Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology

Introduction to Multimedia Signal Processing

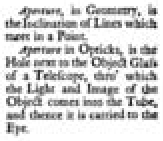
Kai-Lung Hua
2018

What is multimedia?

- Multimedia includes a combination of :
  - Text
  - Still images
  - Audio
  - Animation
  - Video
  - Interactivity content forms

[2]

Examples



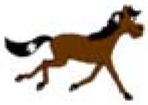
Aperture, in Geometry, is the Inclination of Lines which meet in a Point.

Aperture in Opticks, is the Hole next to the Object Glaſs of a Teleſcope, thro' which the Light and Image of the Object comes into the Tube, and thence it is carried to the Eye.

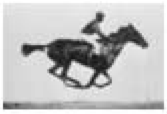
Text

Audio

Still Images



Animation



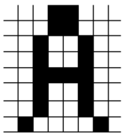
Video Footage

Interactivity

[3]

Text

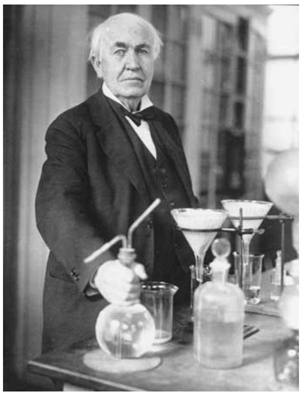
- When PC's were in their infancy running under MS-DOS
  - They only displayed text in one size and one color.
  - A text charter was 8 pixels high and 8 pixels wide



- Nowadays

[4]

Still Images



Thomas Alva Edison 1847 – 1931

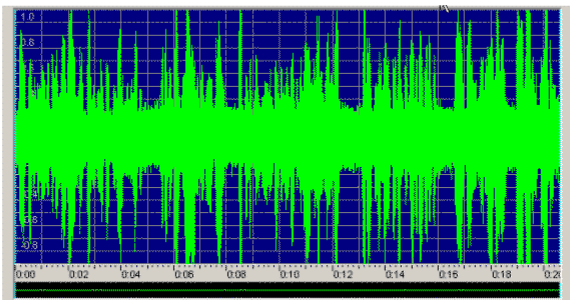
Mark Zuckerberg (1984)
Facebook founder

Processor chips can only copy data in byte multiples (8, 16, 32, or 64 bits).

[5]

Audio

- 1877 Thomas Alva Edison Demonstrated his cylinder phonograph



The image above is a map of the above wave file.

[6]

Animation

- Animation is a sequential series of still images that create an illusion of motion.

[7]

Video

- Unlike an animation that we can create from drawings or images a video movie is created by a photographic process and converted or ported to a computer.

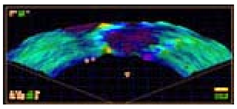
[8]

Introduction to Image and Video Processing

[9]

Images and Videos are Efficient and Effective

A political ad



Sonar Image

http://marsrovers.jpl.nasa.gov/gallery/press/opportunity/20040125a.html
JPL Mars' Panorama captured by the Opportunity

- Visual representations are often the most efficient way to represent information
- Images are used in many scenarios:
  - Politics, advertisements, scientific research, military, etc…
- Video signals can effectively tell a temporally evolving story
  - Arise in cinema (motion pictures)
  - Arise in surveillance
  - Arise in medical applications

[10]

Why Do We Process Images?

- Enhancement and restoration
  - Remove artifacts and scratches from an old photo/movie
  - Improve contrast and correct blurred images
- Transmission and storage
  - Images and Video can be more effectively transmitted and stored
- Information analysis and automated recognition
  - Recognizing terrorists
- Evidence
  - Careful image manipulation can reveal information not present
  - Detect image tampering
- Security and rights protection
  - Encryption and watermarking preventing illegal content manipulation

[11]

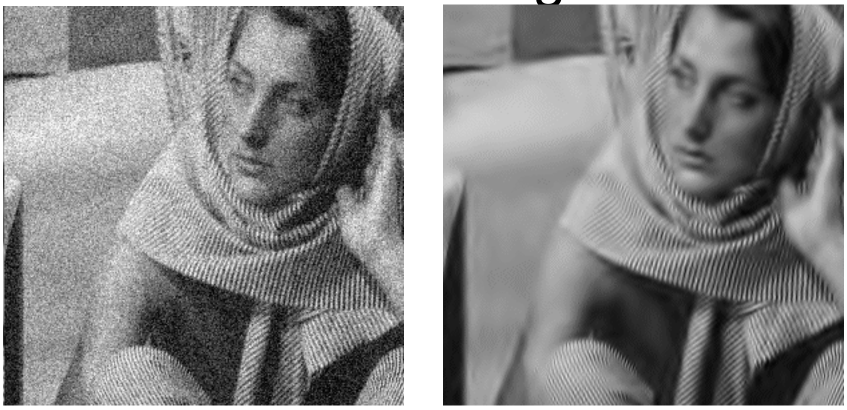
Compression

- Color image of 600x800 pixels
  - Without compression
    - 600*800 * 24 bits/pixel
      = 11.52K bits = 1.44M bytes
  - After JPEG compression (popularly used on web)
    - only 89K bytes
    - compression ratio ~ 16:1
- Movie
  - 720x480 per frame, 30 frames/sec, 24 bits/pixel
  - Raw video ~ 243M bits/sec
  - DVD ~ about 5M bits/sec
  - Compression ratio ~ 48:1

“Library of Congress” by M.Wu (600x800)

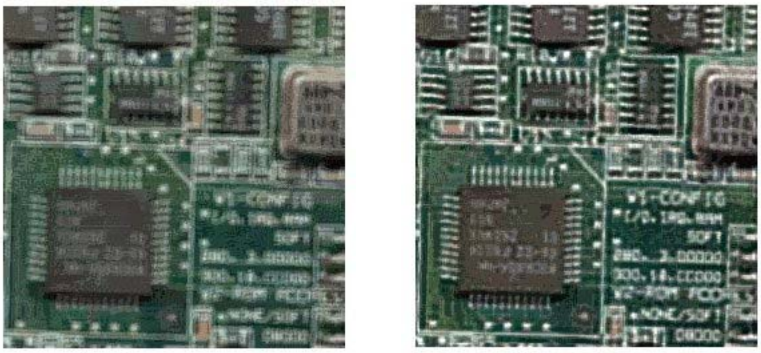
[12]

Denoising



From X.Li http://www.ee.princeton.edu/~lixin/denoising.htm

Deblurring



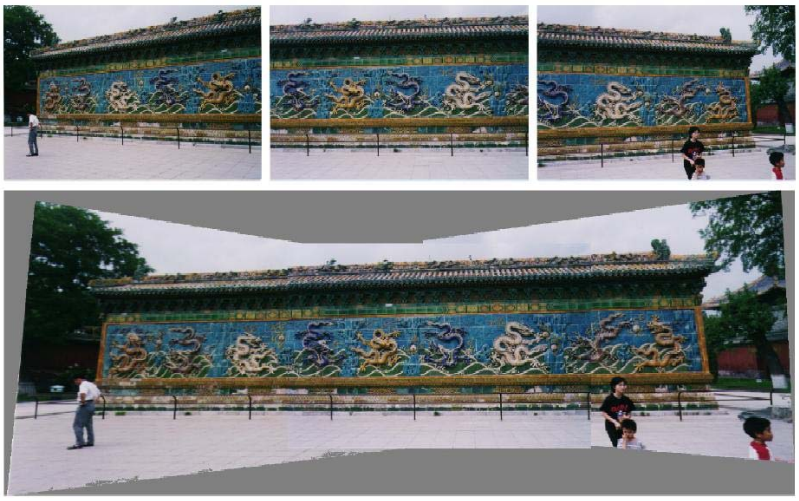
Blurred & noisy image

Restored image

From Mathworks

Visual Mosaicing

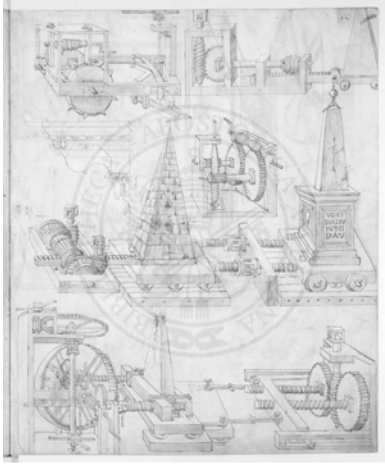
– Stitch photos together without thread or scotch tape



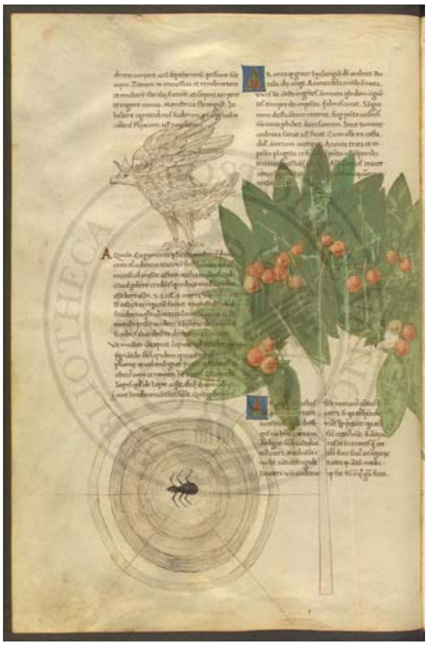
R.Radke – IEEE PRMI journal paper draft 5/01

[15]

Visible Digital Watermarks

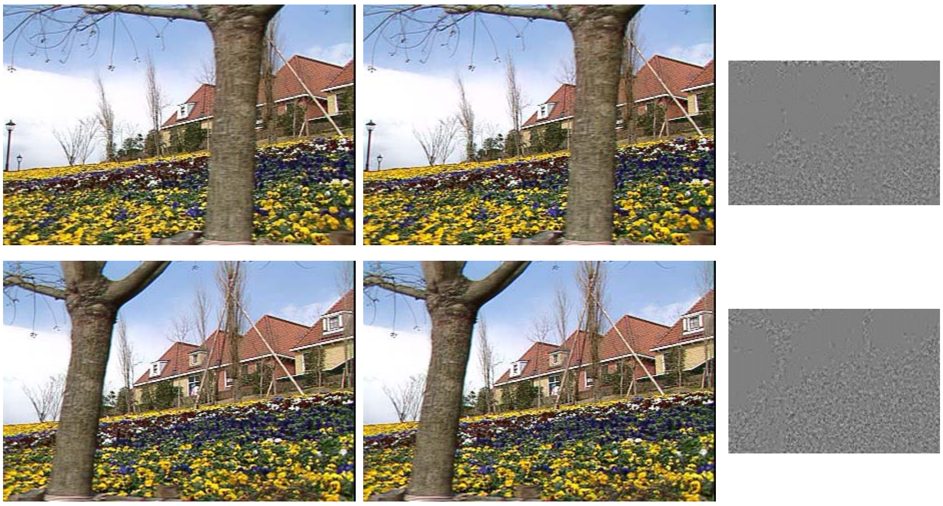


from IBM Watson web page
“Vatican Digital Library”



[16]

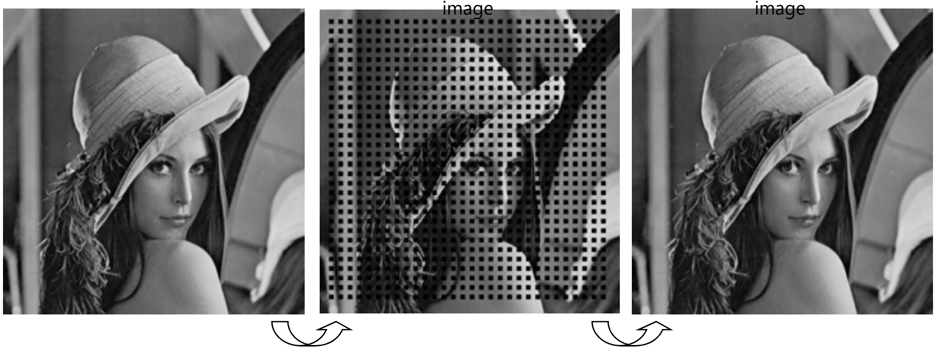
Invisible Watermark



- 1st & 30th Mpeg4.5Mbps frame of original, marked, and their luminance difference
- human visual model for imperceptibility: protect smooth areas and sharp edges

[17]

Error Concealment



(a) original lenna image

(b) corrupted lenna image

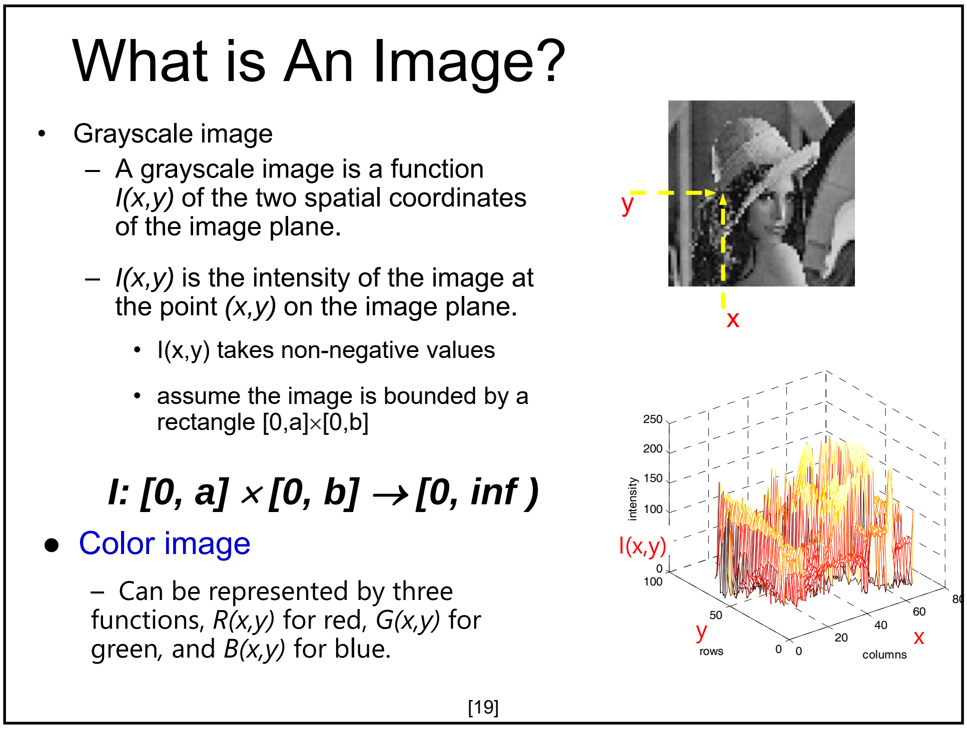
(c) concealed lenna image

25% blocks in a checkerboard pattern are corrupted

corrupted blocks are concealed via edge-directed interpolation

Examples were generated using the source codes provided by W.Zeng.

[18]



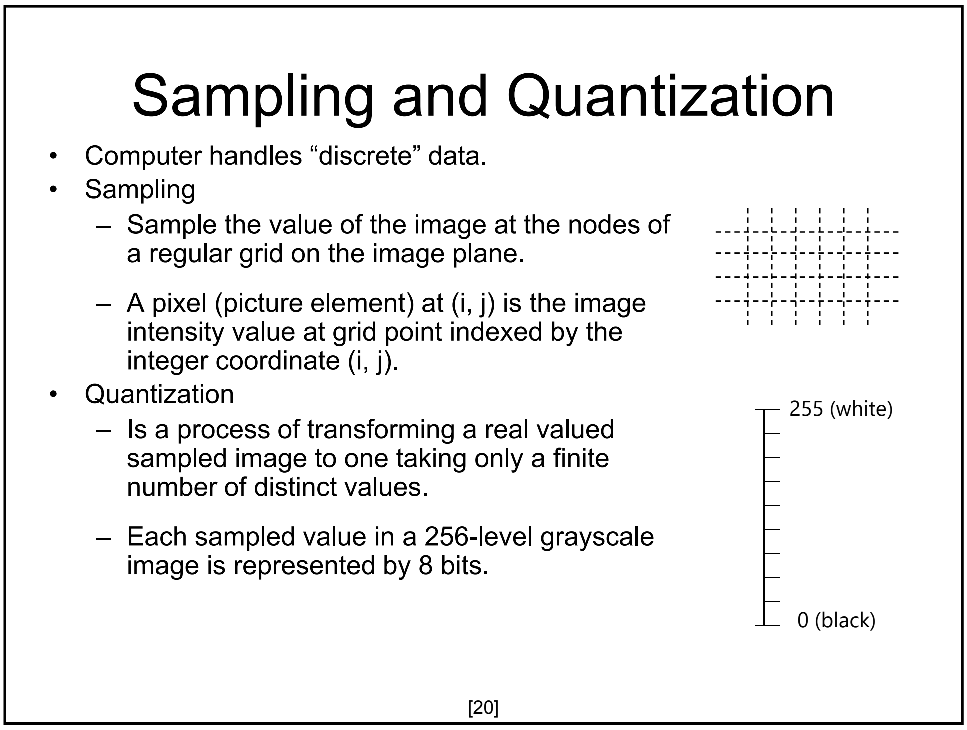
What is An Image?

