K_1 K_2 K_3

- **Independent watermark for each picture**
  - **Many watermarks implicitly use this model**
  - **No temporal redundancy**
  - **Synchronization is difficult or expensive**
  - **High security**

Temporal Redundancy



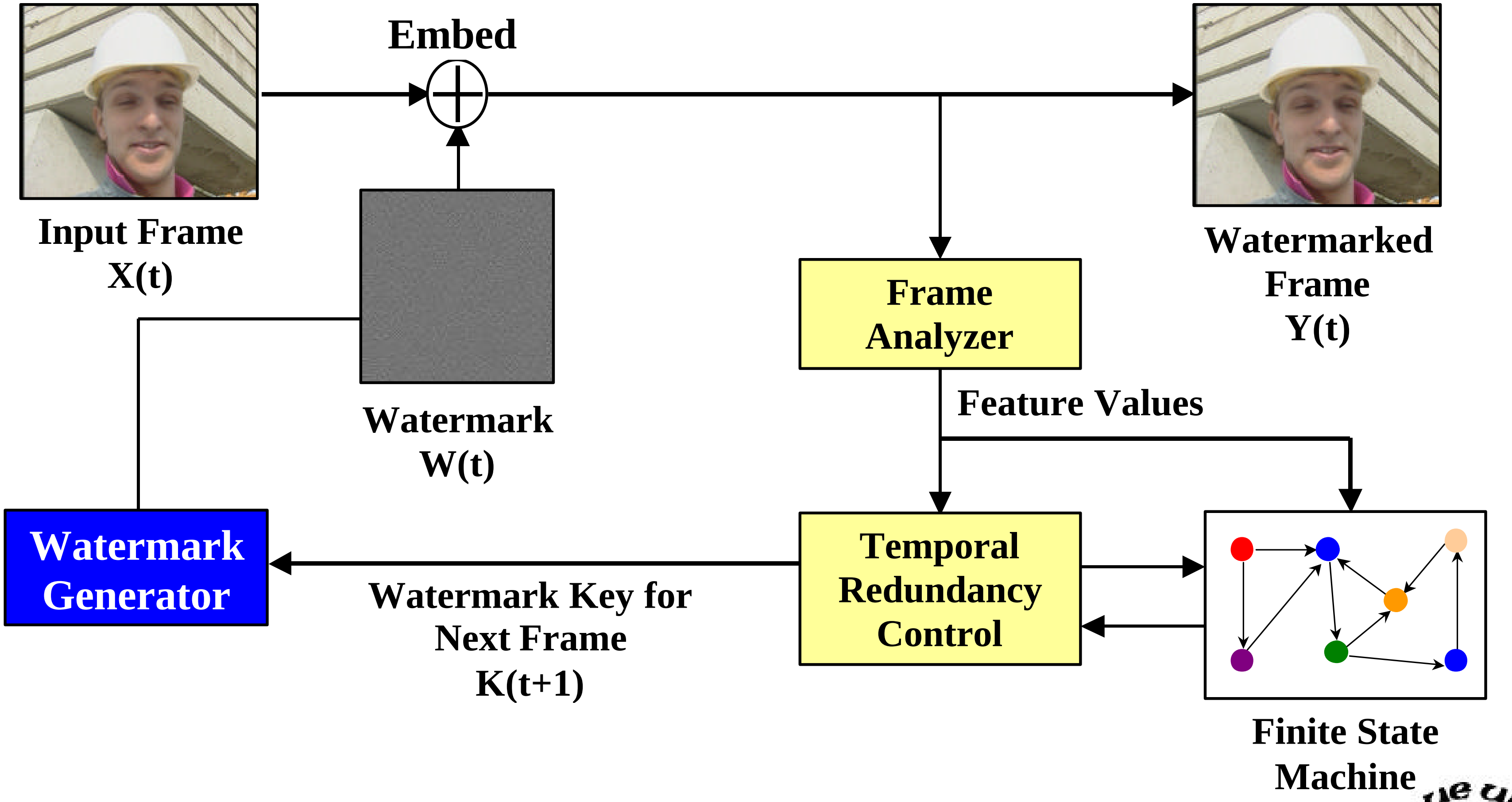
- Repeating (periodic) watermark
  - Synchronization is trivial
  - Security better than time-invariant watermark but the watermark may be deduced by inter-frame correlation

A Protocol for Synchronization

- Exploits temporal redundancy to allow fast synchronization (no templates)
- Time-invariant, periodic, time-independent watermarks are special cases
- Video-dependent

- Embedder: Temporal redundancy is controlled by period and repeat parameters
- Detector: Uses priority queue to perform and maintain synchronization

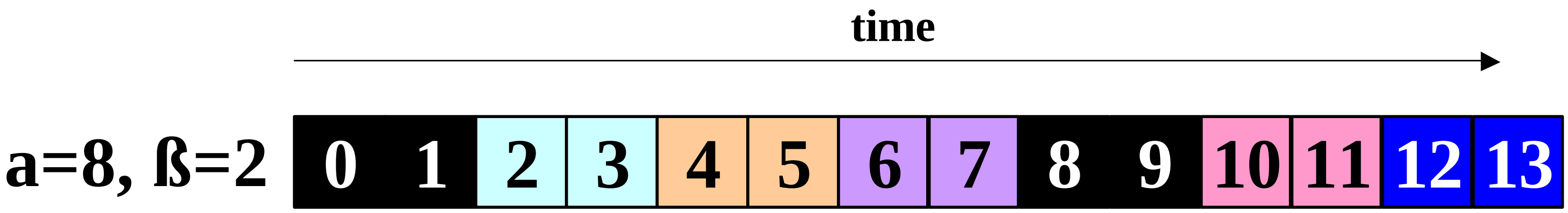
Watermark Embedding Protocol



Temporal Redundancy Control

- a (period) = # of frames watermarked before resetting the state machine to initial state and watermark key to the initial key K_0

