EE 6886: Topics in Signal Processing
-- Multimedia Security System

Lecture 6: Multimedia Authentication

Ching-Yung Lin
Dept. of Electrical Engineering
Columbia University, New York, NY 10027

Outline -- Introduction

- Multimedia Security :
  - Multimedia Standards – Ubiquitous MM
  - Encryption – Confidential MM
  - Watermarking – Uninfringible
  - Authentication – Trustworthy MM
- Security Applications of Multimedia:
  - Audio-Visual Person Identification – Access Control, Identifying Suspects
  - Surveillance Applications – Abnormality Detection
  - Media Sensor Networks – Event Understanding, Information Aggregation

Authentication

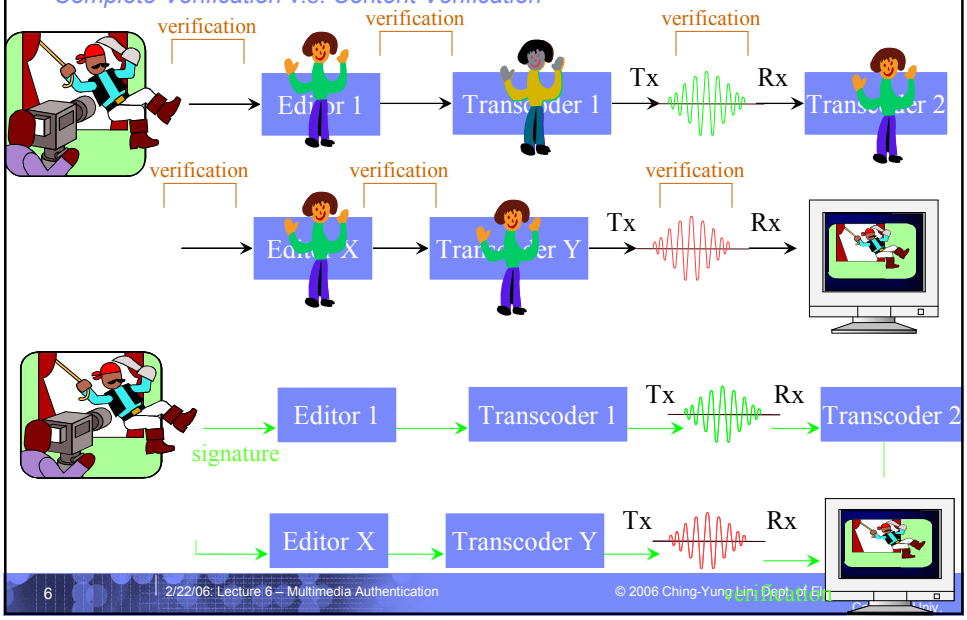
Why we need authentication?

A Historical Review

- Digital Signature, Diffie and Hellman (1976).
  1. Depends on the content of data;
  2. Generates by some secret information only known to the signer.
     - It can be used to verify the data integrity which is endorsed by the signer.
- Trustworthy Digital Camera, Friedman (1993)
  - provide some credibility of the reality

Watermarking ?

Multimedia Authentication Objectives:
Complete Verification v.s. *Content Verification*



Multimedia Authentication Sources: *Raw Data* v.s. *Compressed Data*

- Is raw data available?
- *Completeness*:
  - Raw Data + Complete Authentication.
- *Compression:*
  - Raw Data + Content Authentication or Compressed Data + Complete Authentication?
- *Robustness*:
  - Compressed Data + Content Authentication.

Multimedia Authentication Methods: *Watermarking* v.s. *Digital Signature*

- Watermarking:
  - embedding digital producer identification label, or
  - embedding content-based codes generated by producer specific rule.

- hash codes or traditional digital signature?

Multimedia Authentication Methods: *Watermarking* v.s. *Digital Signature*

- Watermarking for complete authentication of raw data:
  - *convenient* and *no sensible degradation*.

- Watermarking for complete authentication of compressed data:

Multimedia Authentication Methods: *Watermarking* v.s. *Digital Signature*

- Watermarking for Content Authentication
  - proposed methods were all applied on raw data;
  - failed to distinguish compression from other manipulations;
  - the probability of false alarm and the probability of miss can not be achieved simultaneously;
- Robust Digital Signature
  - saved at the header or as an external file;
  - digital signature remain intact;
  - a better prospect for robustness.

Multimedia Authentication: six requirements

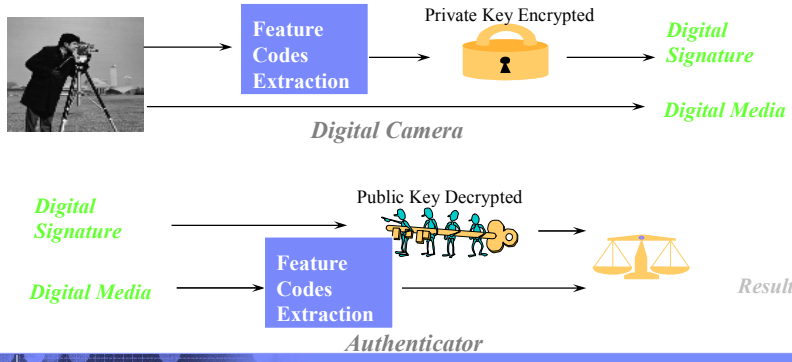
- Sensitivity: crop-and-replacement, others.
- Robustness: lossy compression, content-preserving manipulation.
- Security: non-forged signature, non-transparent mechanism.
- Portability: watermarking v.s. digital signature.
- Localization: detect changed area.
- Recovery: approximation of the original in the changed area.