- Grayscale image
  - A grayscale image is a function *I(x,y)* of the two spatial coordinates of the image plane.
  - *I(x,y)* is the intensity of the image at the point *(x,y)* on the image plane.
    - I(x,y) takes non-negative values
    - assume the image is bounded by a rectangle [0,a]×[0,b]

$$I: [0, a] \times [0, b] \rightarrow [0, \mathit{inf}\ )$$

- Color image
  - Can be represented by three functions, *R(x,y)* for red, *G(x,y)* for green, and *B(x,y)* for blue.

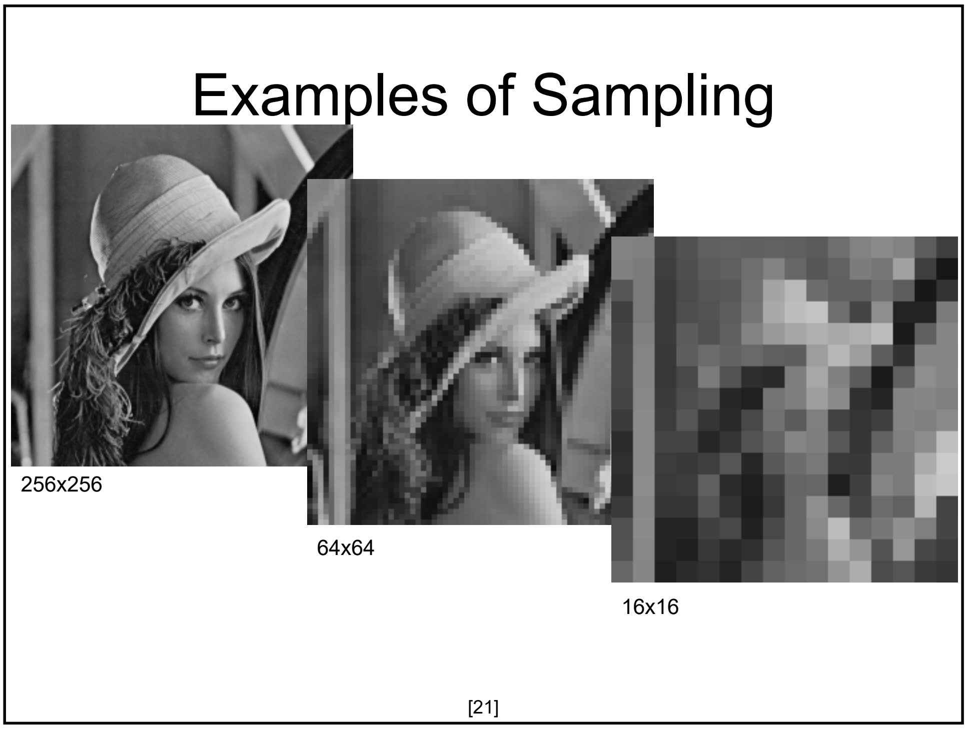
[19]



Sampling and Quantization

- Computer handles "discrete" data.
- Sampling
  - Sample the value of the image at the nodes of a regular grid on the image plane.
  - A pixel (picture element) at (i, j) is the image intensity value at grid point indexed by the integer coordinate (i, j).
- Quantization
  - Is a process of transforming a real valued sampled image to one taking only a finite number of distinct values.
  - Each sampled value in a 256-level grayscale image is represented by 8 bits.

[20]



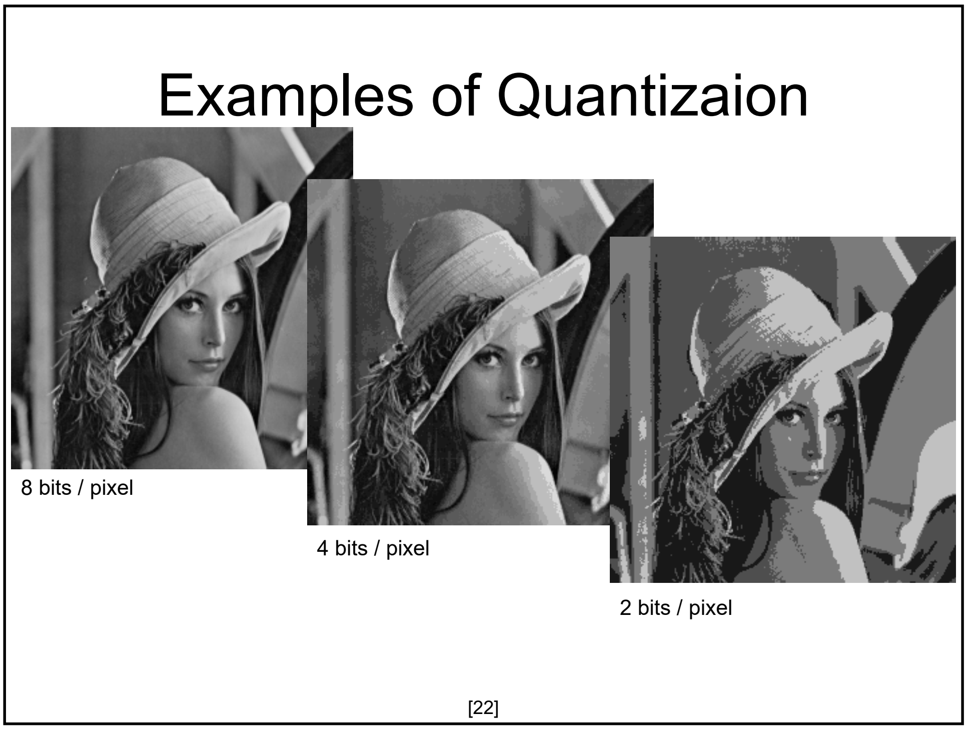
Examples of Sampling

256x256

64x64

16x16

[21]



Examples of Quantizaion

8 bits / pixel

4 bits / pixel

2 bits / pixel

[22]

Introduction to Multimedia Data Hiding

Why Hide Data in Multimedia

Crypto is Useful to Protect MM, but Not Enough …

- Encryption
  - Helps to protect confidentiality
  - Protection vanishes after decryption
  - Prefer a way to associate copyright info. with multimedia source even after decryption/compression/transmission/etc.

- Digital cryptographic signature
  - Helps to authenticate sender's identity and data integrity
  - Need to attach a separate signature to the data source
  - Audio/image/video allows imperceptible changes
  - Opportunities for new and seamless ways of authentication

Demands on Info. Security and Protection

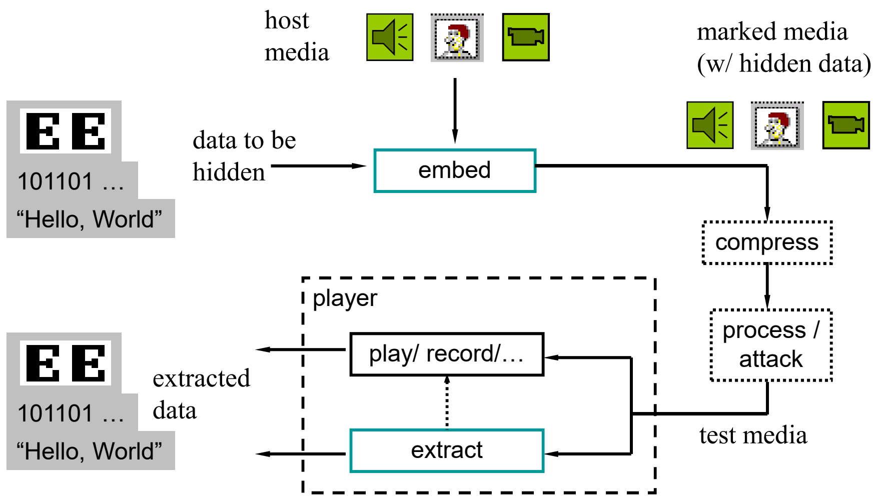
- Intellectual property management for digital media
  - Promising electronic marketplace for digital music and movies
  - Napster controversy
- Conventional encryption alone still leaves many problems unsolved
  - Protection from encryption vanishes once data is decrypted
    - Still want establish ownership and restrict illegal re-distributions
  - How to distinguish changes introduced by compression vs. malicious tampering?
    - Bit-by-bit accuracy is not always desired authenticity criterion for MM

Multimedia Data Hiding / Digital Watermarking

- What is it?
  - Example: Picture in picture, words in words
  - It is "Steganography"
  - Secondary information in perceptual digital media data
- Why hide information?
  - Convey information without an additional channel
  - Hidden data is tied to content: Can be made difficult to separate data from content.
  - Covert communication: Adversary does not know it is there!

[27]

General Framework of Data Hiding



host media

marked media (w/ hidden data)

data to be hidden

embed

compress

process / attack

test media

player

play/ record/…

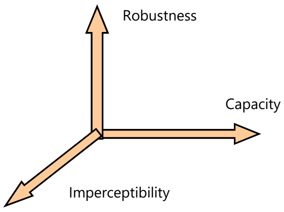
extract

extracted data

[28]

Issues and Challenges

- Tradeoff among conflicting requirements
  - Imperceptibility
  - Robustness & security
  - Capacity
- Key elements of data hiding
  - Perceptual model
  - Embedding one bit
  - Multiple bits
  - Uneven embedding capacity
  - Robustness and security
  - What data to embed

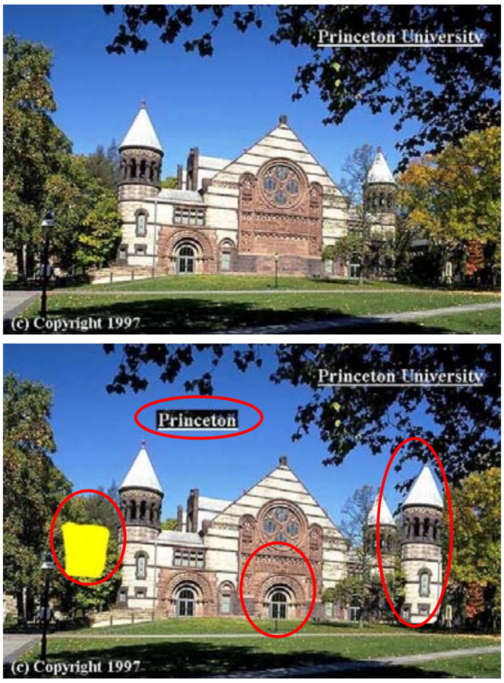


Upper Layers	……
	Coding of embedded data
	Security
	Error correction
	Uneven capacity equalization
Lower Layers	Multiple-bit embedding
	Imperceptible embedding of one bit

[29]

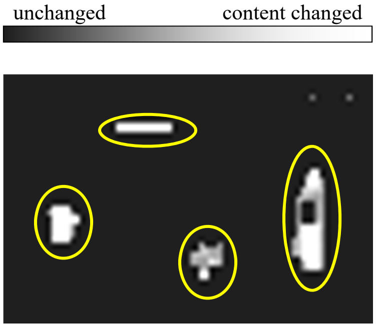
Examples of Multimedia Data Hiding

[30]



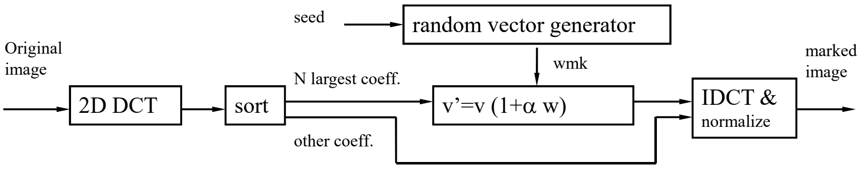
Watermark-based Authentication

- Embed patterns and content features using a lookup-table
- High embedding capacity/security via shuffling
  - *locate alteration*
  - *differentiate content vs. non-content change (compression)*



Spread Spectrum Approach: Cox et al (NECI)

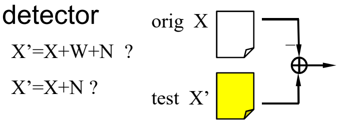
- Key points
  - Place wmk in perceptually significant spectrum (for robustness)
    - Modify by a small amount below Just-noticeable-difference (JND)
  - Use long random vector as wmk to avoid artifacts (for imperceptibility & robustness)
- Embedding $v'_i = v_i + \alpha v_i w_i = v_i (1+\alpha w_i)$ $\alpha = 0.1$, $w_i \sim$ iid, zeromean, unit variance
  - Perform DCT on entire image and embed wmk in DCT coeff.
  - Choose N=1000 largest AC coeff. and scale $\{v_i\}$ by a random factor



[32]

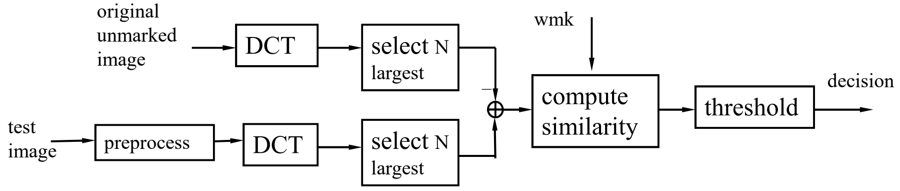
Cox's Scheme (cont'd)

- Detection
  - Subtract original image from the test one before running through detector



$$Y = X' - X \Rightarrow \begin{cases} H0: Y = N & \text{no watermark} \\ H1: Y = W + N & \text{watermark} \end{cases}$$

  - Original detection measure used by Cox et al. $sim(Y, W) = \frac{\langle Y, W \rangle}{\sqrt{\langle Y, Y \rangle}}$
    - a correlator normalized by |Y|



[33]

Theoretical Foundations

- Optimal detection for On-Off Keying (OOK)

$$\begin{cases} H_0: & y_i = d_i & \text{(if } b = 0) \\ H_1: & y_i = s_i + d_i & \text{(if } b = 1) \end{cases} \quad \text{for i} = 1 \sim \text{n}$$

  - OOK under i.i.d. Gaussian noise $\{d_i\}$
    - b∈{0,1} represents absence vs. presence of ownership mark
    - Use a correlator-type detector (recall the review last week)
  - Need to determine how to choose $\{s_i\}$
- Neyman-Pearson Detection [Poor's book Sec.2.4]
  - False-alarm ~ claiming wmk existence when nothing embedded
  - Given max. allowed false-alarm, try to minimize prob. of miss detection
    - Use likelihood ratio as detection statistic
    - Determine threshold according to false-alarm prob.

[34]

Performance of Cox's Scheme

- Robustness
  - (claimed) scaling, JPEG, dithering, cropping, "printing-xeroxing-scanning", multiple watermarking

Distortion	none	scale 25%	JPG 10%	JPG 5%	dither	crop 25%	print-xerox-scan
similarity	32.0	13.4	22.8	13.9	10.5	14.6	7.0

threshold = 6.0 (determined by setting false alarm probability)

  - No big surprise with high robustness
    - equiv. to conveying just 1-bit {0,1} with O(10^3) samples
- Comment
  - Must store original unmarked image ⇒ "private wmk", "non-blind" detect.
  - Perform image registration if necessary
  - Adjustable parameters: N and α

[35]

Invisible Robust Wmk: Improved Schemes

- Apply better Human-Perceptual-Model
  - Global scaling factor is not suitable for all coefficients
  - Explicitly computes Just-noticeable-difference (JND)
    - JND ~ max amount each freq. coeff. can be modified imperceptibly
    - Use α_i for each coeff. ⇒ finely tune wmk strength

$$v_i' = v_i \cdot (1 + \alpha_i \cdot w_i)$$

  - Better tradeoff between imperceptibility and robustness
    - Try to add a watermark as strong as possible
- Block-DCT based schemes:
  - Podilchuk-Zeng & Swanson et al.
  - Existing visual model for block DCT: JPEG Quality Factors

[36]

Compare Cox & Podilchuk Schemes

Original

Cox

whole image DCT

Embed in 1000 largest coeff.

