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

EPICS Spring 2003 Slide 1
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
- Auditing (fingerprinting)
  - Who did what and when?
- Playback Control
- Record Control
- Generation Control
What Do Users Want?

• Time-shifting
• Space-shifting
• Single copy (back ups?)
Digital Communication System

Digital Information Source → Source Encoder → Encrypt → Channel Encoder

Channel

User → Source Decoder → Decrypt → Channel Decoder
Typical Cryptography System: Trusted Users

Source → Insecure Channel → User

Attacker

Is this a pirate device?
Cryptography System: User Not Trusted

Source → Insecure Channel → User

Authentication
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 $\Rightarrow$ watermarking
- Copyright Protection $\Rightarrow$ 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 exist
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 $\Rightarrow$ 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 cryptography in that an illegal user may intercept the message.
Why is Watermarking Important?
Why is Watermarking Important?
Why Watermarking is Important?
Why is Watermarking Important?
Why is Watermarking Important?
Why is Watermarking Important?
Why is Watermarking Important?
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
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 ⇒ 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

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

- \( X_{u,v} \) = original transform coefficient
- \( J_{u,v} \) = JND value for particular coefficient
- \( W_{u,v} \) = watermark element
- \( Y_{u,v} \) = marked coefficient
Image Adaptive Watermarks (DCT)
Image Adaptive Watermarks (DCT)
Image Adaptive Watermarks (DCT)
IBM ATTACK

• Description
  – $X \sim$ original, $Y_I \sim$ image marked with $W_I$
    \[ Y_I = X + W_1 \]
  – Create counterfeit original, $X_F$
    \[ X_F = Y_I - W_2 \]
  – $Y_I$ now appears to be a marked version of $X_F$
    \[ \Rightarrow Y_I = X_F + W_2 \]
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

```
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
```

RGB values of pixel: (240 216 206)
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

<table>
<thead>
<tr>
<th>Index</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>84</td>
<td>2</td>
</tr>
<tr>
<td>239</td>
<td>160</td>
<td>2</td>
<td>233</td>
</tr>
<tr>
<td>240</td>
<td>240</td>
<td>216</td>
<td>206</td>
</tr>
<tr>
<td>241</td>
<td>239</td>
<td>200</td>
<td>102</td>
</tr>
<tr>
<td>254</td>
<td>255</td>
<td>255</td>
<td>255</td>
</tr>
<tr>
<td>255</td>
<td>128</td>
<td>127</td>
<td>124</td>
</tr>
</tbody>
</table>

Color Lookup Table

Pixel value == index in color table

RGB values of pixel
Digital Image Steganography

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

Cover Image → Steganographic Encoder (S-Tools) → Stego Image

Secret Message

Stego Key
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 \[\rightarrow\] Steganographic Decoder (S-Tools) \[\rightarrow\] Extracted Message

Stego Key

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)

EPICS  Spring 2003 Slide 59
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
- Stego Key
- Message (Files)

1. Cover Image
2. Image size
3. Message Processing
4. Message Embedding
5. 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.

These bits are changed to that of the encrypted message.

EPICS  Spring 2003 Slide 68
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 reduces colors in image
- Dithering can improve visual quality
- 8-bit embedding inserts encrypted message
Color Reduction

- Median Cut Algorithm described in

  - 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”

EPICS  Spring 2003 Slide 72
Dithering

• Dithering process improves visual quality by reducing “contouring” effects
  

  – 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 avoid suspicion

Color Index

Selects 1 of the 32 colors used in the reduced color image

MSB Bits LSB

Set to match 3 bits of the encrypted message

Palette

RGB values of pixel

EPICS Spring 2003 Slide 75
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

EPICS  Spring 2003 Slide 79
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

Steganalysis References

“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:

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

• 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

Beginning of stream (Decoded First)
End of stream (Decoded Later)
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

<table>
<thead>
<tr>
<th>Original Signal</th>
<th>3</th>
<th>6</th>
<th>4</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>4</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predictive (Differential) Coding</td>
<td>3</td>
<td>+3</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Residual

<table>
<thead>
<tr>
<th>Original Coded Signal</th>
<th>3</th>
<th>+3</th>
<th>-2</th>
<th>-2</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watermark</td>
<td></td>
<td>+1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Watermarked Signal</td>
<td>3</td>
<td>+3</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Reconstructed Watermarked Signal</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
Drift Compensation

- Must compensate for drift error during watermarking

<table>
<thead>
<tr>
<th>Original Signal</th>
<th>3</th>
<th>6</th>
<th>4</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>4</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Coded Signal</td>
<td>3</td>
<td>+3</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Watermark</td>
<td></td>
<td>+1</td>
<td></td>
<td></td>
<td>+1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drift Compensation Signal</td>
<td></td>
<td>-1</td>
<td></td>
<td></td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Watermarked Signal</td>
<td>3</td>
<td>+3</td>
<td>-1</td>
<td>-3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Reconstructed Watermarked Signal</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
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

- Original image
- Watermarked image
- Information bits (+1/-1)
- Pseudo-noise signal
- Spreading by factor cr
- Modulation
- Spread and modulated information bits = watermark
- Amplitude
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
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
Synchronization Attack Example
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

- 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

Input Frame $X(t)$

Watermark Generator

Watermark $W(t)$

Embed

Watermarked Frame $Y(t)$

Feature Values

Temporal Redundancy Control

Finite State Machine

Watermark Key for Next Frame $K(t+1)$
Temporal Redundancy Control

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

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

Numbers indicate frame index

EPICS  Spring 2003 Slide 131
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 $a$, $\beta$
- 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

Percent Watermarked Pictures Detected

Redundancy (Period [pictures] / Repeat [pictures])

EPICS Spring 2003 Slide 138
Frame Insertion

Redundancy (Period [pictures] / Repeat [pictures])

Percent Watermarked Pictures Detected

- 5% Insertions
- 25% Insertions
- 50% Insertions

EPICS  Spring 2003 Slide 139
Frame Transposition

Redundancy (Period [pictures] / Repeat [pictures])

Percent Watermarked Pictures Detected

5 % Transpose  25 % Transpose  50 % Transpose

EPICS  Spring 2003 Slide 140
Frame Transposition II

Percent Watermarked Pictures Detected

Redundancy ( Period [pictures] / Repeat [pictures] )
Frame Averaging

Percent Watermarked Pictures Detected

Redundancy (Period [pictures] / Repeat [pictures])

Window Size=2
Window Size=3
Window Size=4
Combined Attack

Combined Attack (10% Insert, 10% Transpose, 10% Drop, Averaging)

- Not Attacked
- Deletion
- Insertion
- Transposition
Combined Attack

Percent Watermarked Pictures Detected

Redundancy (Period [pictures] / Repeat [pictures])

EPICS  Spring 2003 Slide 144
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 \( a, \beta \) 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? \(\Rightarrow\) 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


Available at:

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