Some Examples of Watermarking for Robust/Content Authentication

- Checksum and LSB: Wolfgang and Delp, 1996.
  -- Can localize manipulation and some robustness to filtering, but may have a lot of false alarm by using the checksum
- Error Measurement: Zhu, Swanson, Tewfik, 1996.
  -- Measuring errors between the watermarked and the manipulated image. But, may not get a static frequency/spatial masking.
- Spread Spectrum: Fridrich 1998.
  -- Using 64x64 block. Robust to manipulation. But, may not distinguish JPEG and other manipulations, and may miss small area manipulations.

(Robust) Digital Signature

- Digital Signature, Diffie and Hellman (1976).
  - Verify the data integrity which is endorsed by the signer.
- Trustworthy Digital Camera, Friedman (1993).
  - Non-Repudiation Signature to prove *Reality*
- Content-Based Digital Signature, Schneider, Chang (1996)
  - Using content-related feature codes instead of the hash values.
- Robust Digital Signature, Lin and Change (1997).
  - Digital Signature which distinguish content-preserving operations from malicious attacks



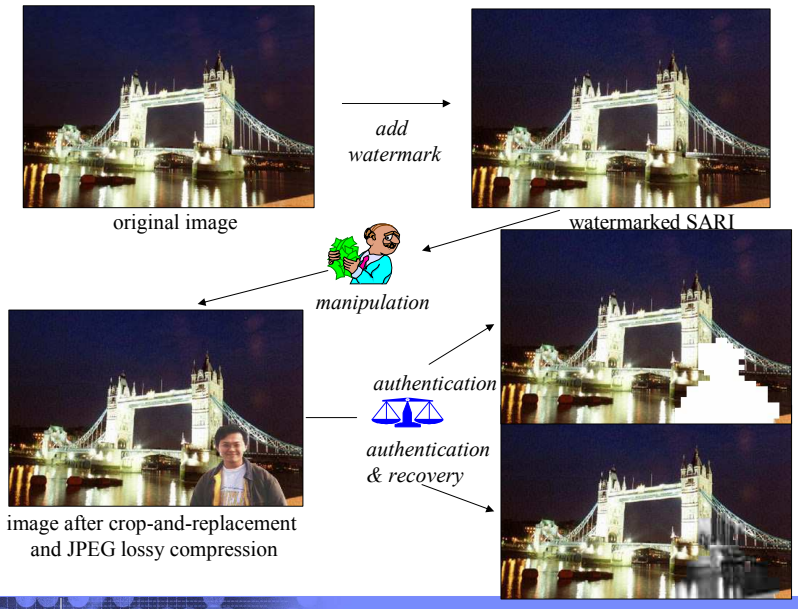
Public Key Cryptography

- RSA: Inventors (Rivest, Shamir, and Adleman).
- The key length is variable. The more commonly used key length for RSA is 512 bits.
- The block size in RSA is also variable.
- Procedure:
  - Choose two large primes p and q (probably around 256 bits each).
  - Multiply them together, and call the result n. (then the factors p and q will remain secret).
  - To generate the public key, choose a number e that is relatively prime to $\phi(n)$. Since you know p and q, you know $\phi(n)$, which is (p-1)(q-1). Your public key is <e, n>.
  - To generate the private key, find the number d that is the multiplicative inverse of e mod $\phi(n)$. <d, n> is your private key.
  - To encrypt a message m (<n), someone using your public key should compute ciphertext $c = m^e \bmod n$.
  - To decrypt, using your private key to compute $m = c^d \bmod n$.

Message Digest Standard (MD5)

- In MD5, the message is processed in 512-bit blocks (sixteen 32-bit words). The message digest is a 128-bit quantity (four 32-bit words).
- Each stage consists of computing a function based on the 512-bit message chunk and the message digest.
- There are four passes, each of which deals with 16 message words. 64 32-bit constants used in MD5.
- Procedures:
  - Message Padding
  - Pass 1: selection function – takes three 32-bit words x,y, and z, and produces an output 32-bit word. If the n-th bit of x is a 1 it selects the n-th bit of y for the n-th bit of the output.
  - Pass 2: majority function – the n-th bit of the output is a 1 iff at least two of the three input words' n-th bits are a 1.
  - Pass 3: XOR functions of x, y, and z.
  - Pass 4: y XOR with (bitwise or of x and the complement of z)

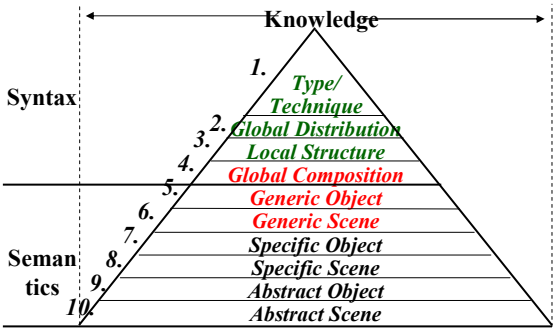
Self Authentication-and-Recovery Images (SARI)



Properties of the SARI System

- Proposal -- *A Semi-Fragile Watermarking Method:*
  - Accepts JPEG lossy compression to a pre-determined lowest quality factor, => *guaranteed*
  - Accepts small noises, => *probabilistic*
  - Rejects crop-and-replacement process, => *probabilistic*
  - Detect the manipulated area,
  - Recover approximate values in the changed area.
- Utilizing two invariant properties of JPEG
  - Consistent relationship between coefficients (Lin and Chang, 1997),
  - Exactly reconstructable coefficients.