"embeddables"

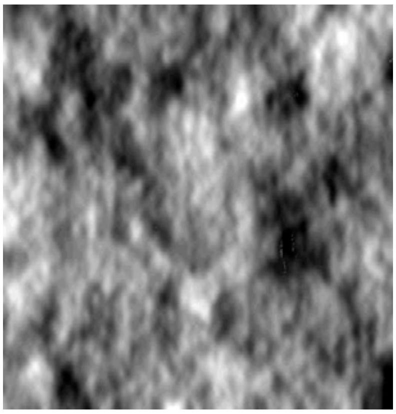
Podilchuk

block-DCT

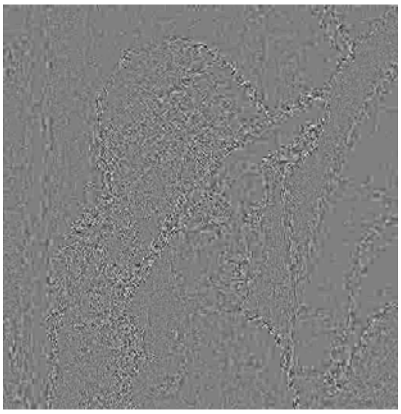
Embed to all

[37]

Compare Cox & Podilchuk Schemes (cont'd)

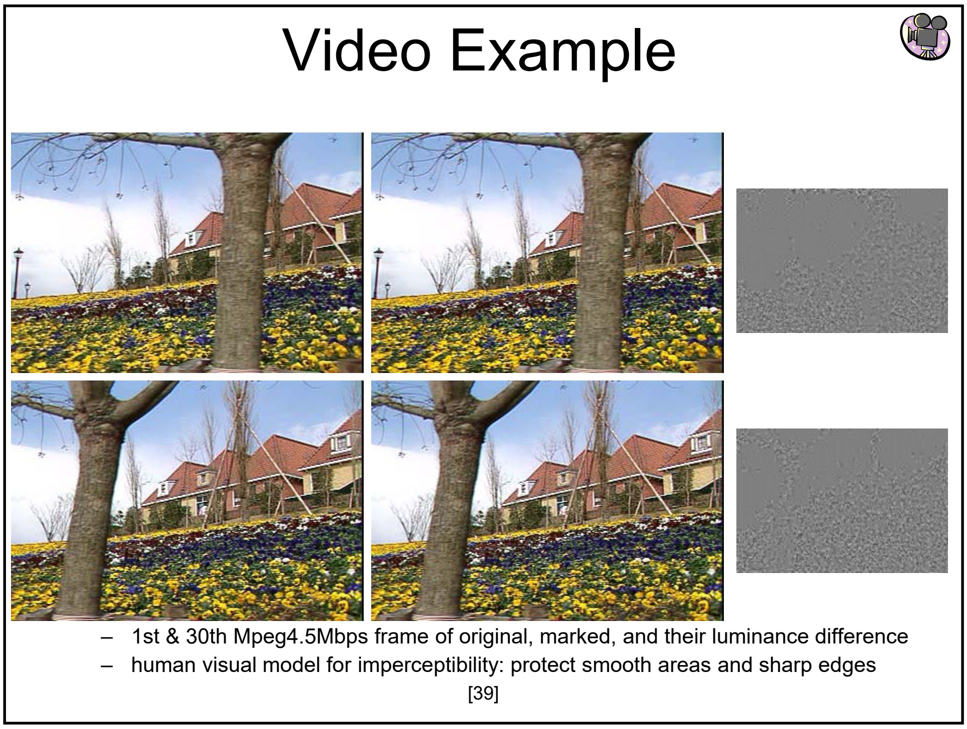


Cox



Podilchuk

[38]



Video Example

- 1st & 30th Mpeg4.5Mbps frame of original, marked, and their luminance difference
- human visual model for imperceptibility: protect smooth areas and sharp edges

[39]

Summary and Conclusions

- Data hiding in digital multimedia for a variety of purposes, involving multiple disciplines
- Tradeoff among many criterions
- Important to think both as designer and as attacker
- Emerging problems
  - how to effectively combine with other security mechanisms
- Data hiding in market
  - digital cameras with authentication watermark module
  - plug-in for image editors
  - video watermark proposals for DVD copy control
  - on-going effort for digital music
  - "Digital Rights Management (DRM)" for multimedia data

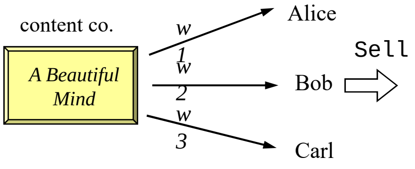
[40]

Suggested reading

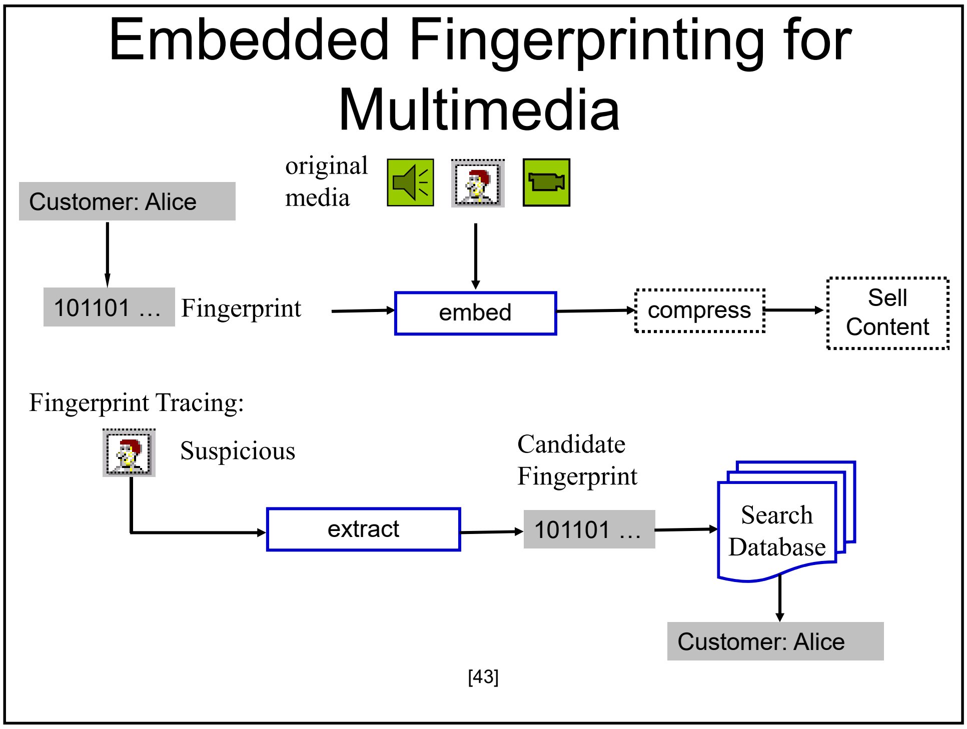
- I. Cox, J. Kilian, T. Leighton, T. Shamoon: “Secure Spread Spectrum Watermarking for Multimedia", IEEE Transaction on Image Processing, vol.6, no.12, pp.1673-1687, 1997.
  - Download from IEEE online journal, or http://www.neci.nj.nec.com/homepages/ingemar/papers/ip97.ps
- C. Podilchuk and W. Zeng, “Image Adaptive Watermarking Using Visual Models,” IEEE Journal Selected Areas of Communications (JSAC), vol.16, no.4, May, 1998.
  - Download from IEEE online journal
- M. Wu and B. Liu: “Multimedia Data Hiding”, Springer Verlag, to appear Dec. 2002. An earlier dissertation version is at http://www.ece.umd.edu/~minwu/research/phd_thesis.html.

Digital Fingerprinting and Tracing Traitors

- Recent advances in communications allow for convenient information sharing
- Severe damage is created by unauthorized information leakage
  - e.g., pirated content or classified document



- Promising countermeasure: robustly embed digital fingerprints
  - Insert ID or “fingerprint” to identify each customer
  - Prevent improper redistribution of multimedia content



Embedded Fingerprinting for Multimedia

Customer: Alice

original media

101101 … Fingerprint

embed

compress

Sell Content

Fingerprint Tracing:

Suspicious

extract

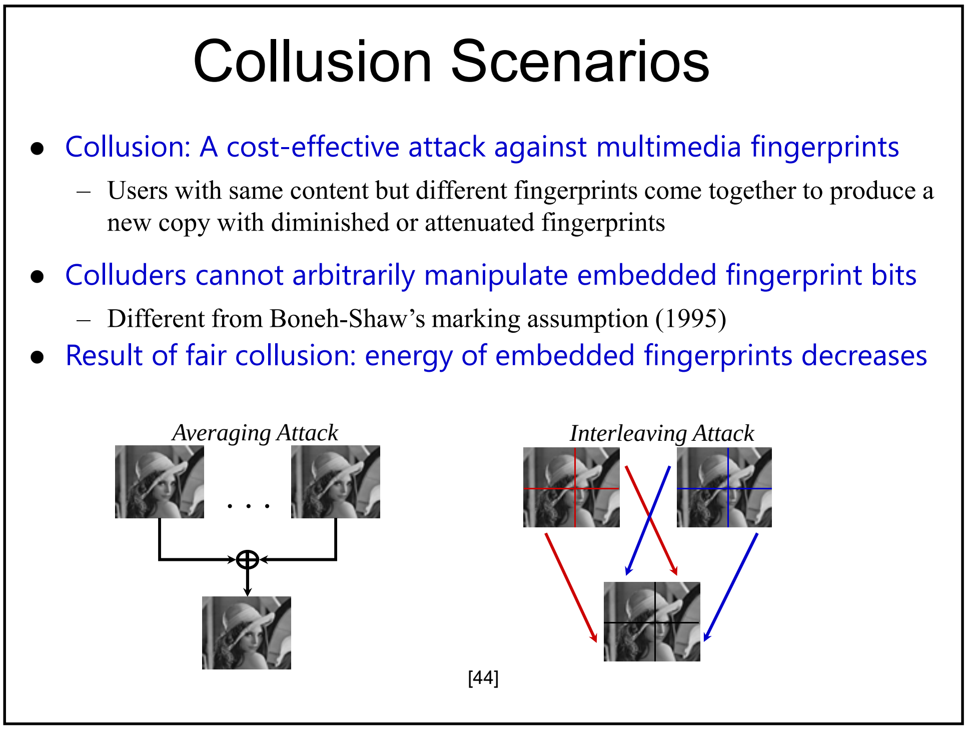
Candidate Fingerprint

101101 …

Search Database

Customer: Alice

[43]



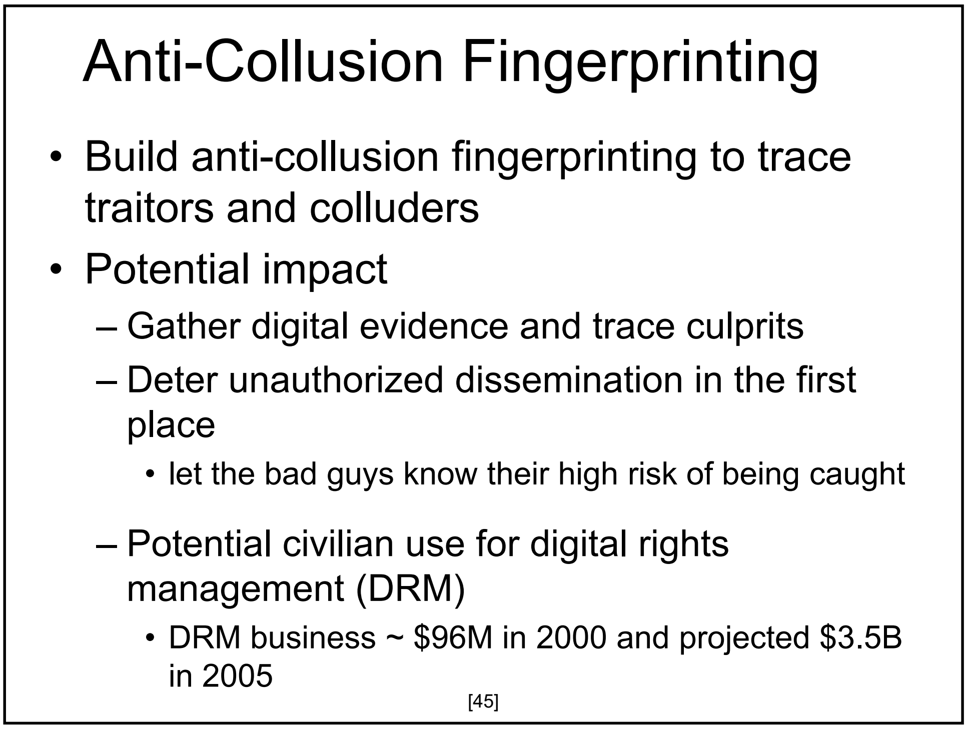
Collusion Scenarios

- Collusion: A cost-effective attack against multimedia fingerprints
  - Users with same content but different fingerprints come together to produce a new copy with diminished or attenuated fingerprints
- Colluders cannot arbitrarily manipulate embedded fingerprint bits
  - Different from Boneh-Shaw’s marking assumption (1995)
- Result of fair collusion: energy of embedded fingerprints decreases

Averaging Attack

Interleaving Attack

[44]



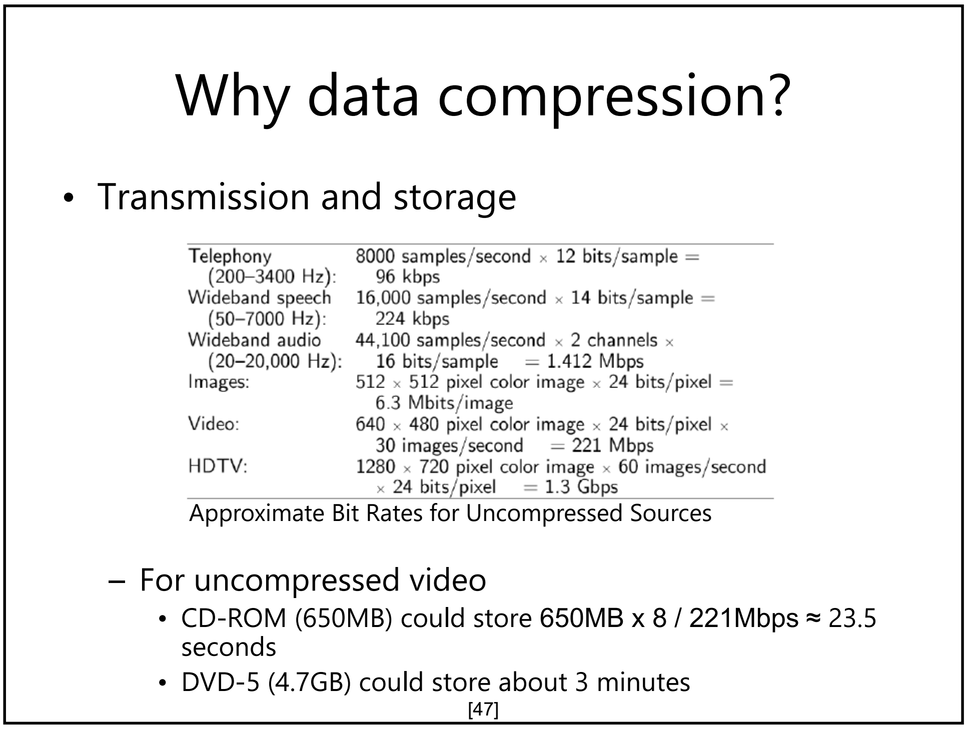
Anti-Collusion Fingerprinting

- Build anti-collusion fingerprinting to trace traitors and colluders
- Potential impact
  - Gather digital evidence and trace culprits
  - Deter unauthorized dissemination in the first place
    - let the bad guys know their high risk of being caught
  - Potential civilian use for digital rights management (DRM)
    - DRM business ~ $96M in 2000 and projected $3.5B in 2005

[45]

Introduction to data compression

[46]



Why data compression?

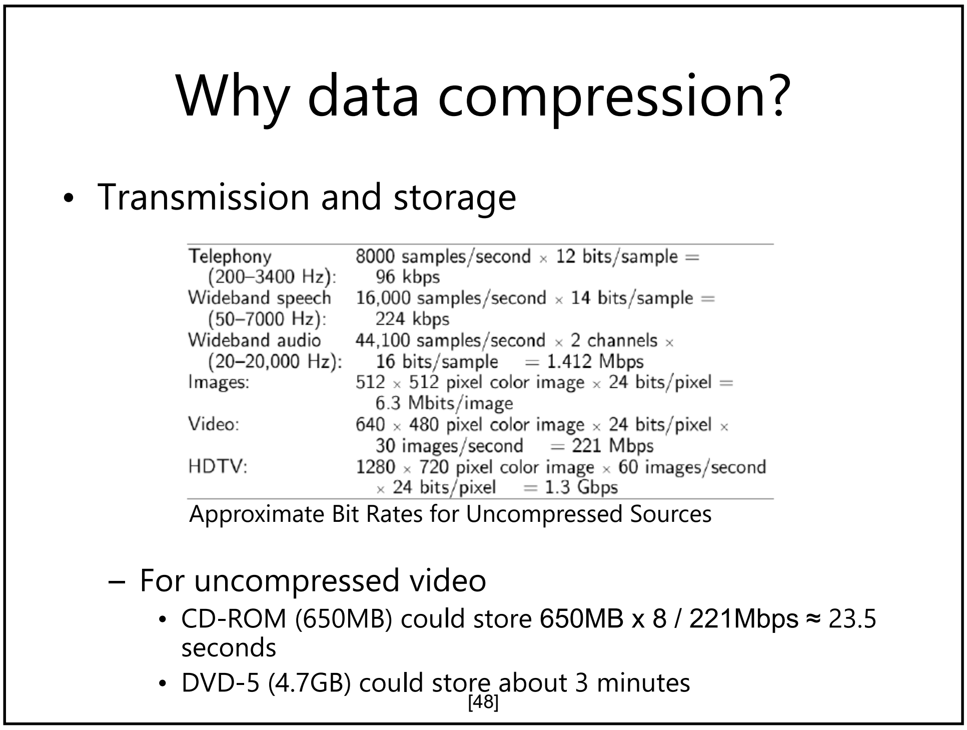
- Transmission and storage

Telephony (200–3400 Hz):	8000 samples/second × 12 bits/sample = 96 kbps
Wideband speech (50–7000 Hz):	16,000 samples/second × 14 bits/sample = 224 kbps
Wideband audio (20–20,000 Hz):	44,100 samples/second × 2 channels × 16 bits/sample = 1.412 Mbps
Images:	512 × 512 pixel color image × 24 bits/pixel = 6.3 Mbits/image
Video:	640 × 480 pixel color image × 24 bits/pixel × 30 images/second = 221 Mbps
HDTV:	1280 × 720 pixel color image × 60 images/second × 24 bits/pixel = 1.3 Gbps

Approximate Bit Rates for Uncompressed Sources

- For uncompressed video
  - CD-ROM (650MB) could store 650MB x 8 / 221Mbps ≈ 23.5 seconds
  - DVD-5 (4.7GB) could store about 3 minutes

[47]



Why data compression?