- ß (repeat) = # of frame watermarked by identical watermark before watermark key is changed



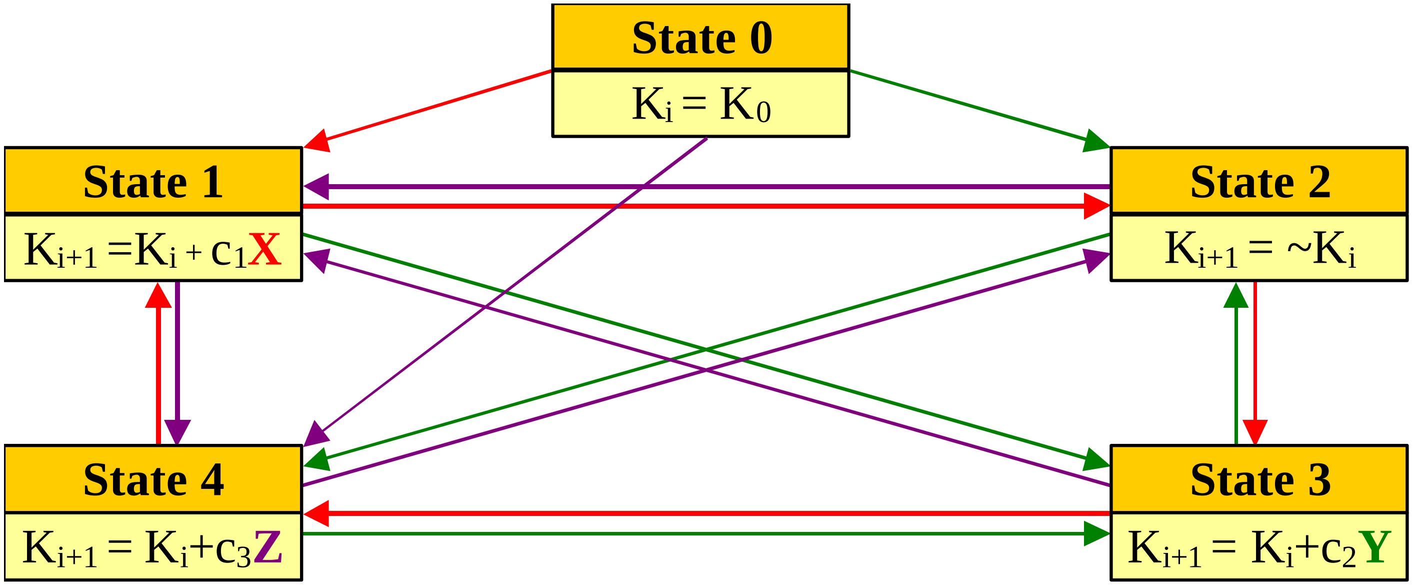
Numbers indicate frame index

Frame Analyzer

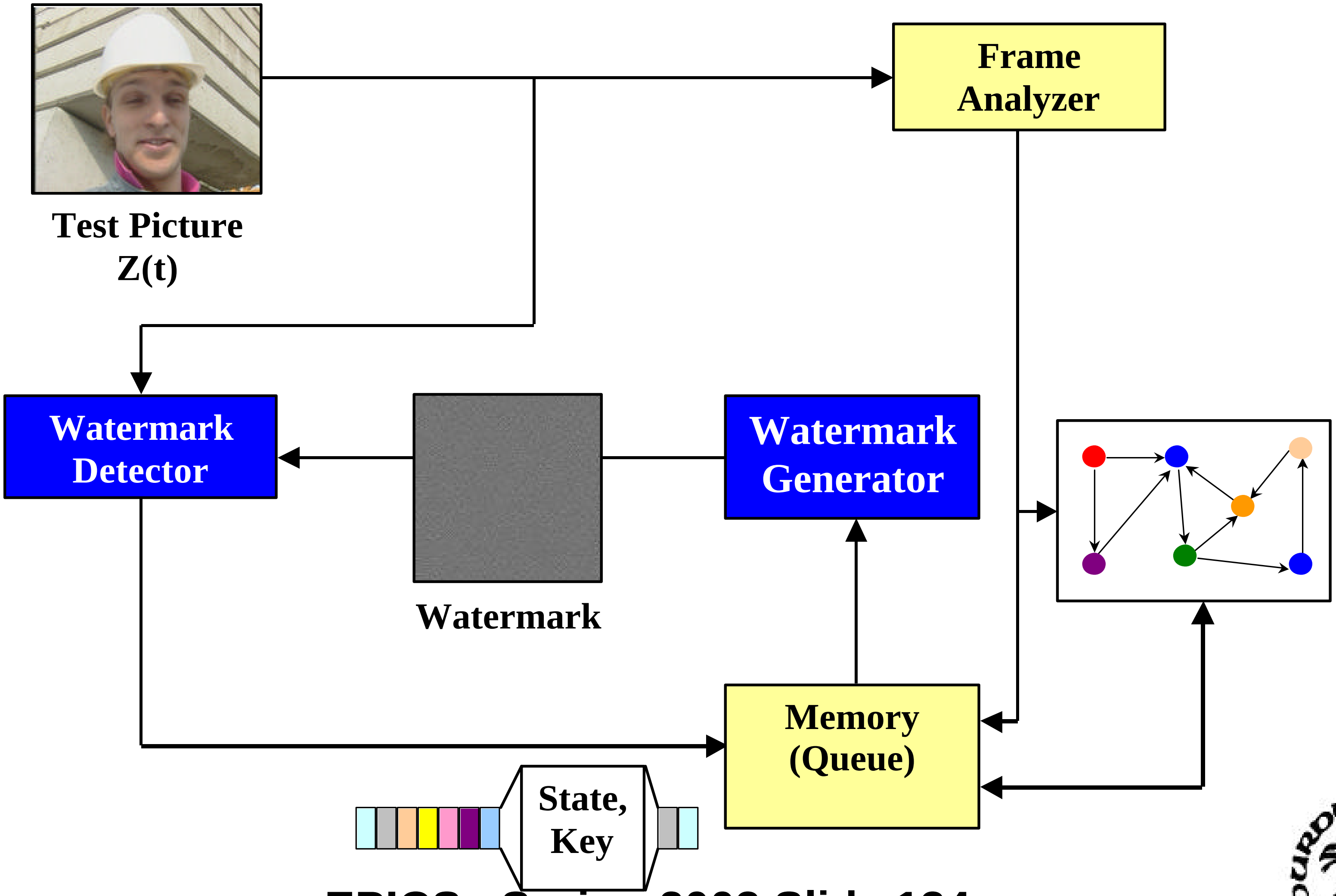
- Analyzes the watermarked frames, output is a vector of feature values
  - Used with the state machine to generate the next watermark when needed
  - Allows the watermark to be video dependent
  - Features should be robust and not change in value unless significant change occurs to image