What are Content-Related Feature Codes?



Conceptual Framework for Indexing Visual Information
By A. James and S.-F. Chang (2000)

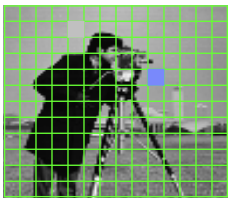
- Syntax Level:
  - E.g.: histogram, texture, shape of objects, color
- Semantic Level:
  - E.g.: man, woman, peace

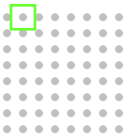
- Our solution: using DCT coefficient relationships to generate authentication bits

➔ *sensitive to attack and immune to DCT-based compression, structured, flexible, scalable,, secure,*

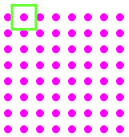
Invariance of DCT-based Lossy Compression

Fp(i,j) : The original DCT coefficient at the position (i,j) of block p
Fp'(i,j) : The reconstructed DCT coefficient of Fp(I,j) at decoding
= Integer Rounding(Fp(i,j) / Qp(i,j)) • Qp(i,j)





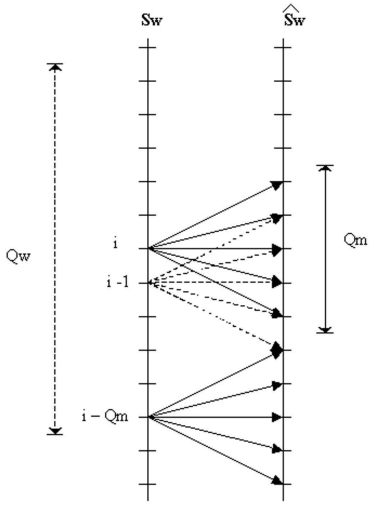
DCT values of block ***p***



DCT values of block ***q***

- Theorem 1:
  - If Fp (i,j) > Fq (i,j) then Fp'(i,j) >= Fq'(i,j)
  - If Fp (i,j) < Fq (i,j) then Fp'(i,j) <= Fq'(i,j)
  - If Fp (i,j) = Fq (i,j) then Fp'(i,j) = Fq'(i,j)

Adjacency-Reducing Mapping of Discrete Values given Bounded Noises



Adjacency-reducing mapping
two input nodes are adjacent if there is a common output node which can be caused by either of these two.
-- Shannon *The zero-error capacity of a noisy channel*, Trans. on IT, 1956)

$$C(Q_w, Q_m) = \log_2 (\lfloor Q_w / Q_m \rfloor + 1) \text{ bits}$$

➔ *A bound for private/public watermarking*

Watermarking using Exactly Reconstructable DCT Coefficients

Fp(i,j) : The original DCT coefficient at the position (i,j) of block p

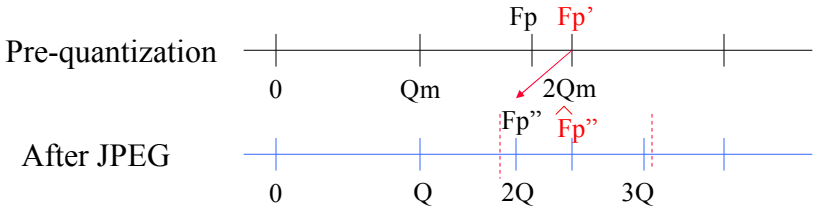
Fp'(i,j) : The pre-quantized DCT coefficient by Qm(i,j)

Fp"(i,j) : The quantized result of Fp'(i,j) after lossy compression using Q(i,j).

- Theorem 2: For all $Q(i,j) \leq Qm(i,j)$

  Fp"(i,j) ≡ Integer Rounding[Fp"(i,j) / Qm(i,j)] Qm(i,j)

  = Fp'(i,j)



capacity for watermark embedding.