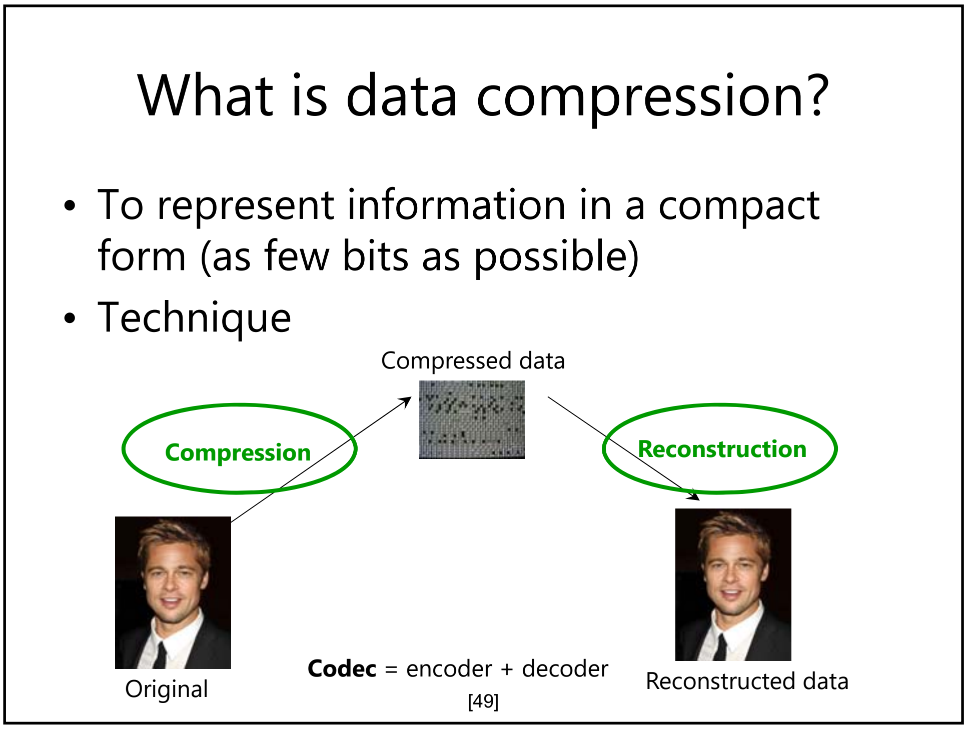
- Transmission and storage

Telephony (200–3400 Hz):	8000 samples/second × 12 bits/sample = 96 kbps
Wideband speech (50–7000 Hz):	16,000 samples/second × 14 bits/sample = 224 kbps
Wideband audio (20–20,000 Hz):	44,100 samples/second × 2 channels × 16 bits/sample = 1.412 Mbps
Images:	512 × 512 pixel color image × 24 bits/pixel = 6.3 Mbits/image
Video:	640 × 480 pixel color image × 24 bits/pixel × 30 images/second = 221 Mbps
HDTV:	1280 × 720 pixel color image × 60 images/second × 24 bits/pixel = 1.3 Gbps

Approximate Bit Rates for Uncompressed Sources

- For uncompressed video
  - CD-ROM (650MB) could store 650MB x 8 / 221Mbps ≈ 23.5 seconds
  - DVD-5 (4.7GB) could store about 3 minutes

[48]

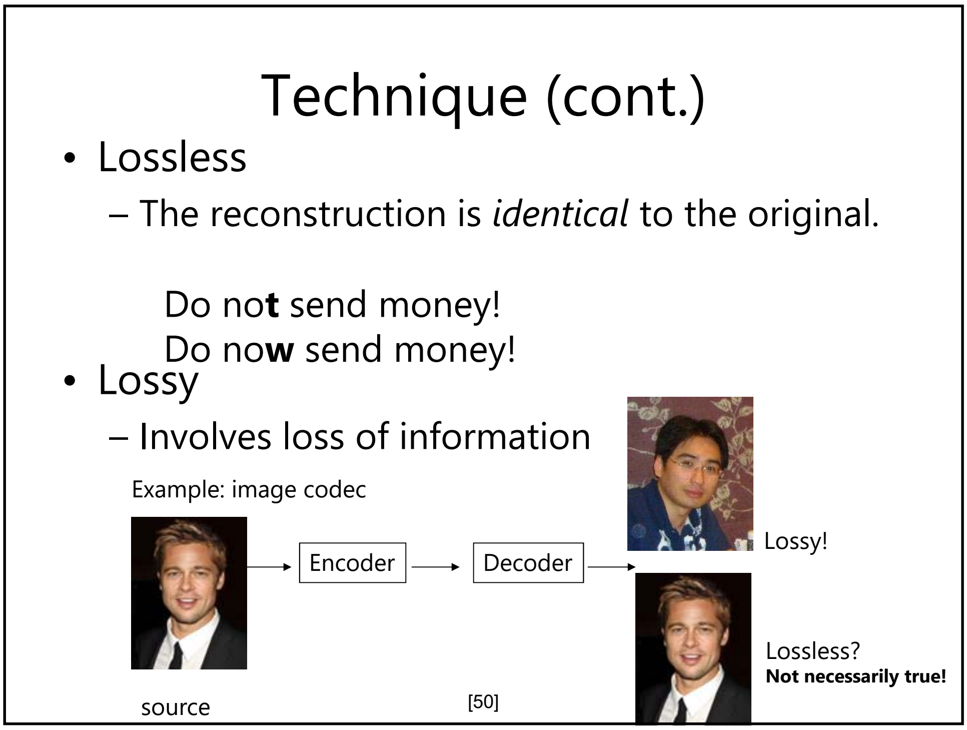


What is data compression?

- To represent information in a compact form (as few bits as possible)
- Technique

Codec = encoder + decoder

[49]



Technique (cont.)

- Lossless
  - The reconstruction is *identical* to the original.

    Do no**t** send money!
    Do no**w** send money!
- Lossy
  - Involves loss of information

Example: image codec

[50]

Two phases: modeling and coding

Original

→ Encoder →

Compressed data

Fewer bits!

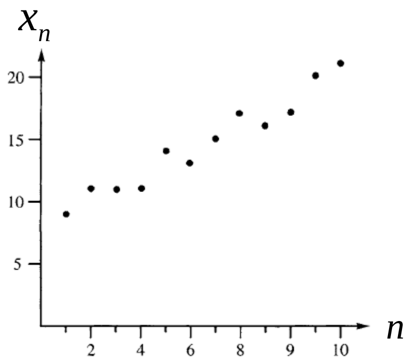
- Modeling
  - Discover the structure in the data
  - Extract information about any redundancy
- Coding
  - Describe the model and the ***residual*** (how the data differ from the model)

[51]

Example: Modeling

Data: x_n

9	11	11	11	14	13	15	17	16	17	20	21



Model: $\hat{x}_n = n + 8 \quad n = 1,2,\ldots$

[52]

Example: Coding

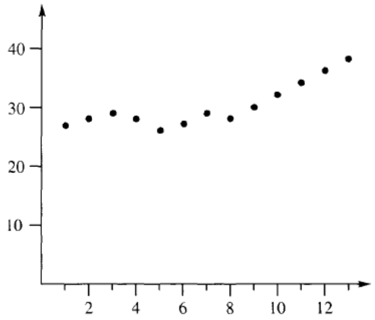
Original data x_n	9	11	11	11	14	13	15	17	16	17	20	21
Model $\hat{x}_n = n + 8$	9	10	11	12	13	14	15	16	17	18	19	20
Residual $e_n = x_n - \hat{x}_n$	0	1	0	-1	1	-1	0	1	-1	-1	1	1

- {-1, 0, 1}
- 2 bits * 12 samples = 24 bits (compared with 60 bits before compression)

We use the model to ***predict*** the value, then encode the residual!

Example (2)

27	28	29	28	26	27	29	28	30	32	34	36	38
27	1	1	-1	-2	1	2	-1	2	2	2	2	2



$$\hat{x}_n = x_{n-1} \quad n = 2,3,\ldots$$

Example (4)

- Morse Code (1838)

a	•▬
b	▬•••
c	▬•▬•
d	▬••
e	•
f	••▬•
g	▬▬•
h	••••
i	••
j	•▬▬▬
k	▬•▬
l	•▬••
m	▬▬
n	▬•
o	▬▬▬
p	•▬▬•
q	▬▬•▬
r	•▬•
s	•••
t	▬
u	••▬
v	•••▬
w	•▬▬
x	▬••▬
y	▬•▬▬
z	▬▬••

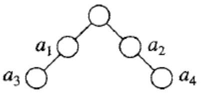
Shorter codes are assigned to letters that occur more frequently!

[55]

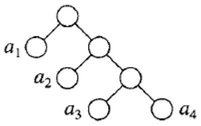
Prefix Codes

- No codeword is a *prefix* to another codeword
- Uniquely decodable

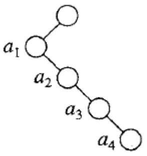
Letters	Code 2	Code 3	Code 4
a_1	0	0	0
a_2	1	10	01
a_3	00	110	011
a_4	11	111	0111



Code 2



Code 3



Code 4

[56]

Coding (cont.)

- For compression
  - Uniquely decodable
  - Short codewords
  - Instantaneous (easier to decode)

1. What sets of code lengths are possible?
2. Can we always use prefix codes?

[57]

Introduction to Huffman codes

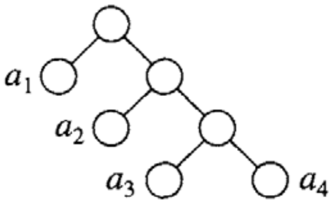
[58]

Outline

- Huffman codes
  - – Basic form and the algorithm
  - – Length of Huffman codes
  - – Extended form

Huffman Coding

- Observations of prefix codes

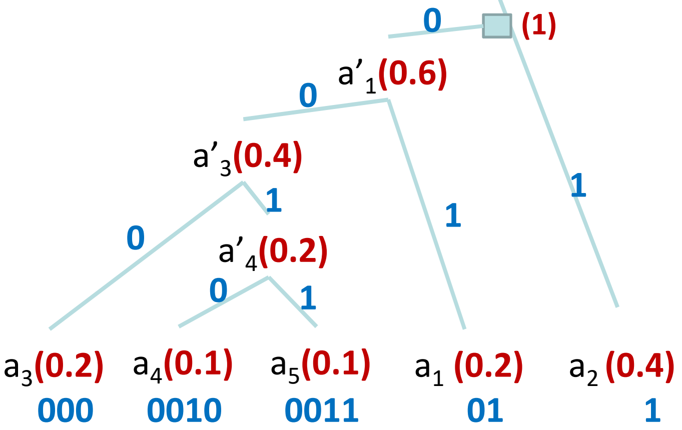


  - – Frequent symbols have shorter codewords
  - – The two symbols that occur ***least frequently*** have the same length

Huffman procedure:
Two least frequent symbols differ only in the last bit. Ex: ***m***0, ***m***1

Algorithm

- A = {a_1, a_2, a_3, a_4, a_5}
- P(a_1) = 0.2, P(a_2) = 0.4, P(a_3) = 0.2, P(a_4) = 0.1, P(a_5) = 0.1

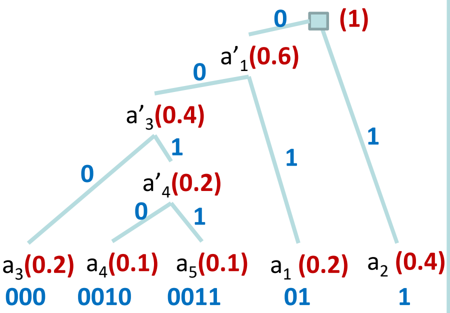


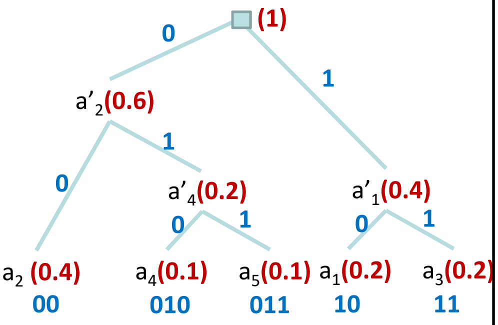
[61]

Algorithm

Letter	Probability	Codeword
a_2	0.4	1
a_1	0.2	01
a_3	0.2	000
a_4	0.1	0010
a_5	0.1	0011

- Entropy: 2.122 bits/symbol
- Average length: 2.2 bits/symbol





Which code is preferred? ***The one with minimum variance!***

Length of Huffman Codes

- Lower and upper bounds

$$\boxed{H(\mathcal{S}) \leq \bar{l} < H(\mathcal{S}) + 1}$$

  - H(S): entropy
  - $\bar{l}$: average code length

Extended Huffman Codes (1)

- Consider small alphabet and skewed probabilities
  - Example:

symbol	Prob.	codeword
a	0.9	0
b	0.1	1

1 bit / letter
No compression!

- Block multiple symbols together

symbol	Prob.	codeword
aa	0.81	0
ab	0.09	10
ba	0.09	110
bb	0.01	111

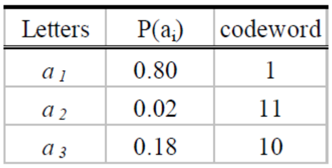
0.645 bit / letter

- Bounds: $H(S) \leq R \leq H(S) + \frac{1}{n}$

Number of letters per block

Extended Huffman Codes (2)

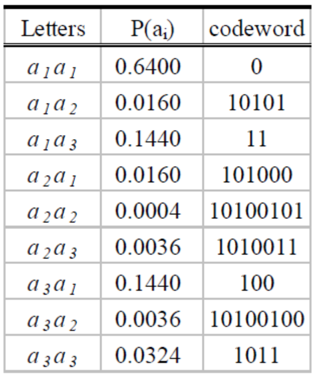
Another example:



Letters	P(a$_i$)	codeword
a_1	0.80	1
a_2	0.02	11
a_3	0.18	10

H = 0.816 bits/symbol

$\bar{l}$ = 1.2 bits/symbol

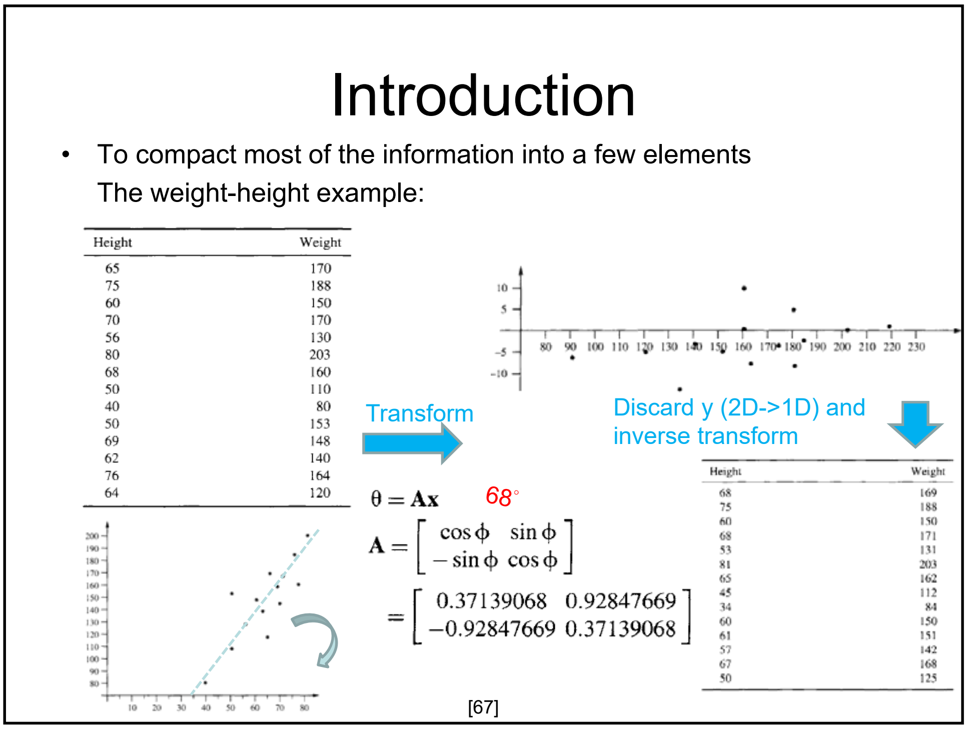


Letters	P(a$_i$)	codeword
a_1a_1	0.6400	0
a_1a_2	0.0160	10101
a_1a_3	0.1440	11
a_2a_1	0.0160	101000
a_2a_2	0.0004	10100101
a_2a_3	0.0036	1010011
a_3a_1	0.1440	100
a_3a_2	0.0036	10100100
a_3a_3	0.0324	1011