State Machine

- Each state describes how to generate next watermark K(t+1) from the current watermark K(t)
- The state machine itself has redundancy because possible number of next states from a given state is finite



Watermark Detection Protocol



Watermark Detection Protocol

- Detector does not know a, ß
- The detector tries the following keys:
  - The initial key K_0
  - Every key value stored in the queue

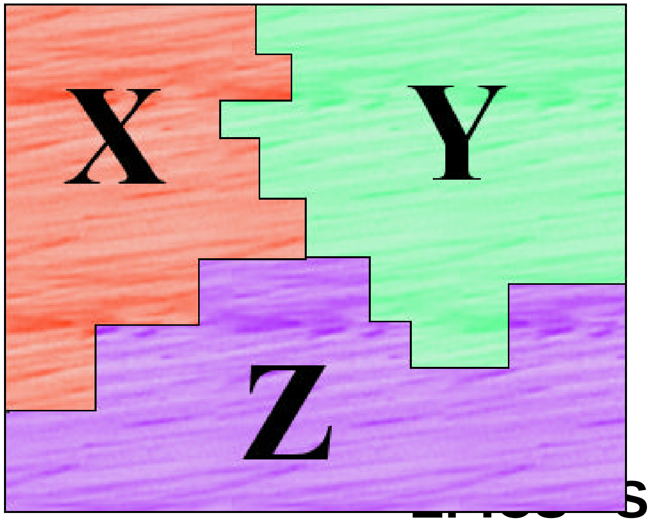
- If no watermark found: Queue not updated
- If watermark found:
  - Move (state,key) to top of queue
  - Use state machine, features to find (next state, next key) and insert into top of queue

Experimental Setup

- **Uncompressed video (4:1:1 352x288 CIF)**
- **Attacks**
  - **Frame dropping**
  - **Frame insertion**
    - **Inserted frames are not watermarked**
  - **Local frame transposition**
  - **Frame averaging**
  - **Combined attack**
- **Ten trials for each attack, average detection results are shown**

Experimental Setup

- **Watermark: Spatial domain Gaussian**
- **Detection: Correlation detector**
- **State Machine: Previously shown**
- **Features:**