Produce reconstructable DCT coefficients surviving JPEG compression

- Setting Qm

Q_{50} : (a) luminance, (b) chrominance

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(a)

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

(b)

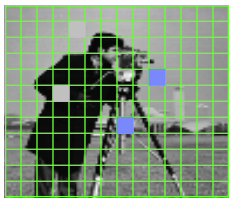
Q_{qf} = Integer Round [Q_{50} * q]

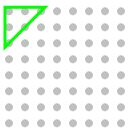
*where q=(2-qf*0.02) if qf ≥ 50*

q = 50/qf, if qf<50

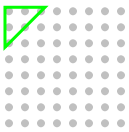
- Integral DCT and IDCT: *methods used in commercial software.*
- Iteration: *extracted information match watermarked image.*
- Using compressed bitstream

Embed Authentication Bits and Recovery Bits



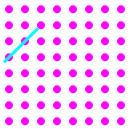


DCT values of block ***p1***

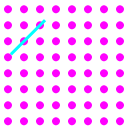


DCT values of block ***p2***

- Divide coefficients in each block into three areas:
  - Area 1: for generating feature codes
  - Area 2: for embedding feature codes and recovery bits
  - Area 3: no-information



DCT values of block ***q1***

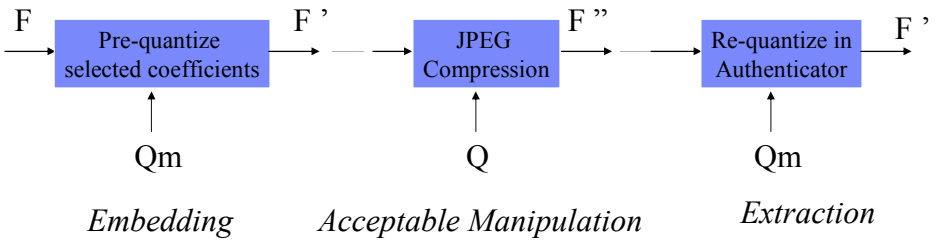


DCT values of block ***q2***

Embedding information bits

- Pre-quantizing a DCT coefficient can get one exactly reconstructable coefficient.

  => Minimal modification: using the LSB of a pre-quantized value to represent 1 bit.

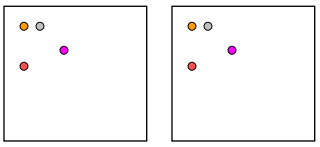


Generation of authentication bits

- Define a secret mapping function ***W*** , s.t.,

$$q = W(p)$$

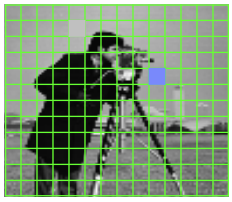
- For each block pair, compare the DCT values at the ***b*** low frequency positions or secret selected positions.

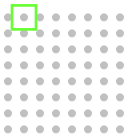


Block p Block $W(p)$

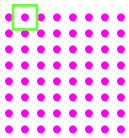
If $Fp(i,j) - Fq(i,j) >= 0$
then $Z_{(p,i,j)} = 1$
If $Fp(i,j) - Fq(i,j) < 0$
then $Z_{(p,i,j)} = 0$

More sophisticated embedding strategy





DCT values of block p



DCT values of block q

- Using XOR of two coefficients' LSBs to embed 1-bit.
  - Larger changes on non-zero coefficient
  - Larger changes with smaller distortion

- Advantage: Improvement on visual quality of watermarked image

Generation of Recovery Bits

Original Image

Scaled down
Quantization,
Huffman Coding,
QF=25

Image constructed
from recovery bits

Security in SARI

- Generate mapping functions based on
  - Fixed mapping generation hardware/software
  - Embedder ID, Time Stamp, Random Seed
- Embed "information bits"

  Public Key $_{\text{authenticator}}$ { Embedder ID +

  Private Key $_{\text{embedder}}$ { Time Stamp + Random Seed + Hash { auth bits} } }

SARI Performance

- For JPEG:

$$P_{\text{False Alarm}} = 0$$

- If 6 authentication bits are used for each block pair,

$$P_{\text{Success Attack}} = 2^{-9 \times N}$$

where N is the number of manipulated blocks.

SARI has achieved the six requirements for multimedia authentication

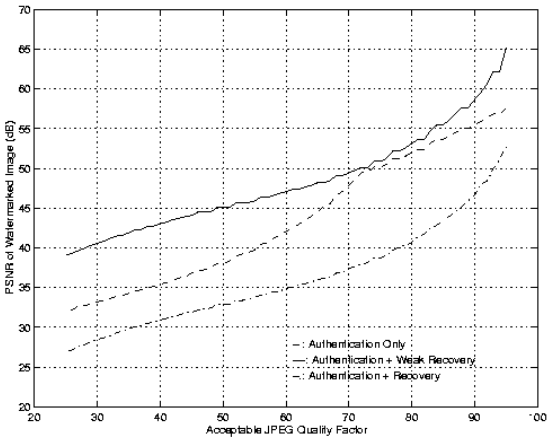
Demo, Encoder Download and Benchmarking

http://www.ctr.columbia.edu/sari

-- Multimedia Authentication On-line Research Resources

http://www.ctr.columbia.edu/~cylin

Image Quality after embedding



Authentication Only : 3 bits/block, (accepts QF ≥ 50)
Authentication + Weak Recovery: 9 bits/block, (accepts QF ≥ 50 for authentication and QF ≥ 75 for recovery) => *a compromise*
Authentication + Recovery: 9 bits/block, (accepts QF ≥ 50)

Experimental Results (I): Quality of watermarked images

Original Image

Authentication bits (PSNR=40.7 dB)

Authentication bits + Weak recovery (PSNR = 37.0 dB)

Authentication bits + Recovery bits (PSNR = 33.0 dB)