H = 1.6315 bits/block = 0.816 bits/symbol

$\bar{l}$ = 1.7228 / 2 = 0.8614 bits/symbol

Introduction to Transform Coding



Introduction

- To compact most of the information into a few elements
  The weight-height example:

Height	Weight
65	170
75	188
60	150
70	170
56	130
80	203
68	160
50	110
40	80
50	153
69	148
62	140
76	164
64	120

Transform

$$\theta = \mathbf{A}\mathbf{x}$$

68°

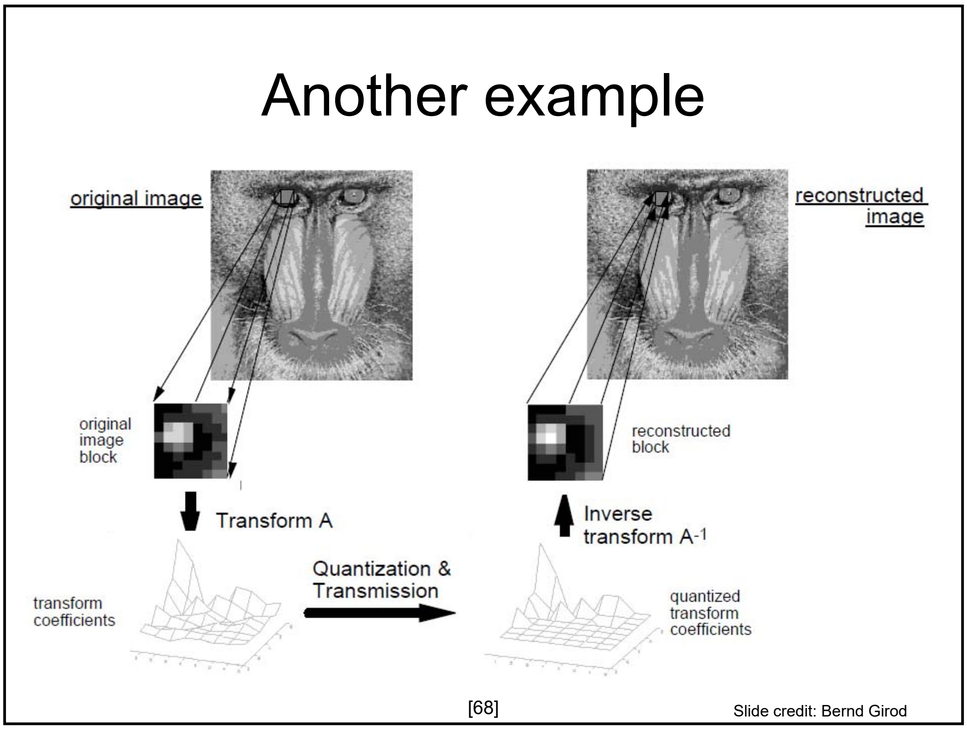
$$\mathbf{A} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

$$= \begin{bmatrix} 0.37139068 & 0.92847669 \\ -0.92847669 & 0.37139068 \end{bmatrix}$$

Discard y (2D->1D) and inverse transform

Height	Weight
68	169
75	188
60	150
68	171
53	131
81	203
65	162
45	112
34	84
60	150
61	151
57	142
67	168
50	125

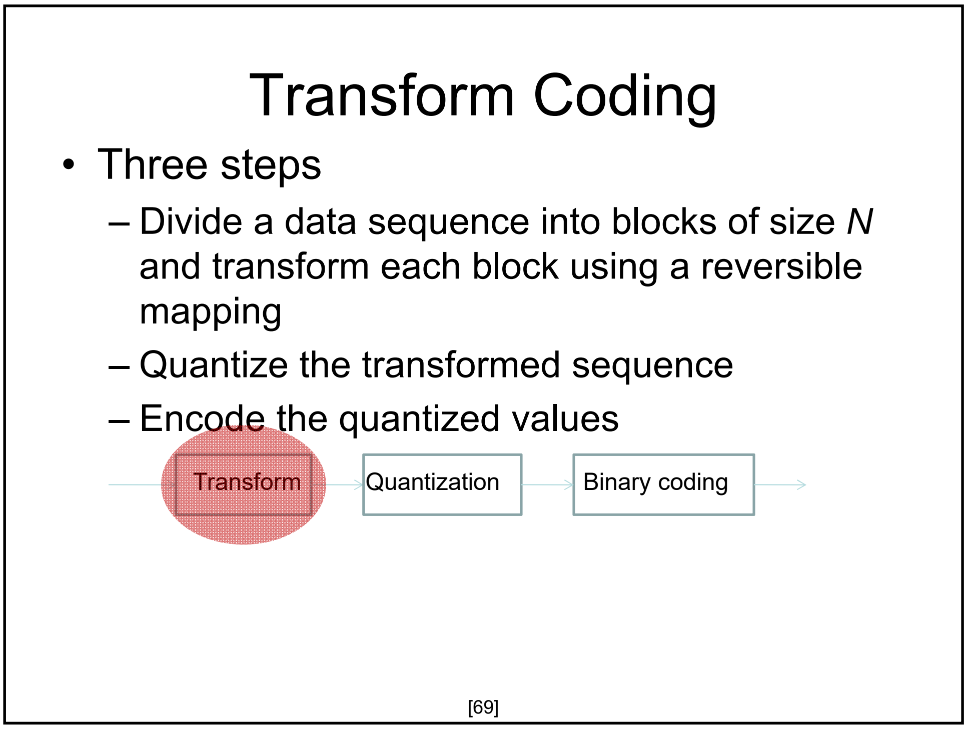
[67]



Another example

[68]

Slide credit: Bernd Girod

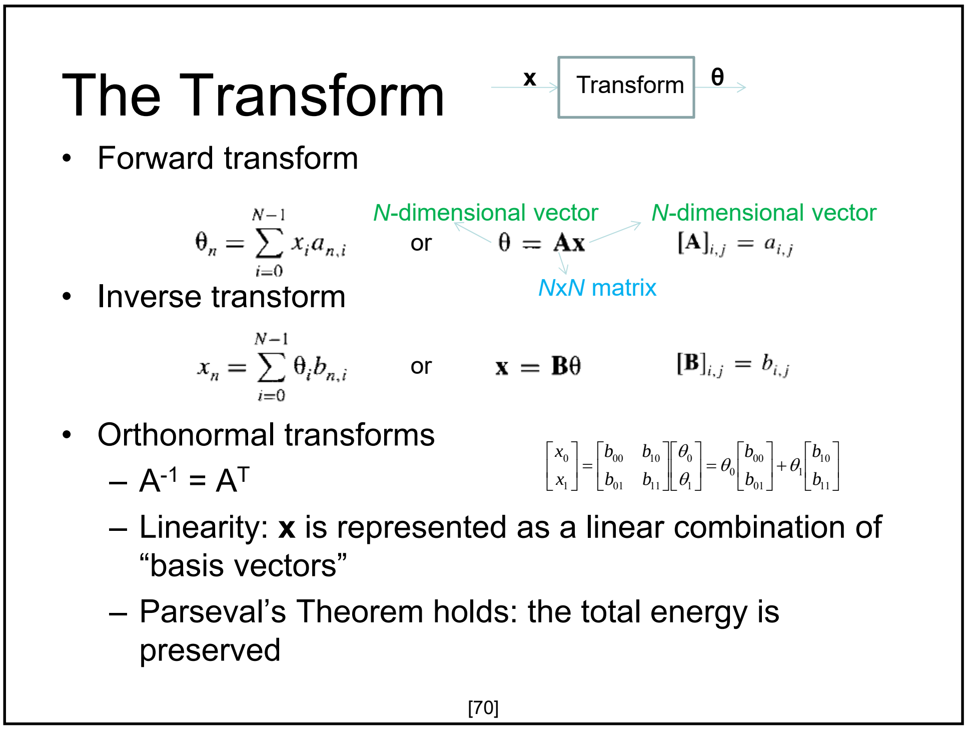


Transform Coding

- Three steps
  - Divide a data sequence into blocks of size *N* and transform each block using a reversible mapping
  - Quantize the transformed sequence
  - Encode the quantized values

Transform → Quantization → Binary coding →

[69]



The Transform

x → Transform → **θ**

- Forward transform

$$\theta_n = \sum_{i=0}^{N-1} x_i a_{n,i} \quad \text{or} \quad \theta = \mathbf{A}\mathbf{x} \quad [\mathbf{A}]_{i,j} = a_{i,j}$$

N-dimensional vector

N-dimensional vector

*N*x*N* matrix

- Inverse transform

$$x_n = \sum_{i=0}^{N-1} \theta_i b_{n,i} \quad \text{or} \quad \mathbf{x} = \mathbf{B}\theta \quad [\mathbf{B}]_{i,j} = b_{i,j}$$

- Orthonormal transforms

$$\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} b_{00} & b_{10} \\ b_{01} & b_{11} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \theta_0 \begin{bmatrix} b_{00} \\ b_{01} \end{bmatrix} + \theta_1 \begin{bmatrix} b_{10} \\ b_{11} \end{bmatrix}$$

  - $A^{-1} = A^T$
  - Linearity: **x** is represented as a linear combination of “basis vectors”
  - Parseval’s Theorem holds: the total energy is preserved

[70]

Two-dimensional transform

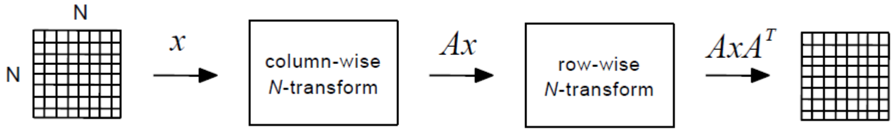
- Separable transforms
  - Take the transform along one dimension, then repeat the operation along the other direction
  - Ex: rows -> columns, or columns -> rows
- Forward transform

*N*x*N* matrix

*N*x*N* matrix

$$\Theta = \mathbf{AXA}^T$$

*N*x*N* matrix



- Backward transform

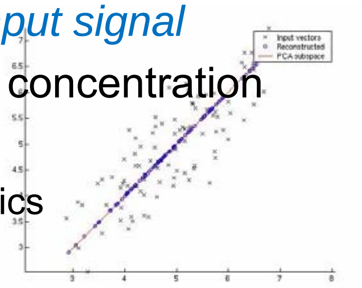
$$\mathbf{X} = \mathbf{B\Theta B}^T$$

Transforms of interest

- Data-dependent
  - Discrete Karhunen Lòeve transform (KLT)
- Data-independent
  - Discrete cosine transform (DCT)

KLT

- Also known as *Principal Component Analysis* (PCA), or the *Hotelling transform*
- Transforms *correlated* variables into *uncorrelated* variables
- Basis vectors are *eigenvectors of the covariance matrix of the input signal*
- Achieves *optimum* energy concentration
- Disadvantages
  - Dependent on signal statistics
  - Not separable



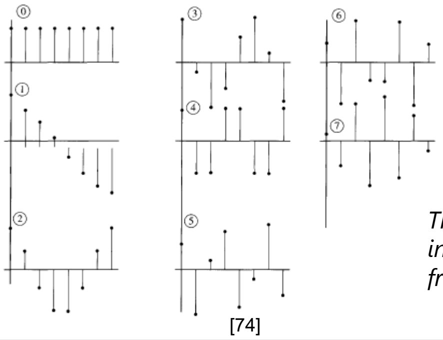
[73]

DCT (1)

- Part of many standards (JPEG, MPEG, H.261, …)
- Th

$$[\mathbf{C}]_{i,j} = \begin{cases} \sqrt{\frac{1}{N}}\cos\frac{(2j+1)i\pi}{2N} & i = 0,\ j = 0, 1, \ldots, N-1 \\ \sqrt{\frac{2}{N}}\cos\frac{(2j+1)i\pi}{2N} & i = 1, 2, \ldots, N-1,\ j = 0, 1, \ldots, N-1 \end{cases}$$

- Visualiz

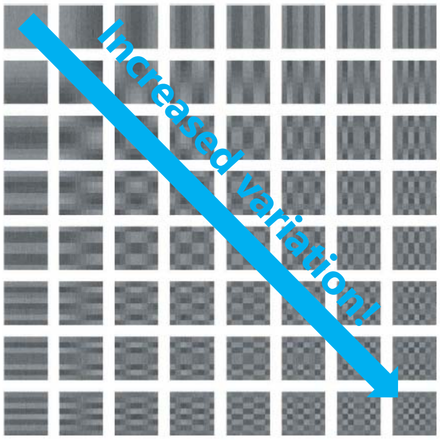


The frequency increases as we go from top to bottom!

[74]

DCT (2)

- 2-D basis matrices of DCT

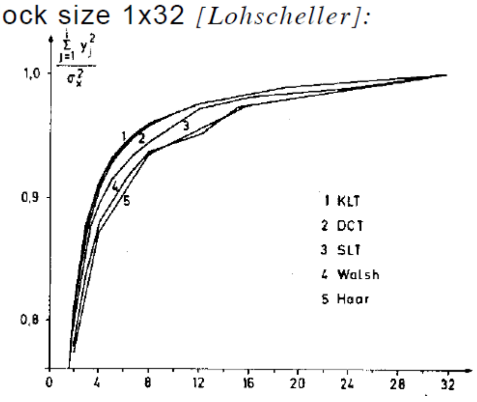


[75]

DCT (3)

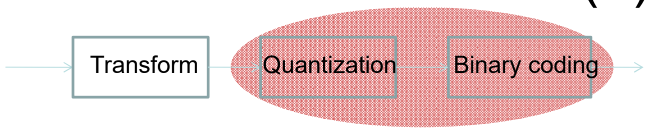
- Performs close to the optimum KLT in terms of compaction

Energy concentration measured for typical natural images, block size 1x32 *[Lohscheller]:*



[76]

Quantization and coding of transform coefficients (1)



- The bit-rate allocation problem
  - Divide bit-rate *R* among transform coefficients such that resulting distortion *D* is minimized

$$R = \sum_i R_i$$

Total rate

Rate for coefficient *i*

$$D = \sum_i D_i$$

Total distortion

Distortion contributed by coefficient *i*

[77]

Quantization and coding (2)

- The optimal bit allocation

$$R_k = R + \frac{1}{2}\log_2 \frac{\sigma^2_{\theta_k}}{\prod_{k=1}^{M}(\sigma^2_{\theta_k})^{\frac{1}{M}}}$$

variance of the transform coefficient θ_k

- The zonal sampling
  - Compute $\sigma^2_{\theta k}$ for each coeffici
  - Set $R_k = 0$ for all k
  - Sort $\{\sigma^2_{\theta k}\}$. Suppose $\sigma^2_{\theta 1}$ is the
  - Increment R_1 by 1, divide $\sigma^2_{\theta 1}$
  - Decrement R_b by 1.
    - If Rb=0, stop;
    - Otherwise, go to step 3

R_b = 72 bits

8	7	5	3	1	1	0	0
7	5	3	2	1	0	0	0
4	3	2	1	1	0	0	0
3	3	2	1	1	0	0	0
2	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

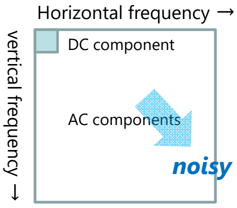
Discarded!

8x8 DCT of an image

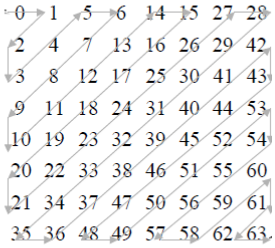
[78]

Quantization and coding (3)

- The threshold coding
  - Transform coefficients that fall below a threshold are discarded
  - Example: a 8x8 block

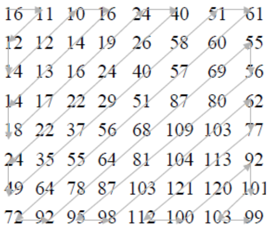


The zigzag scan



0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

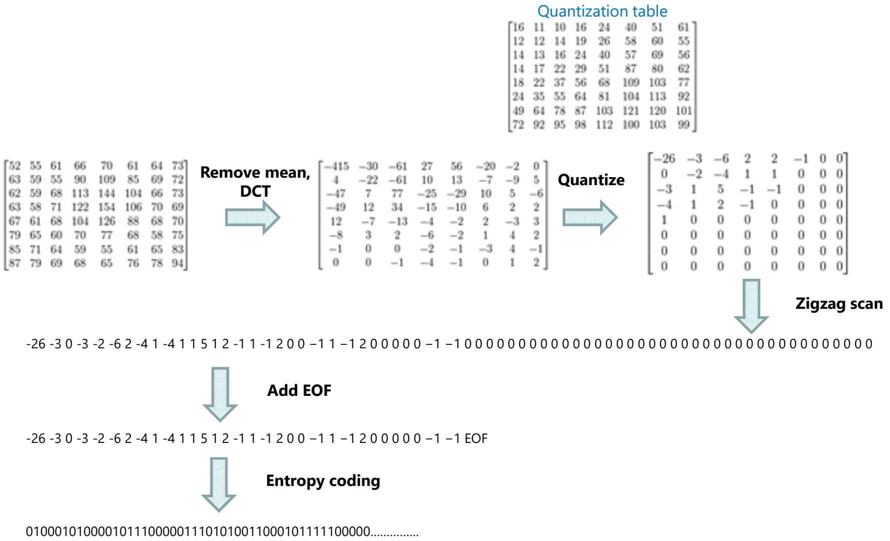
Sample quantization table



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

[79]

Example: Encode



Quantization table

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

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Remove mean, DCT

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

Quantize

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Zigzag scan

-26 -3 0 -3 -2 -6 2 -4 1 -4 1 1 5 1 2 -1 1 -1 2 0 0 −1 1 −1 2 0 0 0 0 0 −1 −1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Add EOF

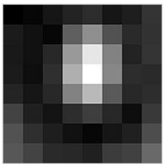
-26 -3 0 -3 -2 -6 2 -4 1 -4 1 1 5 1 2 -1 1 -1 2 0 0 −1 1 −1 2 0 0 0 0 0 −1 −1 EOF

Entropy coding

0100010100001011100000111010100110001011111100000..............