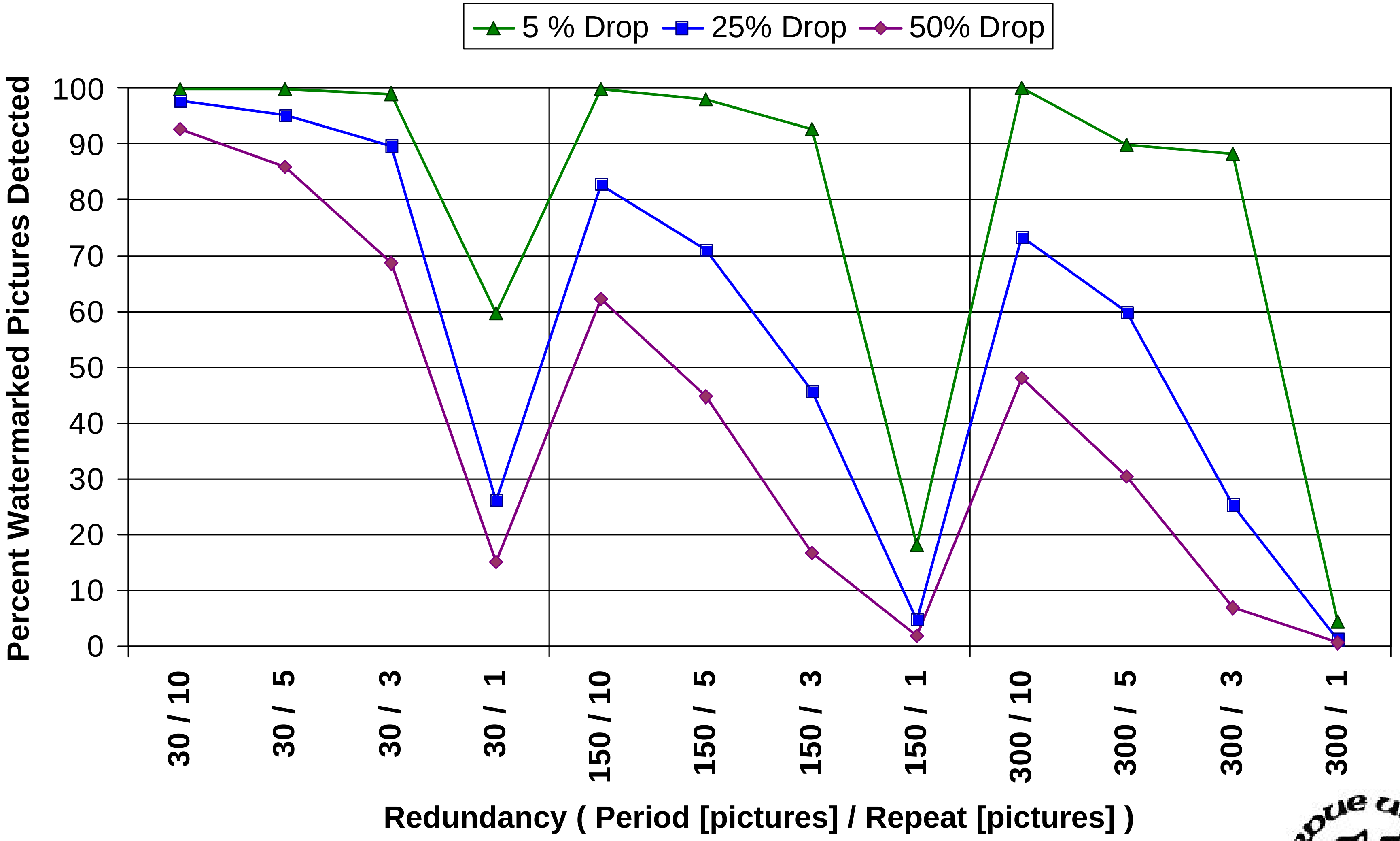


Feature X = Quant[Mean(Pixels in X)]

Feature Y = Quant[Mean(Pixels in Y)]

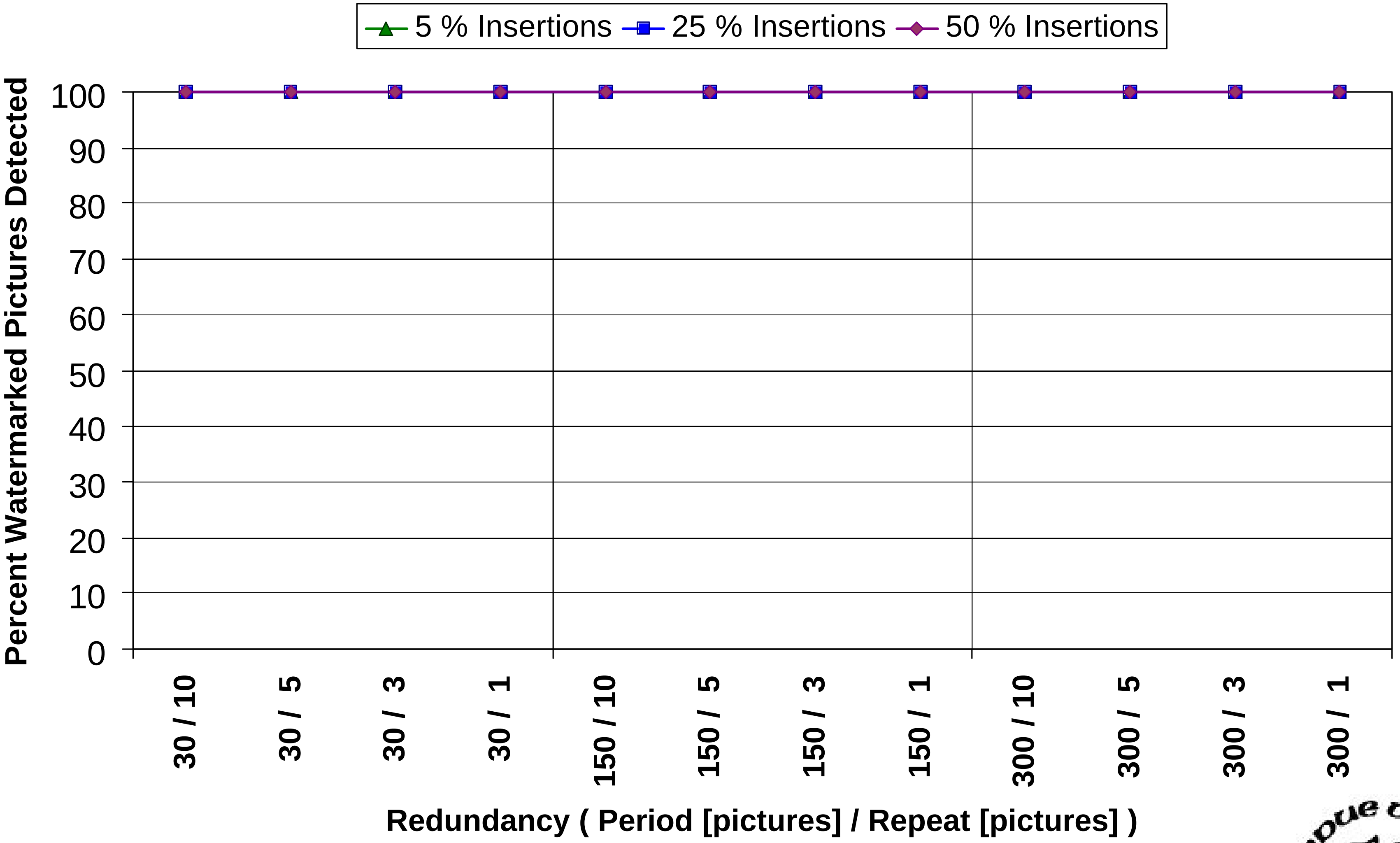
Feature Z = Quant[Mean(Pixels in Z)]