Experimental Results (II): Manipulation and Authentication output

Manipulation and JPEG compression on watermarked Images

A

Authentication bits only

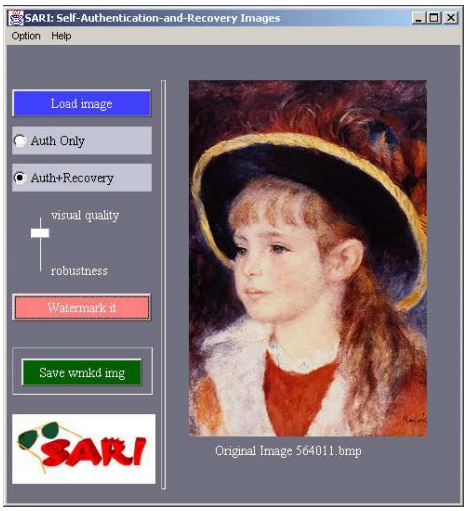
Authentication Output

A

Authentication bits + weak recovery bits

Authentication Output

SARI Example

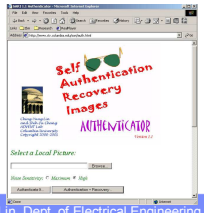


SARI Example: Manipulation by Photoshop

Compressed by JPEG

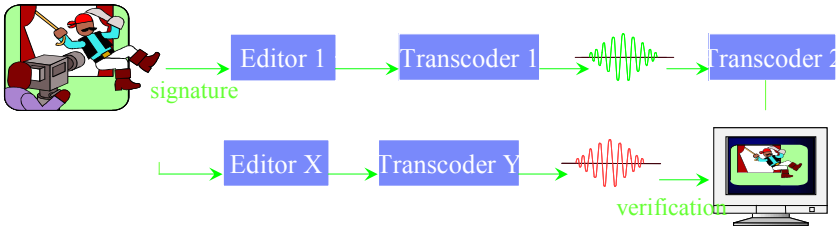
SARI Example: Authentication

using On-Line SARI Authenticator



SARI Example: *Self-Authentication-and-Recovery*

MPEG Video Authentication



- Acceptable Transcoding and Editing Processes:
  1. Dynamic Rate Shaping,
  2. Rate Control without Drift Error Correction,
  3. Rate Control with Drift Error Correction,
  4. Editing with Mostly Consistent Frame Types,
  5. Editing with Inconsistent Frame Types.

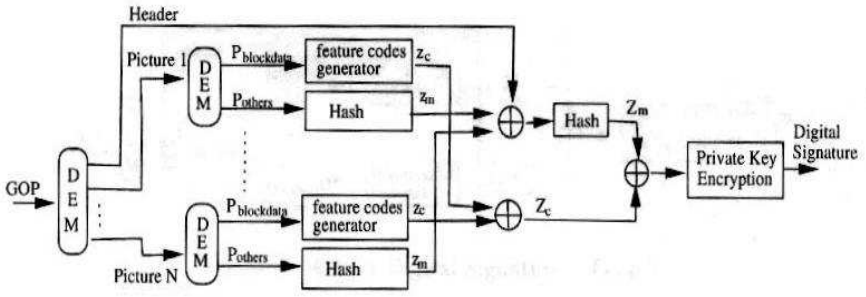
MPEG Video Authentication

- Acceptable Transcoding and Editing Processes:
  1. Dynamic Rate Shaping,
  2. Rate Control without Drift Error Correction,
  3. Rate Control with Drift Error Correction,
  4. Editing with Mostly Consistent Frame Types,
  5. Editing with Inconsistent Frame Types.

	1	2	3	4	5
DCT (residual) coefficients	X (drop some coefficients)	X (requanti-zation)		X	
Motion Vectors	X	X	X	X	
Picture Type (I,P,B)	X	X	X	X (inconsistent in boundary)	

Consistent properties of acceptable processes

Robust Digital Signature: Type I



Digital Signature = *Private Key Encryption* (Zc, Zm)

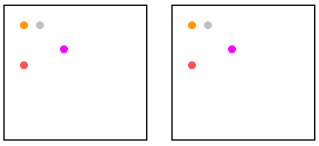
MPEG Compressed Video

- Video is composed of GOPs
- GOP => Pictures => Slices => Macro-Blocks (MB) => blocks
- Blocks:
  - DC coefficients
  - AC coefficients
    - quantization_step_size = (k Q[m][n]) / (8 v)

    where k: quantizer_scale (set by a Slice or MB)
    Q: quantization matrix -- Intra or Non-Intra
    v: 1, for MPEG-1 or 2- for MPEG-2

Feature Codes Generator

- Define a mapping function ***W*** within a macro-block (MB), s.t.,
  q = ***W***(***p***)
- For each block pair, compare the DCT values at the ***b*** selected positions.