[80]

Example: Decode



Original block

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

0100010100001011100000111010100110001011111100000..............

Entropy decoding

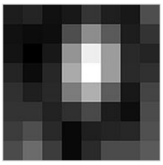
-26 -3 0 -3 -2 -6 2 -4 1 -4 1 1 5 1 2 -1 1 -1 2 0 0 –1 1 –1 2 0 0 0 0 0 –1 –1 EOF

Put into a block

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{Q^{-1}} \begin{bmatrix} -416 & -33 & -60 & 32 & 48 & -40 & 0 & 0 \\ 0 & -24 & -56 & 19 & 26 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -56 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{Inverse DCT, Add mean}}$$

Reconstructed block

$$\begin{bmatrix} 60 & 63 & 55 & 58 & 70 & 61 & 58 & 80 \\ 58 & 56 & 56 & 83 & 108 & 88 & 63 & 71 \\ 60 & 52 & 62 & 113 & 150 & 116 & 70 & 67 \\ 66 & 56 & 68 & 122 & 156 & 116 & 69 & 72 \\ 69 & 62 & 65 & 100 & 120 & 86 & 59 & 76 \\ 68 & 68 & 61 & 68 & 78 & 60 & 53 & 78 \\ 74 & 82 & 67 & 54 & 63 & 64 & 65 & 83 \\ 83 & 96 & 77 & 56 & 70 & 83 & 83 & 89 \end{bmatrix}$$



Quantization table

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

[81]

One more example

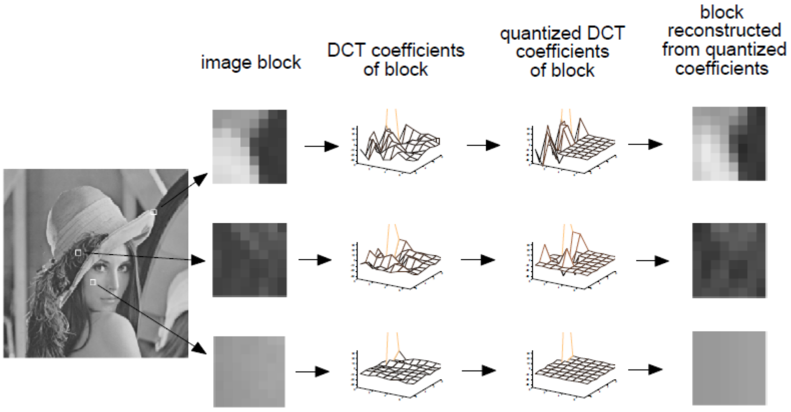


image block
DCT coefficients of block
quantized DCT coefficients of block
block reconstructed from quantized coefficients

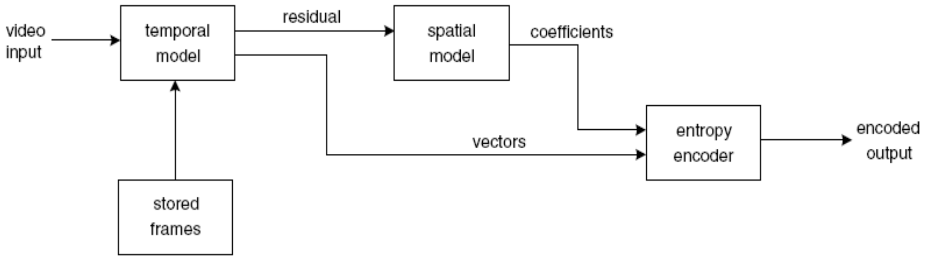
[82]

Introduction to Video compression algorithm

Video compression algorithm

- Temporal compression:
  - continuous frames look similar, only the changes from one frame to the next need to be encoded.
- Spatial compression:
  - Sensitivity of the human visual system.

Video compression algorithm



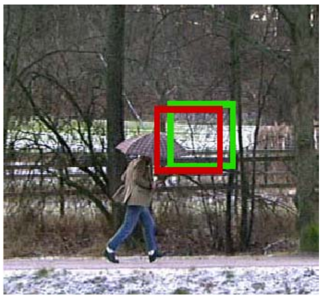
Video encoder block diagram

[85]

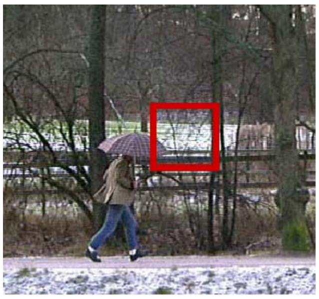
Temporal compression

Block-matching algorithm

Previous Image



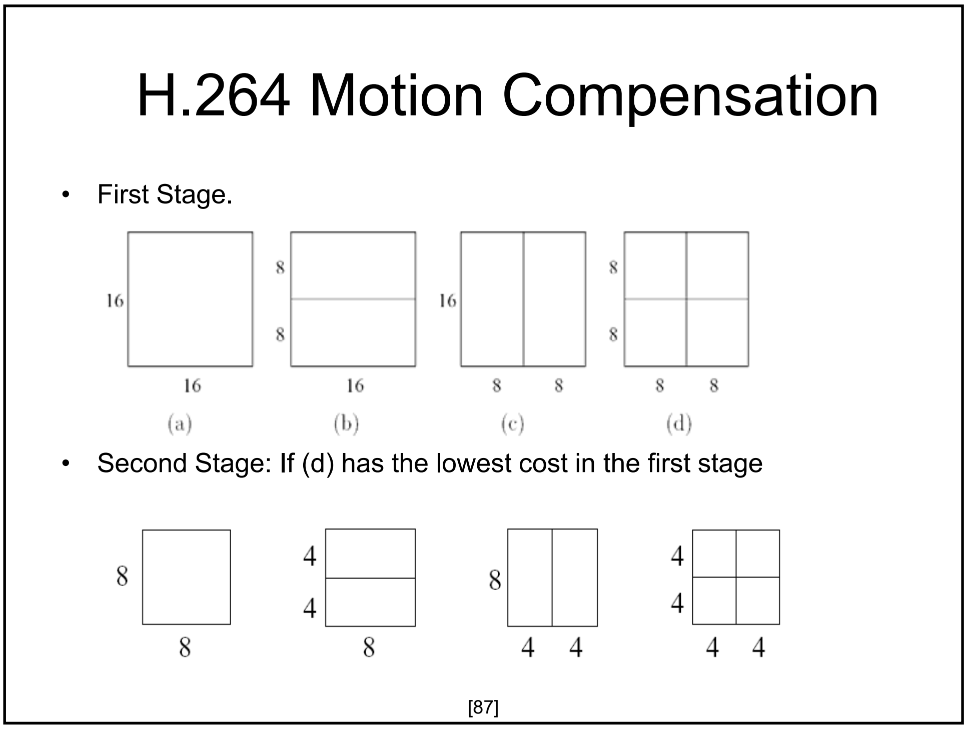
Current Image



Measurement window is compared with a shifted array of pixels in the other image, to determine the best match

Rectangular array of pixels is selected as a measurement window

[86]

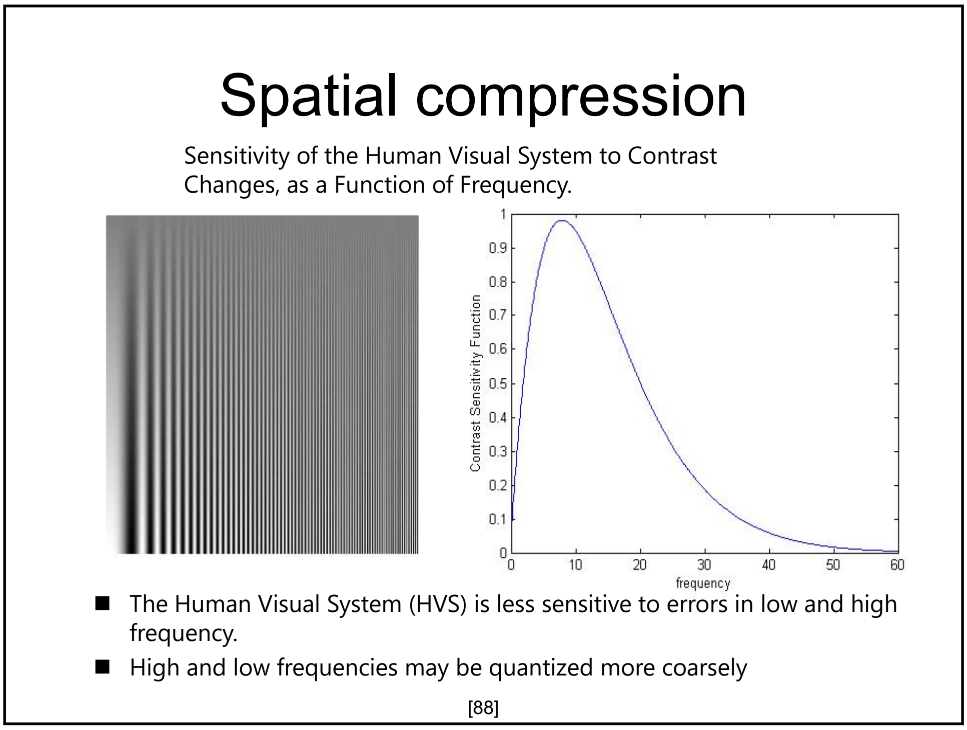


H.264 Motion Compensation

- First Stage.

- Second Stage: If (d) has the lowest cost in the first stage

[87]



Spatial compression

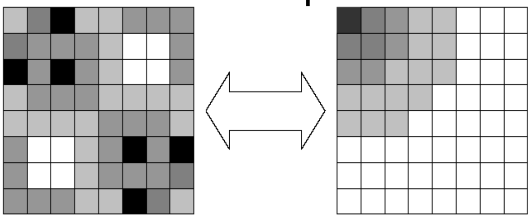
Sensitivity of the Human Visual System to Contrast Changes, as a Function of Frequency.

- The Human Visual System (HVS) is less sensitive to errors in low and high frequency.
- High and low frequencies may be quantized more coarsely

[88]

Spatial compression

8x8 DCT operation



(a) Original picture.

(b) Corresponding DCT mapping of (a)

- The Discrete Cosine Transform (DCT) :
  - decomposes the signal into spatial frequencies, which then allow further processing techniques to reduce the precision of the DCT coefficients consistent with the Human Visual System (HVS) model.
  - concentrates most energy in a few bins (e.g., lots of coefficients near zero).
- Larger blocks → more computation
- Smaller blocks → transform stage less effective

Optimal Tiling for Video Compression

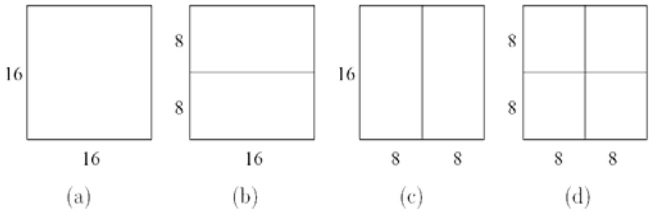
- Key ideas
  - – Use larger dictionaries of tilings than in H.264.
  - – Use recursive tree pruning algorithms to efficiently find the best rectangular tiling for an image.
  - – Apply such algorithms to two stages of video compression.
    - Motion Compensation.
    - Transform.

- Achievements
  - Algorithm I:
    - Use larger dictionary for motion compensation.
    - Our tiling selection method results in up to 19% savings in bit rate as compared to H.264 tiling selection.
  - Algorithm II:
    - Use larger dictionary for both motion compensation and transform stages.
    - Our tiling selection method results in up to 23% savings in bit rate as compared to H.264 tiling selection.

[91]

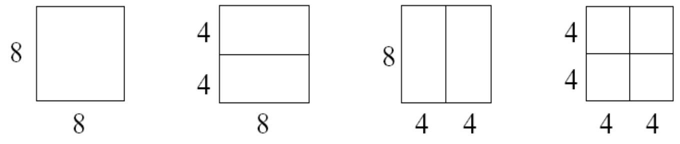
H.264 Motion Compensation

- First Stage.



	H.264
# tilings	259
# tiles	41

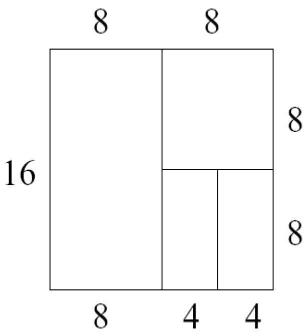
- Second Stage: If (d) has the lowest cost in the first stage



[92]

Dyadic Tilings

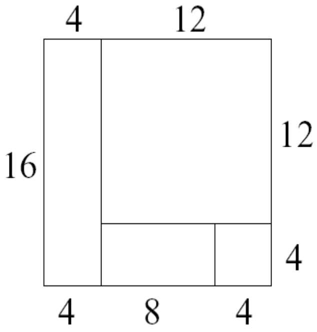
- In addition, allow all dyadic two-way splits down to 4 x 4 tiles. For example,



	Dyadic
# tilings	6857
# tiles	49

Multitree Tilings

- In addition, allow all two-way splits down to 4x4 tiles. For example,



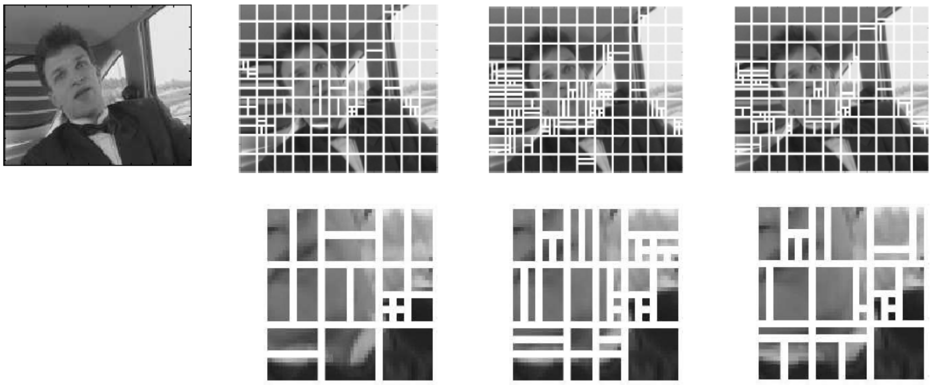
	Multitree
# tilings	68480
# tiles	100

Number of Tilings and Tiles for a 16X16 Block

	H.264	Dyadic	Multitree
# tilings	259	6857	68480
# tiles	41	49	100

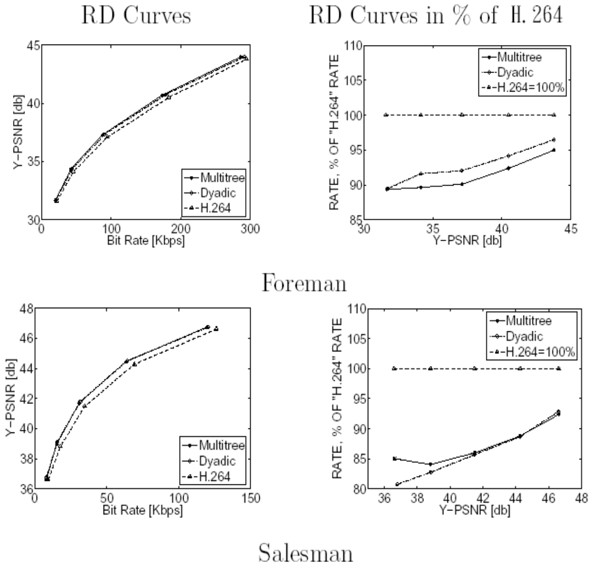
- The H.264 set of tilings is a small subset of the dyadic and multitree tilings.
- The number of tiles, and therefore complexity, increases much more slowly.