Frame Drop

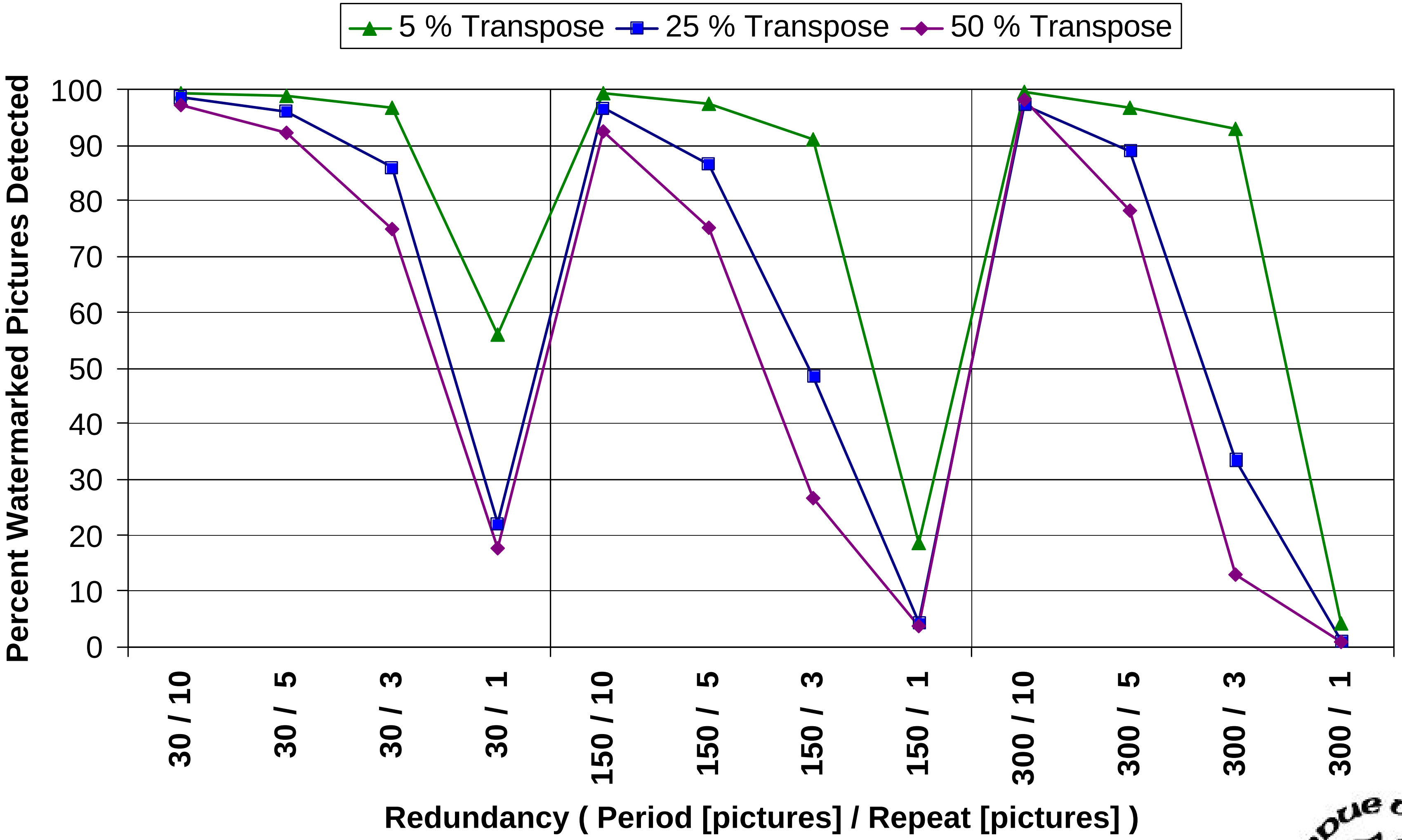


5 % Drop
25% Drop
50% Drop
Percent Watermarked Pictures Detected
100
90
80
70
60
50
40
30
20
10
0
30 / 10
30 / 5
30 / 3
30 / 1
150 / 10
150 / 5
150 / 3
150 / 1
300 / 10
300 / 5
300 / 3
300 / 1
Redundancy (Period [pictures] / Repeat [pictures])