Block ***p*** Block ***W***(***p***)

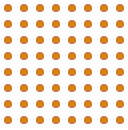
$$z_c = \mathrm{VLC}\left(\bigcup_p \bigcup_b \mathrm{sgn}[\, fp(b) - fq(b) \,] \right)$$

z_c: the feature codes of the MB

Authentication of Video Sequence
-- using Type I Robust Digital Signature

- Situation 1 (Rate Shaping):
  - DCT high frequency coefficients ==> 0
- Situation 2 (Rate Control without Error Drift Correction):
  - The possible changes of the sign values of the difference of a coefficient pair:
    + => +, 0 0 => 0 - => - , 0

Situation 1



Situation 2

- : original coefficients
- : re-quantized coefficients

Authentication of Video Sequence
-- using Type I Robust Digital Signature

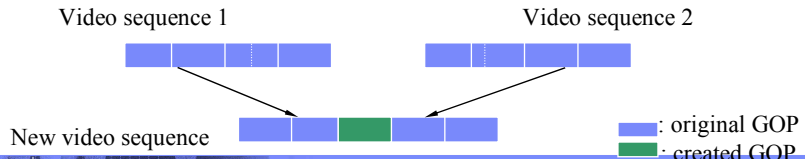
- Situation 3 (Rate Control with Error Drift Correction):

$$\text{If } fp(b) - fq(b) > 0 \text{ then } fp'(b) - fq'(b) >= -\tau$$
$$\text{If } fp(b) - fq(b) < 0 \text{ then } fp'(b) - fq'(b) <= \tau$$
$$\text{If } fp(b) - fq(b) = 0 \text{ then } -\tau <= fp'(b) - fq'(b) <= \tau$$

where
$$\tau = \begin{cases} 0, & \text{intra-block} \\ 1+\Sigma\ (k'_{refi}\, Q_{refi}(b) \ /\ k'\, Q_{nonintra}(b)), & \text{non-intra-block} \end{cases}$$

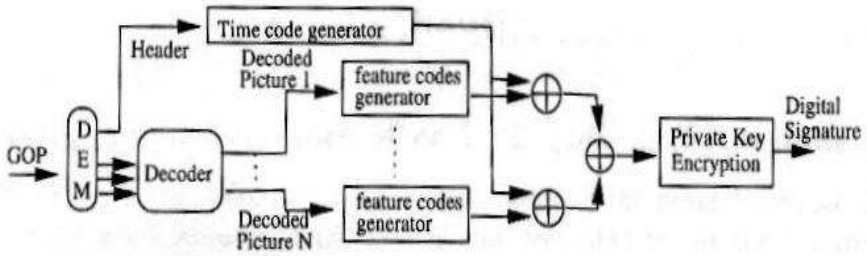
Authentication of Video Sequence
-- using Type I Robust Digital Signature

- Situation 4 (Editing with Mostly Consistent Picture Types):
  - Original GOPs
    - without other editing
    - with other editing such as *intensity enhancement, scaling, filter, etc.*
  - Created GOPs



Robust Digital Signature: Type II

- Situation 5:
  - Inconsistent: *GOP structure, motion vectors, DCT coefficients, …..*
  - Consistent: *Pixel Values (with possible small changes)*



Authentication of Video Sequence
-- using Type II Robust Digital Signature

- Situation 5 (Editing or Transcoding with Inconsistent Picture Types) :
  - small noise-like changes in the spatial domain result in small changes in the DCT domain.
    - 1-bit per comparison,
    - Using tolerance bound:

$$\text{If } fp(b) - fq(b) >= 0 \text{ then } fp'(b) - fq'(b) >= -\tau$$
$$\text{If } fp(b) - fq(b) < 0 \text{ then } fp'(b) - fq'(b) <= \tau$$

  - time codes.