Examples--Carphone Sequence



Original H.264 Dyadic Multitree

Results: Fixed 4x4 Transform


RD Curves

RD Curves in % of H. 264

Foreman

Salesman

[97]

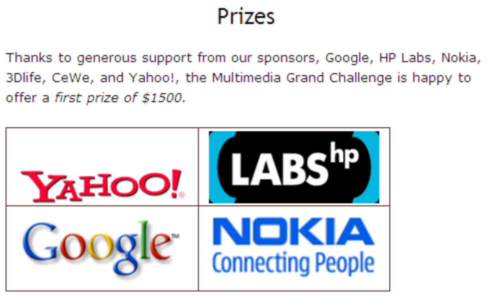
Conclusion

- The use of finer tilings result in more overhead bits.
- We show, however, that our optimal tiling search is so effective that the increase in the number of overhead bits is more than compensated for by the reduction in the number of bits require to encode the motion vectors and transform coefficients.

[98]

Some multimedia research topics

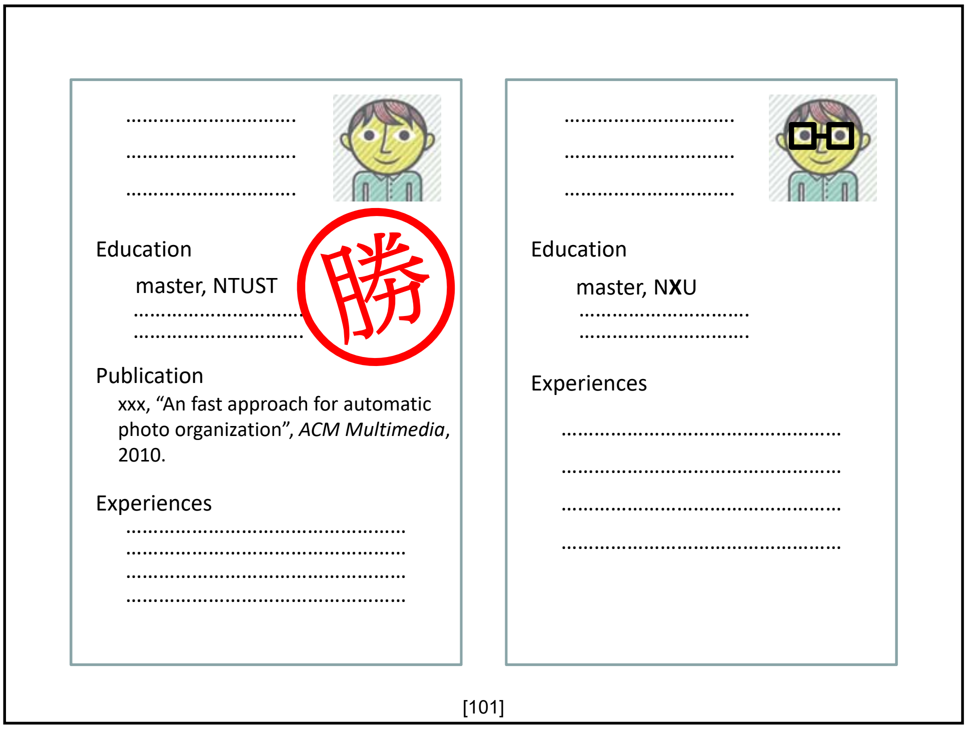
Here comes the opportunity!



Prizes

Thanks to generous support from our sponsors, Google, HP Labs, Nokia, 3Dlife, CeWe, and Yahoo!, the Multimedia Grand Challenge is happy to offer a *first prize of $1500.*

- $1500 (~NT47688) if we earn the first prize
- Make your resume stand out



……………………………
……………………………
……………………………

Education

master, NTUST
……………………………
……………………………

勝

Publication

xxx, "An fast approach for automatic photo organization", *ACM Multimedia*, 2010.

Experiences

……………………………………………
……………………………………………
……………………………………………
……………………………………………

……………………………
……………………………
……………………………

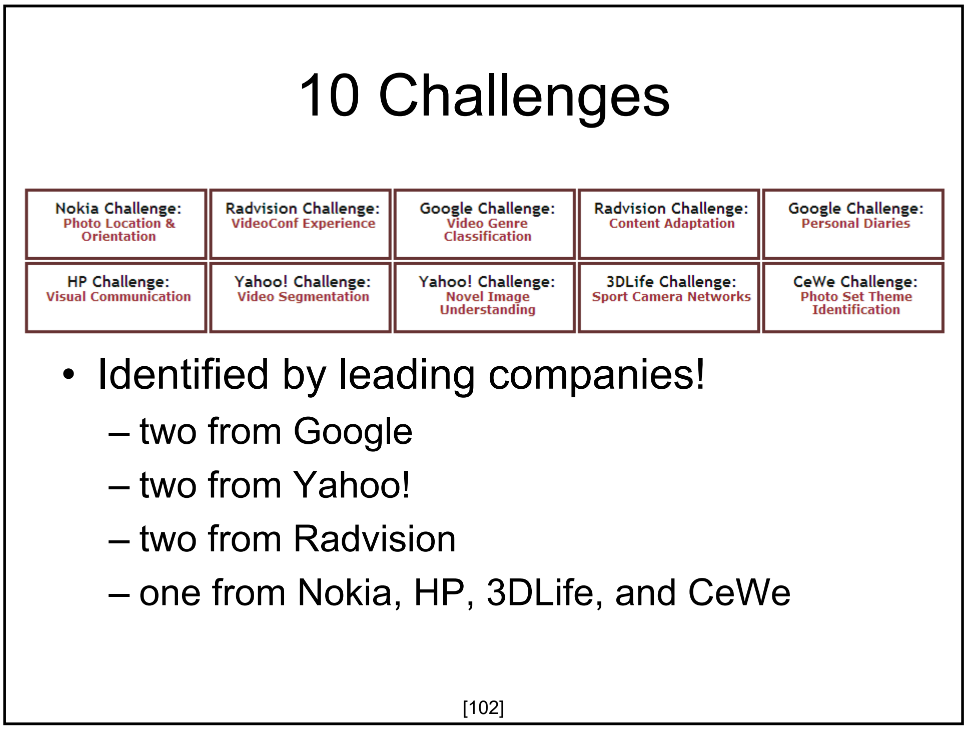
Education

master, N**X**U
……………………………
……………………………

Experiences

……………………………………………
……………………………………………
……………………………………………
……………………………………………

[101]



10 Challenges

Nokia Challenge: Photo Location & Orientation	Radvision Challenge: VideoConf Experience	Google Challenge: Video Genre Classification	Radvision Challenge: Content Adaptation	Google Challenge: Personal Diaries
HP Challenge: Visual Communication	Yahoo! Challenge: Video Segmentation	Yahoo! Challenge: Novel Image Understanding	3DLife Challenge: Sport Camera Networks	CeWe Challenge: Photo Set Theme Identification

- Identified by leading companies!
  - two from Google
  - two from Yahoo!
  - two from Radvision
  - one from Nokia, HP, 3DLife, and CeWe

[102]

Can you tell where are these photos taken?

[103]

Nokia Challenge

- Goal
  - Try to derive exact camera poses (***location*** and ***orientation***) of given photos that are lacking location annotation
- You can assume the availability of nearby photos/video with known location that can be used to derive unknown camera poses; other ideas that do not require existing content will be welcome.
- Applications
  - http://betalabs.nokia.com/apps/nokia-image-space

[104]

<geolocation alt="0" lat="43.731639899896" lng="7.421372230007" />
<orientation pitch="0" roll="0" yaw="104" />

緯度, 經度

<geolocation alt="0" lat="43.731634200204" lng="7.421395112594" />
<orientation pitch="0" roll="0" yaw="100" />

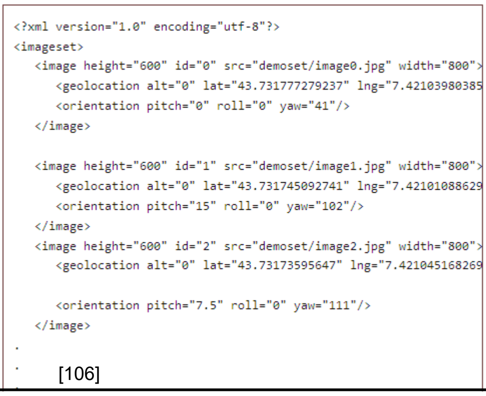
<geolocation alt="0" lat="43.731594553817" lng="7.421487732589" />
<orientation pitch="0" roll="0" yaw="276" />

<geolocation alt="0" lat="43.731658423895" lng="7.421072325625" />
<orientation pitch="7.5" roll="0" yaw="240" />

[105]

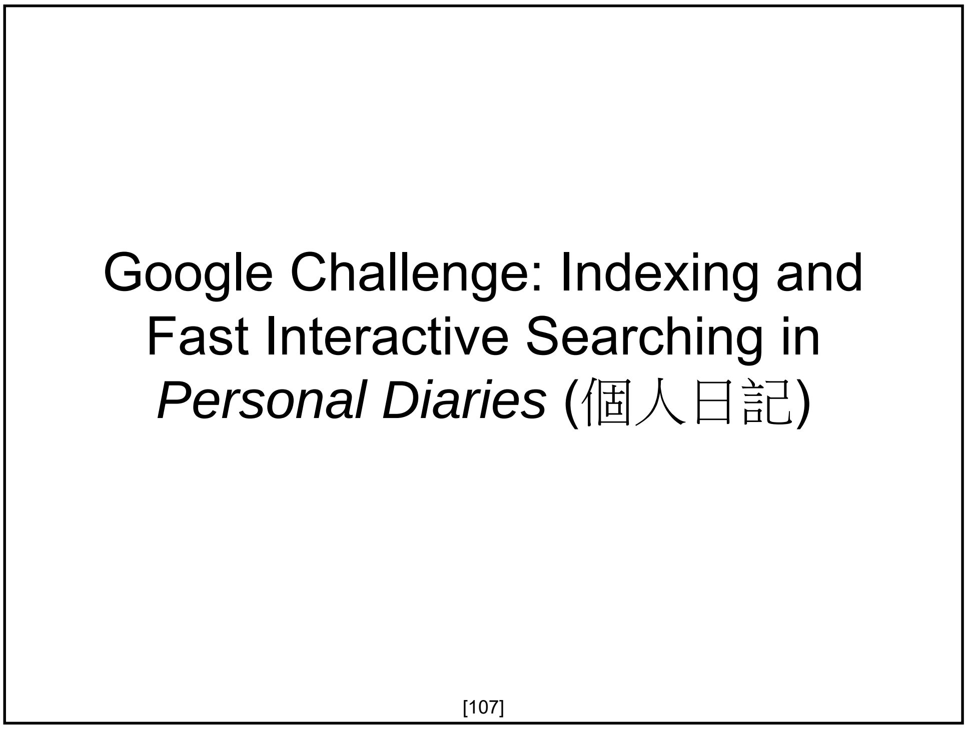
Nokia Challenge

- Matlab tools and datasets are available!
  - photos captured with the Nokia 6210 Navigator
  - each image has an associated GPS/orientation measurement



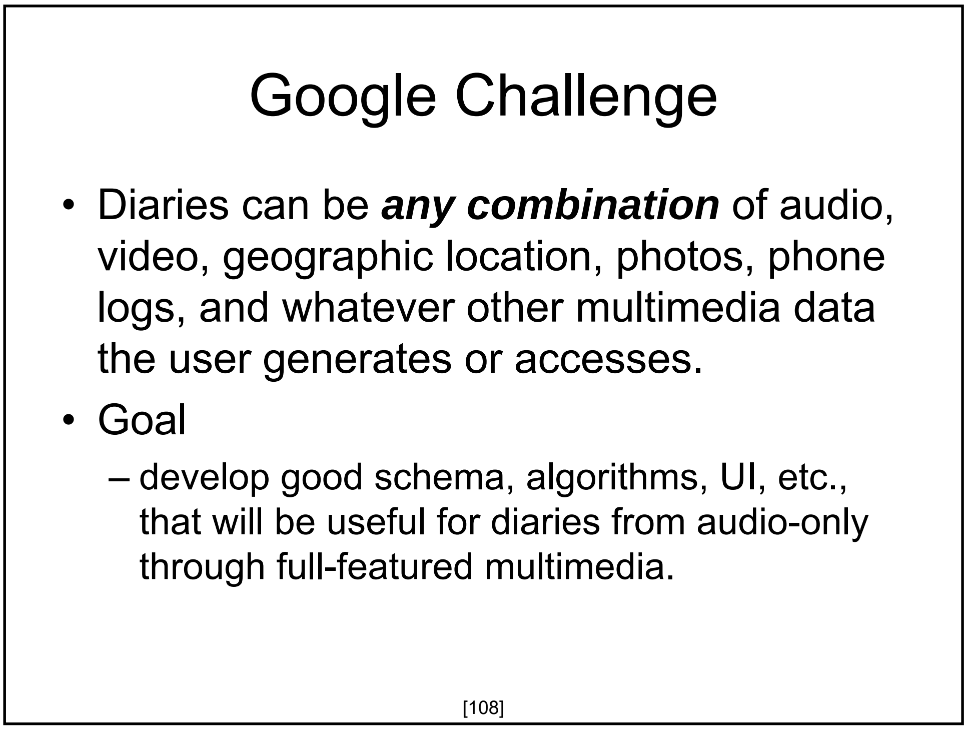
```
<?xml version="1.0" encoding="utf-8"?>
<imageset>
   <image height="600" id="0" src="demoset/image0.jpg" width="800">
      <geolocation alt="0" lat="43.731777279237" lng="7.42103980385
      <orientation pitch="0" roll="0" yaw="41"/>
   </image>

   <image height="600" id="1" src="demoset/image1.jpg" width="800">
      <geolocation alt="0" lat="43.731745092741" lng="7.42101088629
      <orientation pitch="15" roll="0" yaw="102"/>
   </image>
   <image height="600" id="2" src="demoset/image2.jpg" width="800">
      <geolocation alt="0" lat="43.73173595647" lng="7.421045168269

      <orientation pitch="7.5" roll="0" yaw="111"/>
   </image>
.
.
.
```

[106]



Google Challenge: Indexing and Fast Interactive Searching in *Personal Diaries* (個人日記)

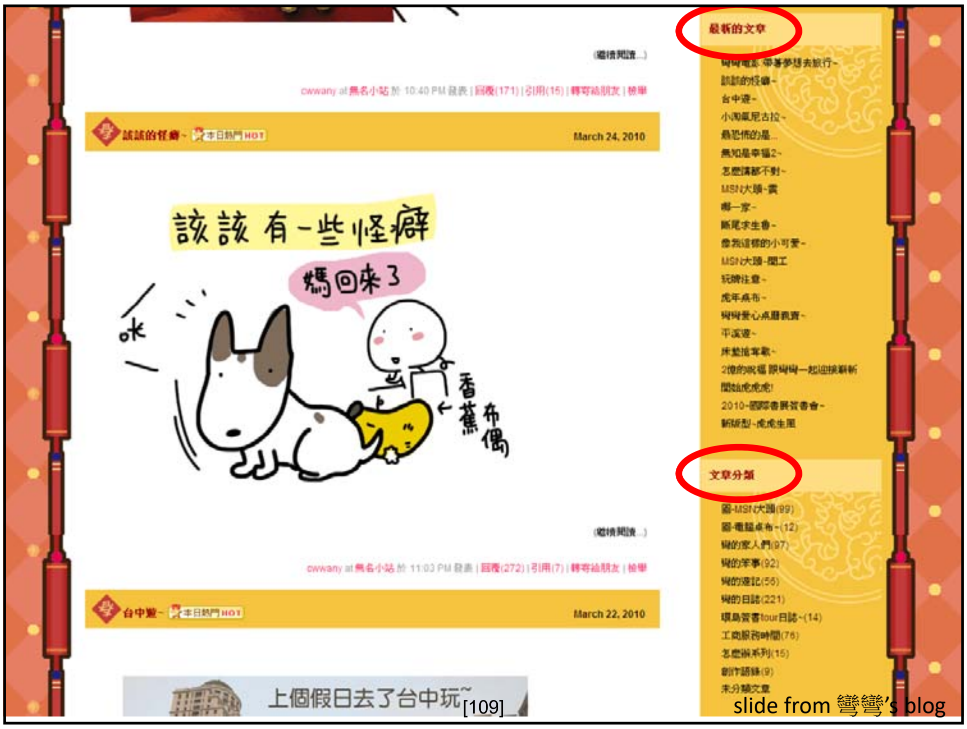
[107]



Google Challenge

- Diaries can be ***any combination*** of audio, video, geographic location, photos, phone logs, and whatever other multimedia data the user generates or accesses.
- Goal
  - develop good schema, algorithms, UI, etc., that will be useful for diaries from audio-only through full-featured multimedia.