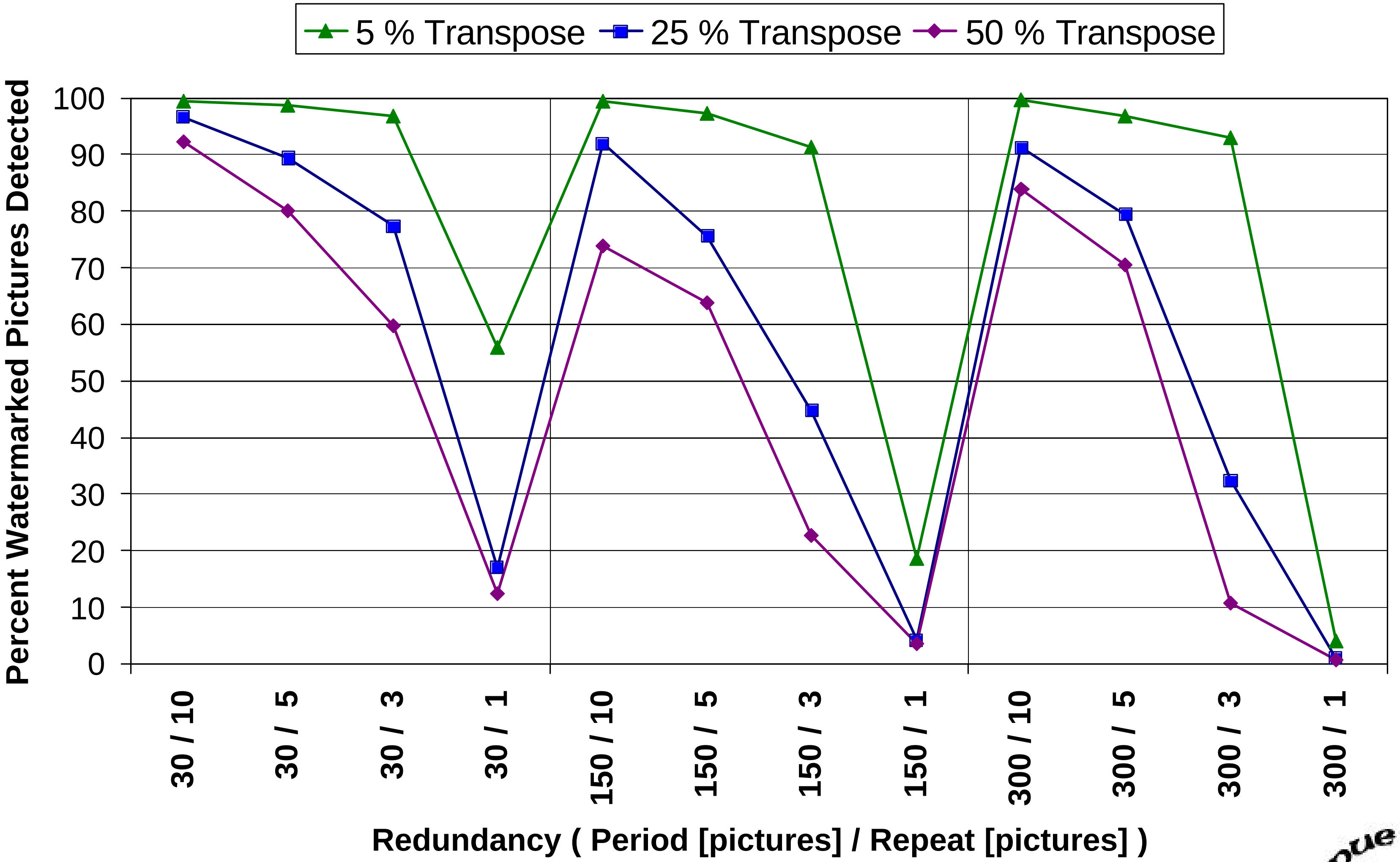
Frame Insertion



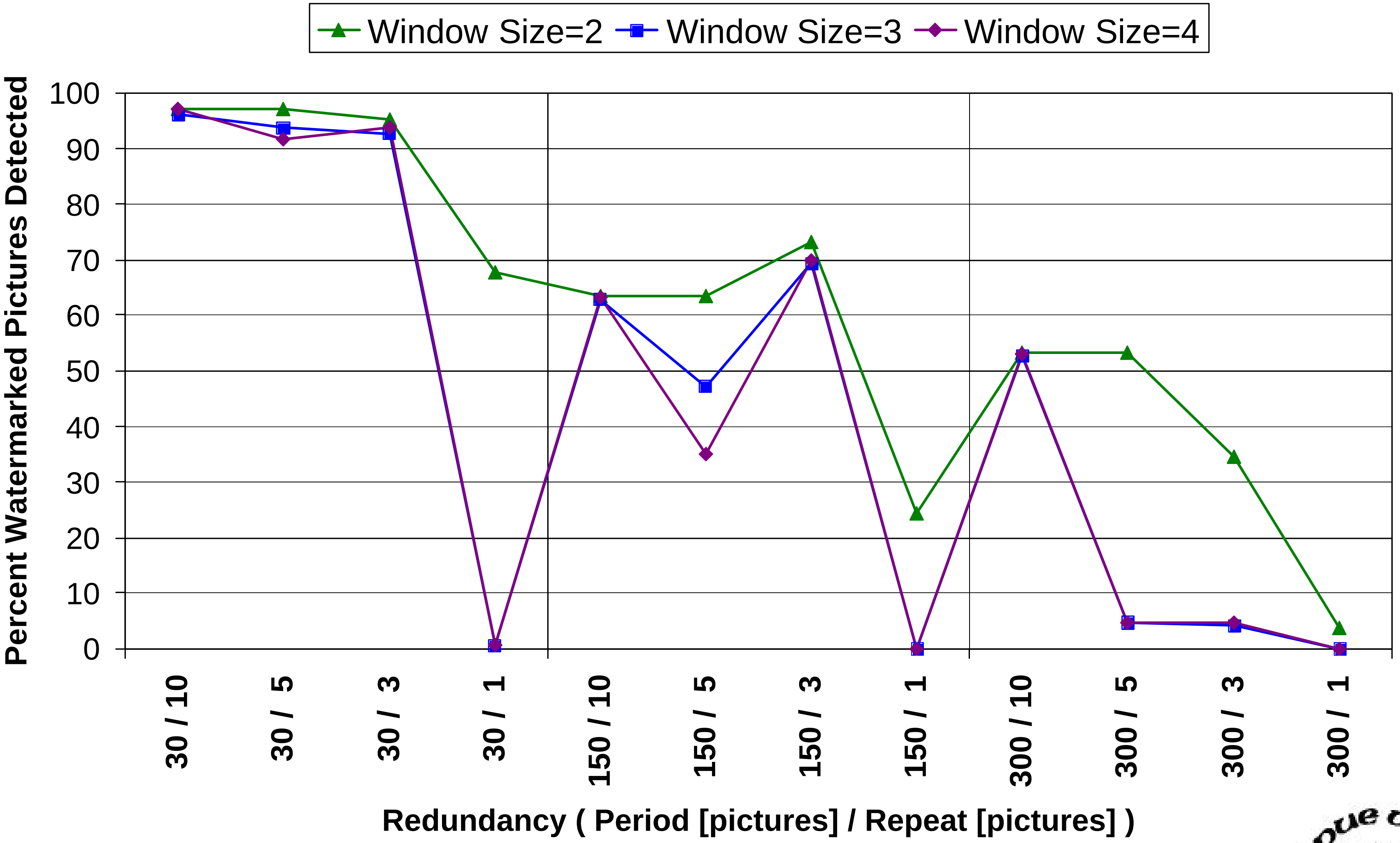
Frame Transposition



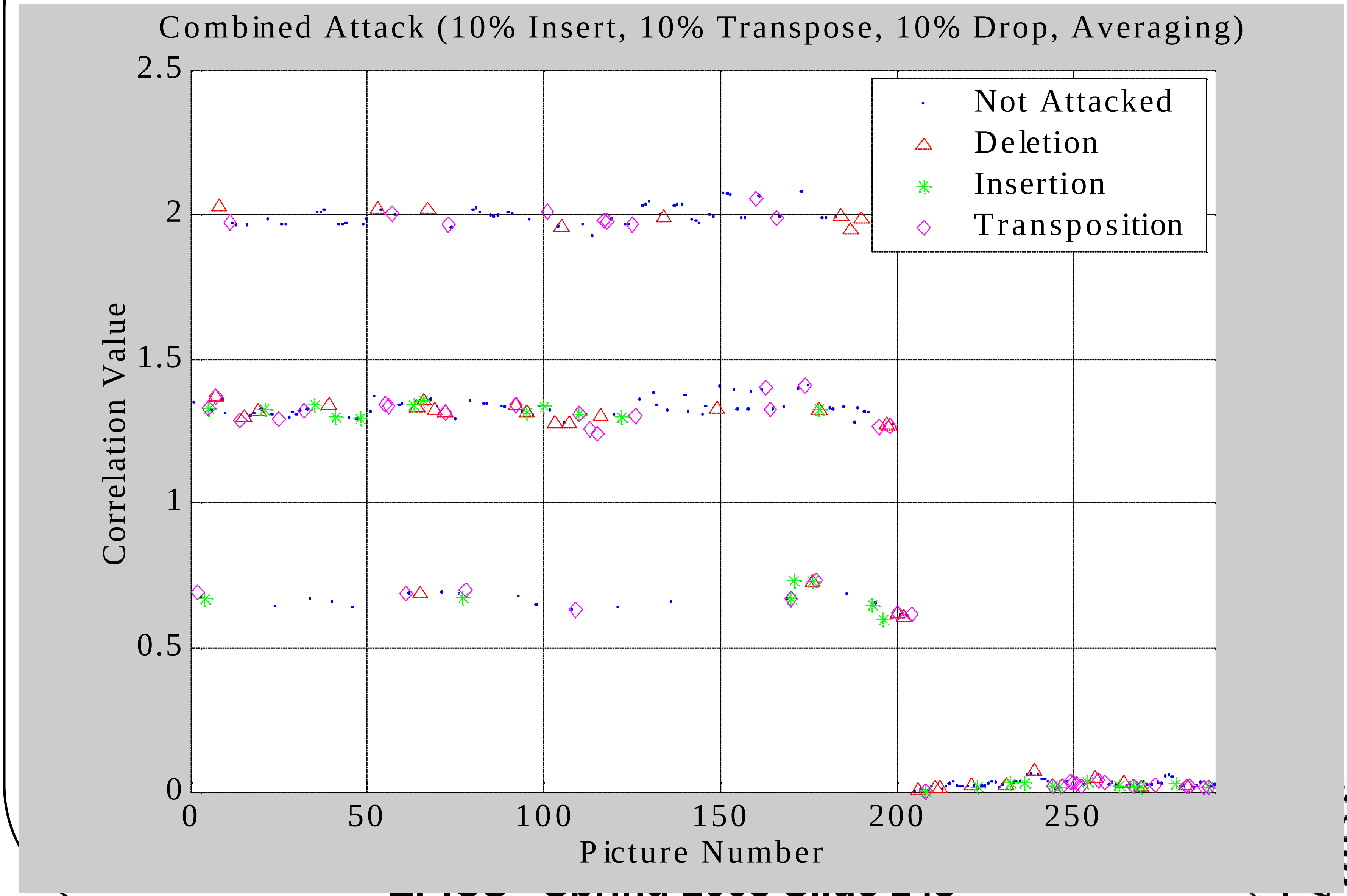
Frame Transposition II



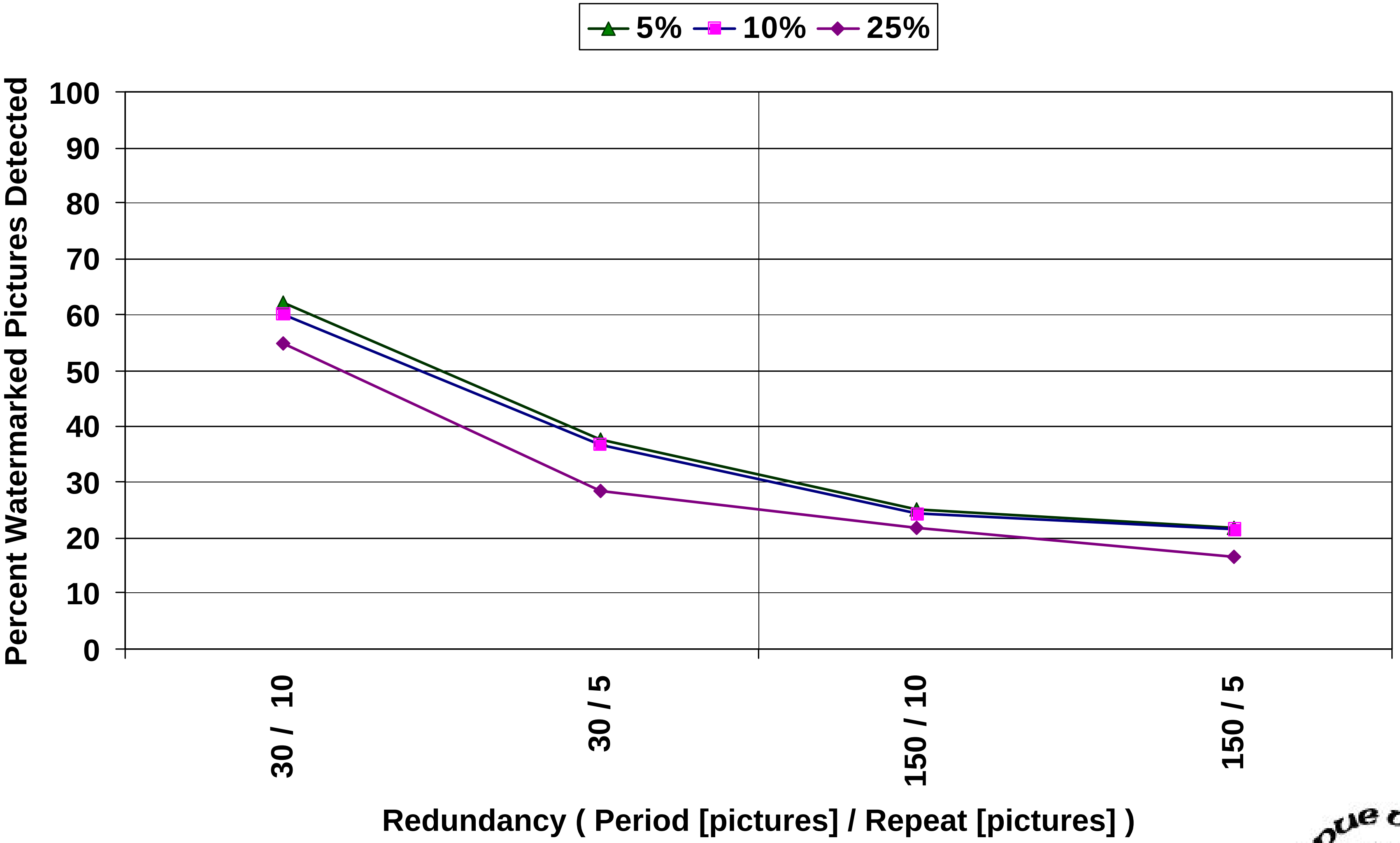
Frame Averaging



Combined Attack



Combined Attack



Improvements

- **Improving security**
  - **Current weakness: Correlation can find frames watermarked with K_0**
  - **Resynchronization watermark does not have to be K_0. Can be video-dependent:**

    $K = f(K_0, X, Y, Z)$

  - **State machine: Ad-hoc design**

Improvements

- Improving robustness
  - Change watermark based on feature values and not on a, ß parameters. Can improve picture deletion and averaging performance
  - Find better or more robust features
  - Find better means of adding temporal redundancy

Conclusions and Future Work

- General method of temporal synchronization by redundancy in the watermark
- A small search is performed instead of using templates, but technique does not preclude the use of templates

- Investigate more appropriate attack models
- Investigate features for compressed video

Current Research Issues

- Theoretical Issues
  - capacity and performance bounds
  - models of the watermarking/detection process