Experiments

Frame 1

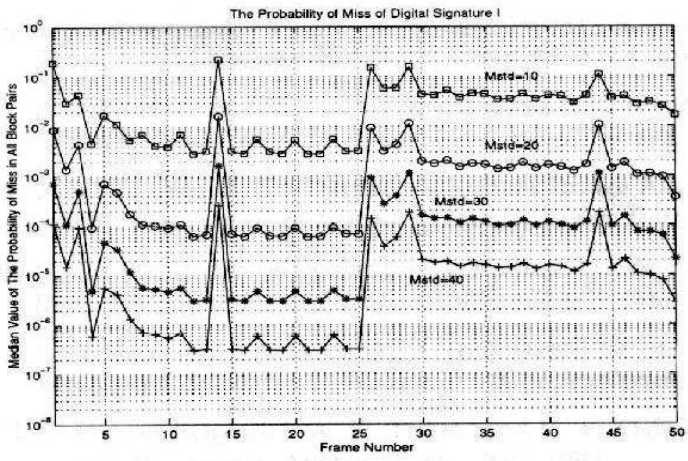
Frame 2

Frame 30

CIF Format, 1.5Mbps

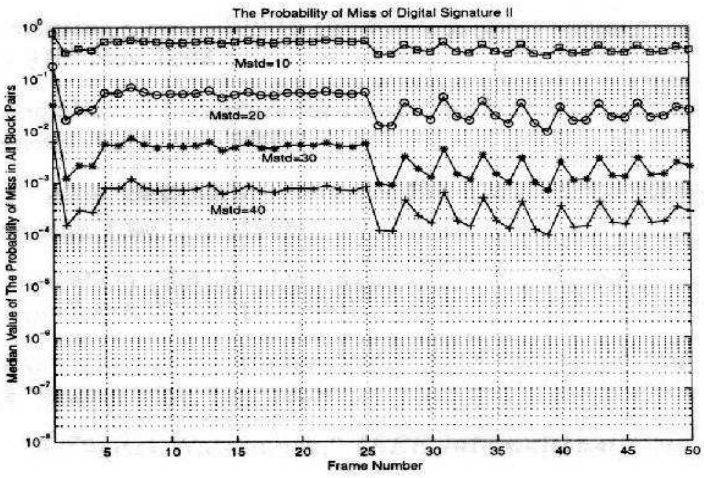
Frame 50

An Experimental Performance of Robust Digital Signature Type I



Manipulation modeled by Gaussian Distribution

An Experimental Performance of Robust Digital Signature Type II



Manipulation modeled by Gaussian Distribution

Summary of Video Authentication Work

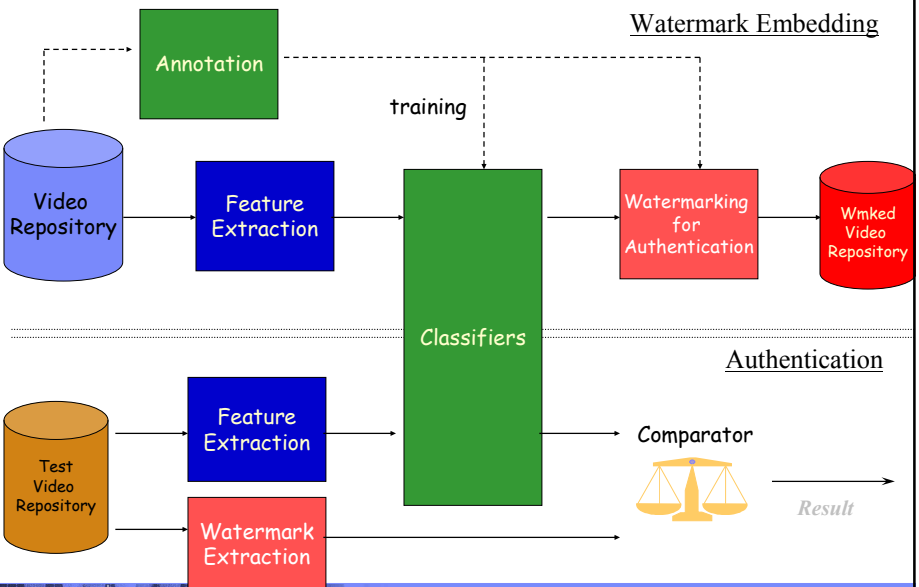
This Video Authentication System:

- Robust digital signatures can survive DCT-based compression and define acceptable manipulations.
- Manipulations can be detected within a block pair.
- It makes an authentication system more realistic to today's application scenarios.

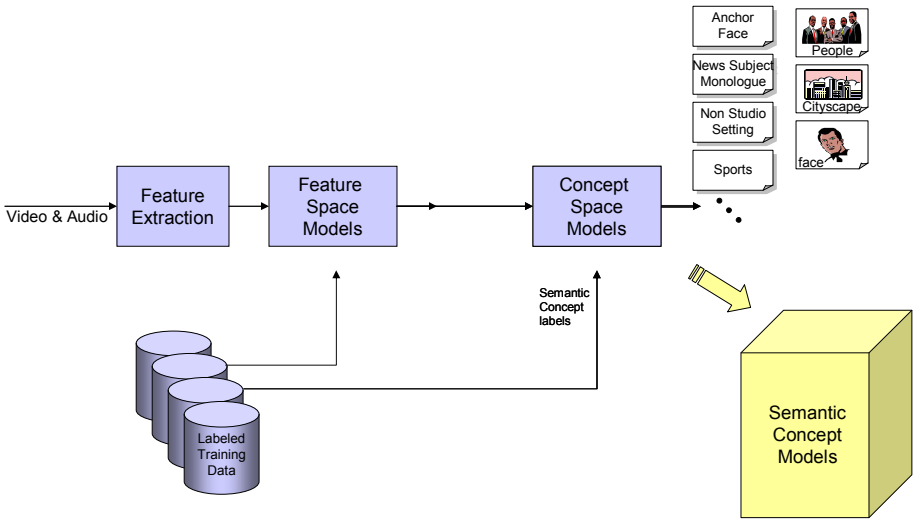
Semantic Authentication

- Objectives:
  - Objects: Male Face, Female Face, Man, Woman, Bill Clinton, Hilary Clinton
  - Events: walking together
  - Scene: lawn, tree, shadows
  - Relationships: hand-in-hand
- Methods: Segmentation, Classification, and Watermarking

Multimedia Semantic Authentication

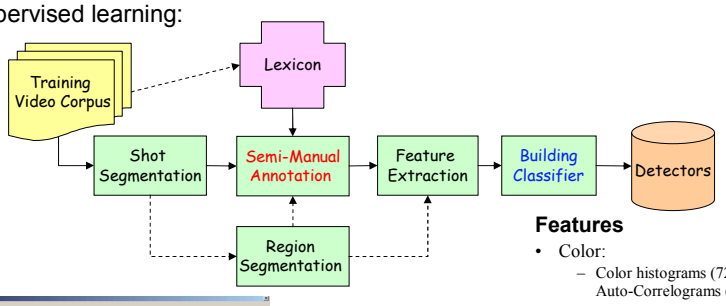


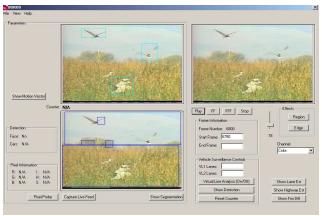
Training Phase – From Pixels to Semantics