[108]



(繼續閱讀...)

cwwany at 無名小站 於 10:40 PM 發表 | 回應(171) | 引用(15) | 轉寄給朋友 | 檢舉

該該的怪癖~ 本日熱門 HOT March 24, 2010

(繼續閱讀...)

cwwany at 無名小站 於 11:03 PM 發表 | 回應(272) | 引用(7) | 轉寄給朋友 | 檢舉

台中遊~ 本日熱門 HOT March 22, 2010

上個假日去了台中玩~

最新的文章

咩咩電影-帶著夢想去旅行~
該該的怪癖~
台中遊~
小淘氣尼古拉~
最恐怖的是...
無知是幸福2~
怎麼講都不對~
MSN大頭-雲
哪一家~
斷尾求生魯~
像我這樣的小可愛~
MSN大頭-園工
玩牌注意~
虎年桌布~
咩咩愛心桌曆義賣~
平溪遊~
床墊搶客戰~
2億的祝福 跟咩咩一起迎接嶄新
開始虎虎虎!
2010-國際書展簽書會~
新版型-虎虎生風

文章分類

圖-MSN大頭(99)
圖-電腦桌布~(12)
咩的家人們(97)
咩的笨事(92)
咩的遊記(55)
咩的日誌(221)
環島簽書tour日誌~(14)
工商服務時間(76)
怎麼辦系列(15)
創作語錄(9)
未分類文章

[109]

slide from 彎彎's blog

Google Challenge

- Some thoughts:
  - What would be a good design for blogs?
  - We now have photos, comments, game records, … on facebook. Could we design new tools for users to create their own "e-diary"?

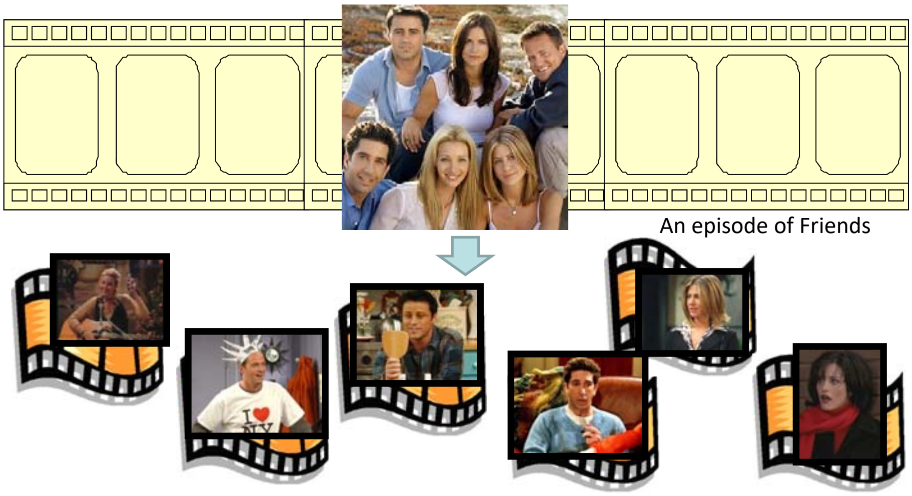
[110]

Yahoo! Challenge: Robust Automatic Segmentation of Video According to Narrative Themes

故事形式的

[111]

One example:



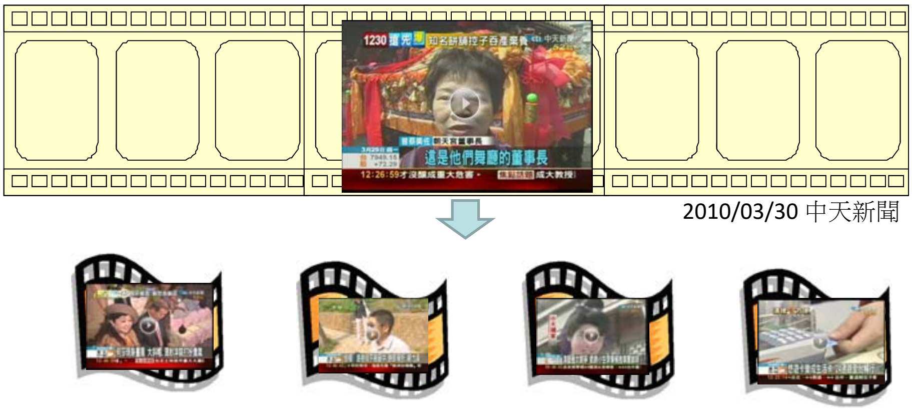
[112]

Yahoo! Challenge

- Goal
  - develop methods, techniques, and algorithms to automatically generate ***narrative themes*** for a given video, as well as present the content in an easy-to-consume manner to end-users in a search engine experience
- Applications
  - allow users consume pieces of a video that would be of interest to them
  - let users kill time during lunch breaks in creative ways

[113]

Another example:



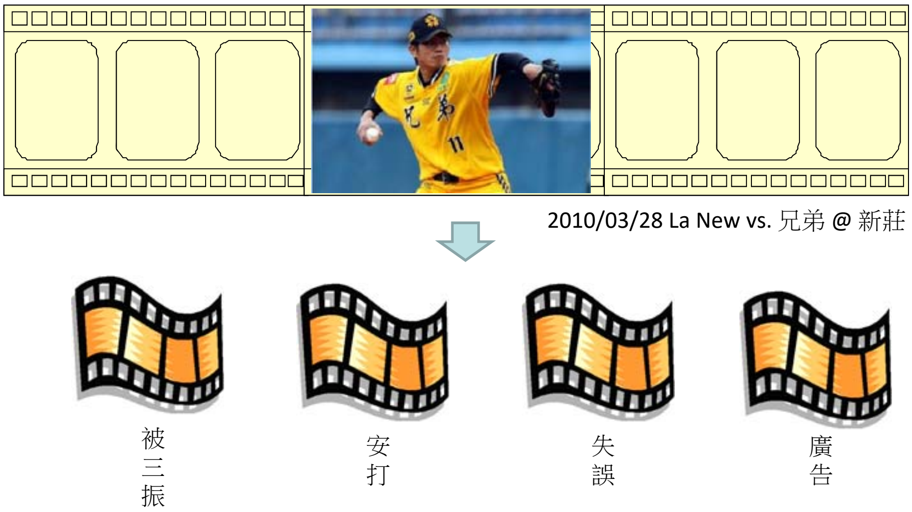
2010/03/30 中天新聞

政治・財經 社會・地方 影視・體育 生活・醫療

1. Train a 馬英九總統 detector, use it to identify video segments, and tag them with “政治”
2. Identify the sound of the anchor man (主播), from there identify keywords such as “娛樂”, “社會”, etc.
3. …..

[114]

One more example:



2010/03/28 La New vs. 兄弟 @ 新莊

1. Background music is different, tag the video segment “廣告”
2. ….

[115]

- My suggestions
  - select a certain type of videos
  - Examples:
    - sitcom or some popular tv show such as 康熙來了
    - your favorite movie
    - a wedding video of your family (進場, 致詞, 敬酒, 玩遊戲…)
    - sport videos
    - educational videos
    - …
- You need to specify not only the themes, but also how your are going to segment the input video into themes *automatically*

[116]

Yahoo! Challenge: Novel Image Understanding

Yahoo! Challenge

- Move beyond simple image classification
- Goal
  - develop novel and useful ways to organize and structure image content

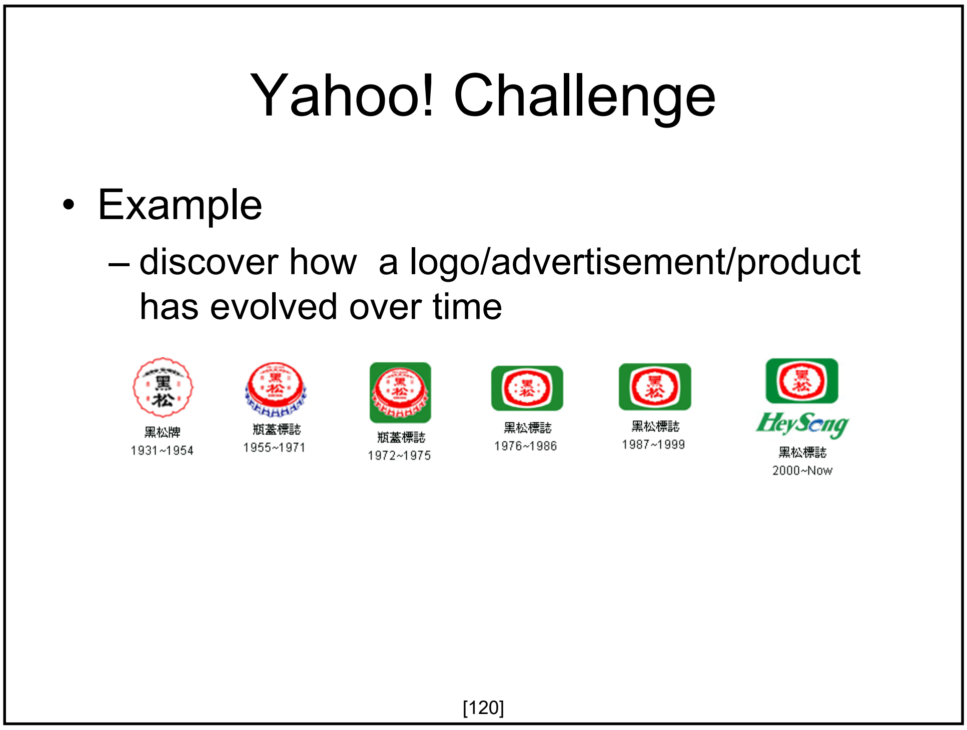
Yahoo! Challenge

- Example
  - Sort celebrity pictures by their subject's age

or hair style

1999 2000 2001 2002…

[119]



Yahoo! Challenge

- Example
  - discover how a logo/advertisement/product has evolved over time

[120]

Yahoo! Challenge

- There are many ways to organize photos!
  - What are the ways that are not obvious?
  - What can we do better than we can do today?

HP Challenge: High Impact Visual Communication

HP Challenge

- Goal
  - find a solution which can create ***a collage*** (美術拼貼) and generate ***a textual description*** that tells the story of a set of photos
- How do we create a high impact picture that can convey information across cultural boundaries and find a thousand words that best describe such a picture?

[123]

HP Challenge

- Input
  - a digital photo collection, such as photos taken during a vacation
- Outputs
  - the most appealing collage picture (1600x1200 pixels) that best represents the original collection
  - a description (<100 words) of the collage picture

[124]

A show at xxx pub. Hang out with my friends….

my favorite cartoon: Sponge bob and square pants….

HP Challenge

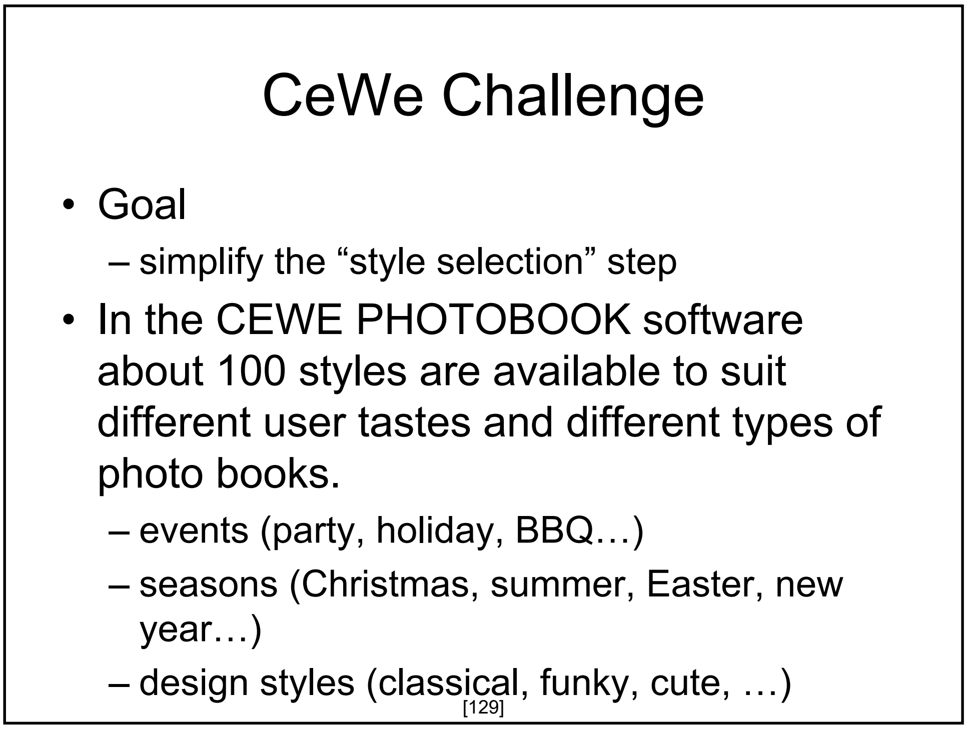
- 6 datasets are available!
  - – each with 20 photos

CeWe Challenge: Automatic Theme Identification of Photo Sets for Digital Print Products

CeWe Challenge

- About CeWe
  - the Number One services partner for first-class trade brands on the European photographic market
  - supplies both stores and Internet retailers with photographic products

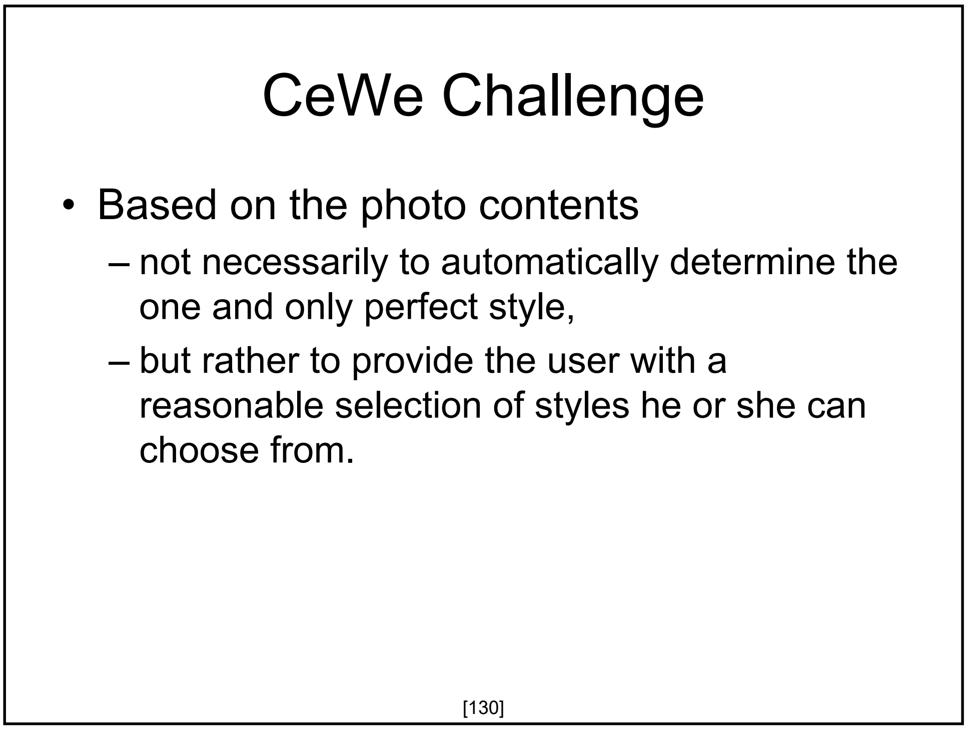
posters, calendars or ***photo books***



CeWe Challenge

- Goal
  - simplify the “style selection” step
- In the CEWE PHOTOBOOK software about 100 styles are available to suit different user tastes and different types of photo books.
  - events (party, holiday, BBQ…)
  - seasons (Christmas, summer, Easter, new year…)
  - design styles (classical, funky, cute, …)

[129]



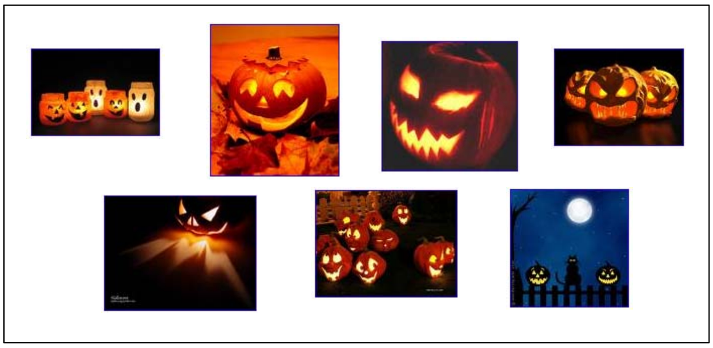
CeWe Challenge

- Based on the photo contents
  - not necessarily to automatically determine the one and only perfect style,
  - but rather to provide the user with a reasonable selection of styles he or she can choose from.

[130]

CeWe Challenge

- Example
  - What would be good background colors for the set of photos?



[131]

Radvision Challenge: Video Conferencing To Surpass “In-Person” Meeting Experience

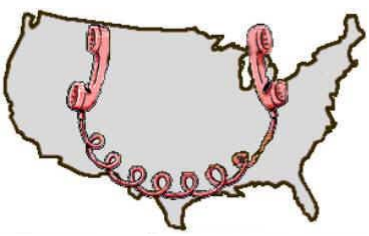
[132]

Radvision Challenge

- Goal
  - – developing new technologies and ideas to surpass (超越, 改善) the “in-person” meeting experience

Radvision Challenge

- Example:
  - – Come up with ways to maintain a long-term relationship



For more information…

ACMmultimedia25

http://www.acmmm.org/2017/

[135]