- Robust Watermarks
  - linear vs. nonlinear
  - scaling and other geometric attacks
  - watermarking analog representations of content
  - new detection schemes
  - what should be embedded (watermark structure)

Audio Watermarking

Use of techniques similar to images

- – perceptual models are better developed
- – attacks are different

“Watermarking” Standards

- Data Hiding Subgroup (DHSG) of the Copy Protection Technical Working Group (CPTWG)
  - two groups have proposed systems for watermarking video used in DVD
  - watermark will be second level of security after encryption
  - the watermarks are relatively easy to defeat by difficult to remove
- SDMI (digital audio)
- MPEG-4/MPEG-21

Watermarking: Legal/Political Issues

- Watermarking technologies have not been tested in court
  - is watermarking the “feel good” technology of multimedia?
- Might one be better off just doing timestamping and/or other forms of authentication?
- What does it mean when a watermarking technique survives an attack (verification based on statistical tests)
- Watermarking may always be the secondary security method

Legal Issues

- When one says: “My watermarking withstands the X attack!”
  - – What does it mean? (Has the watermark been damaged?)
  - – It is legally defensible?
  - – Nearly all watermarks require statistical tests for verification

Unauthorized Distribution and Illegal Copies

- Unauthorized Distribution
  - You took my image from my web site!
  - You are selling my image from the CD-ROM you bought from me!
- Who owns it? Þ hash and timestamp
- Is your image the same as mine? (derived work)

Conclusions

- The “secure” multimedia system is evolving
- Simple add-ons will not work (not like the text-based systems)
- Exploit the unique nature of the type of data
- Digital watermarking is crucial to secure networked multimedia systems
- Time stamping is important
- New techniques tolerate changes to images, and are compatible with compression

Conclusions

- Watermarking is still an interesting research area with many interesting problems
  - where will it be useful?
  - will watermarking only be used a second-tier security system?
  - will there be significant theoretical developments?

- Is watermarking the "feel good" technology of multimedia?

Reference

R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proceedings of the IEEE*, Vol. 87, No. 7, July 1999, pp. 1108-1126.

Available at:

ftp://skynet.ecn.purdue.edu/pub/dist/delp/watermark-proceedings/