Framework to build concept model vectors: supervised or quasi-supervised learning

Supervised learning:





- **Regions**
  - Object (motion, Camera registration)
  - Background (5 lg regions / shot)

Features

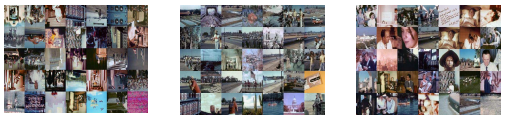
- Color:
  - Color histograms (72 dim, 512 dim), Auto-Correlograms (72 dim)
- Structure & Shape:
  - Edge orientation histogram (32 dim), Dudani Moment Invariants (6 dim), Aspect ratio of bounding box (1dim)
- Texture:
  - Co-occurrence texture (48 dim), Coarseness (1 dim), Contrast (1 dim), Directionality (1 dim), Wavelet (12 dim)
- Motion:
  - Motion vector histogram (6 dim)

VideoAL: IBM MPEG-7 Video Automatic Labeling System

MPEG-1 sequence

- detectors

"Text-overlay" "Outdoors" "People"



...

More than 100 Detectors for Automatic Labeling (as of Sept. 2003)

- Sport_Event
- Transportation_Event
- **Cartoon**
- Weather_News
- Physical_Violence
- Indoors
- Outdoors
- Outer_Space
- Animal
- Human
- Man_Made_Object
- Food
- Transportation
- Graphics_And_Text
- Sitting
- Standing
- Walking
- Running
- Addressing
- Parade
- Picnic
- Meeting
- Baseball
- Basketball
- Hockey
- Ice_Skating
- Swimming
- Tennis
- Football
- Soccer
- Car_Crash
- Road_Traffic
- Airplane_Takeoff
- Airplane_Landing
- Space_Vehicle_Launch
- Missle_Launch
- Explosion
- Riot
- Fight
- Gun_Shot
- Studio_Setting
- Non-Studio_Setting
- Nature_Vegetation
- Nature_Non-Vegetation
- Man_Made_Scene
- Chicken
- Fire
- Smoke
- Bridge
- Male_Face
- Female_Face
- Bill_Clinton
- Newt_Gingrich
- Male_News_Person
- Male_News_Subject
- Madeleine_Albright
- Female_News_Person
- Female_News_Subject
- Cityscape
- Cow
- Dog
- Fish
- Horse
- Pig
- **Face**
- **Person**
- People
- Crowd
- Clock
- Chair
- Desk
- Telephone
- Flag
- Newspaper
- Blackboard
- Monitor
- Whiteboard
- Microphone
- Podium
- Airplane
- Bicycle
- Boat
- **Car**
- Tractor
- Train
- Truck
- Bus
- Building
- Text_Overlay
- Scene_Text
- Graphics
- Painting
- Photographs
- House_Setting
- Classroom_Setting
- Factory_Setting
- Laboratory_Setting
- Meeting_Room_Setting
- Briefing_Room_Setting
- Office_Setting
- Store_Setting
- Transportation_Setting
- Flower
- Tree
- Forest
- Greenery
- Cloud
- **Sky**
- Water_Body
- Snow
- Beach
- Desert
- Land
- Mountain
- Rock
- Waterfall
- Road

Total Number:
2 (Aug 01) ->
47 (Aug 02) ->
119 (Sep 03)

Reference Papers

- C.-Y. Lin, "Digital Signature and Watermarking Techniques for Multimedia Authentication and Copyright Protection," *Columbia Ph.D. Thesis*, Chapter 2 and 3, Dec. 2000.
- C.-Y. Lin and B. L. Tseng, "**Segmentation, Classification, and Watermarking for Multimedia Semantic Authentication**," *IEEE Intl. Workshop on Multimedia Signal Processing*, US Virgin Islands, Dec. 2002.
- I. J. Cox, M. L. Miller and J. A. Bloom,, "Digital Watermarking," *Morgan Kaufmann*, Chapter 9 and 10, 2001.
- C. Kaufman, R. Perlman, and M. Speciner, "Network Security", Prentice-Hall, 1995, Chapters 4 and 5.

Homework Assignment #2 (due March 10, 2006)

1. [30 pts] Implement the LSB-based watermarking methods:
   1. Change the least significant bits in the spatial domain
   2. Change the least significant bits in the Block-based DCT domain before quantization
   3. Change the least significant bits in the Block-based DCT domain after quantization.
2. [30 pts] Implement the Spread Spectrum watermarking technique as mentioned in Cox et al. in the paper IEEE Trans. on Image Processing, Dec. 1997.
3. [40 pts] Embed watermark into video sequences by using:
   1. (Invisible watermark) Cox Spread Spectrum method
   2. (Visible watermark) High-Frequency DCT coefficient substitution

- Please discuss the robustness, invisibility and security issues of your implementations.