ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Multimedia Technology

Section # 5: Principles of Coding

Instructor: George Xylomenos

Department: Informatics

Contents

- Coding requirements
- Types of coding
- Entropy coding
- Differential / predictive coding
- Transform coding
- Layered coding
- Vector quantization

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Coding requirements

Class: Multimedia Technology, **Section # 5:** Principles of Coding

Instructor: George Xylomenos, **Department:** Informatics

Basic requirements (1 of 4)

- Uncompressed data volume for media
  - Voice (telephony): 64 Kbps (8 KHz, 8 bits)
  - Audio (CD): 705,6x2 Kbps (44,1 KHz, 16 bits)
  - Video (CCIR-601): 216 Mbps (13,5 MHz, 4:2:2, 8 bits)
    - Without blanking intervals: 166 Mbps
- Compression and decompression
  - Storage to disks
  - Transmission via network
  - Caching in memory

Basic requirements (2 of 4)

- Desirable features
  - High fidelity
    - Lowering fidelity allows more compression
  - Low complexity
    - Impacts cost of implementation
    - What is “low” changes over time
  - Low latency (delay)
    - Important in real-time communication

Basic requirements (3 of 4)

- Requirements of synchronous applications
  - Compression/decompression latency < 50 ms
    - End-to-end latency < 150 ms for telephony
    - Much (much!) lower for music performance
- Requirements of interactive applications
  - Quick repositioning (forward / backward)
  - Random access in <500 ms
  - Decompression with no additional data

Basic requirements (4 of 4)

- General requirements
  - Flexible formats
  - Different bitrates
    - Ideally, variable bitrates
  - Synchronization points between media
  - Software or hardware decoding
    - Depending on the application
  - Standardized techniques
    - Allowing mass production

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Types of coding

Class: Multimedia Technology, **Section # 5:** Principles of Coding

Instructor: George Xylomenos, **Department:** Informatics

Categorization

- Lossless
  - Information can be fully recovered (FLAC)
- Lossy
  - Controlled loss of information (MP3)
- Symmetric
  - Same cost for compression / decompression
- Asymmetric
  - Almost always, decompression is simpler

Types of coding (1 of 2)

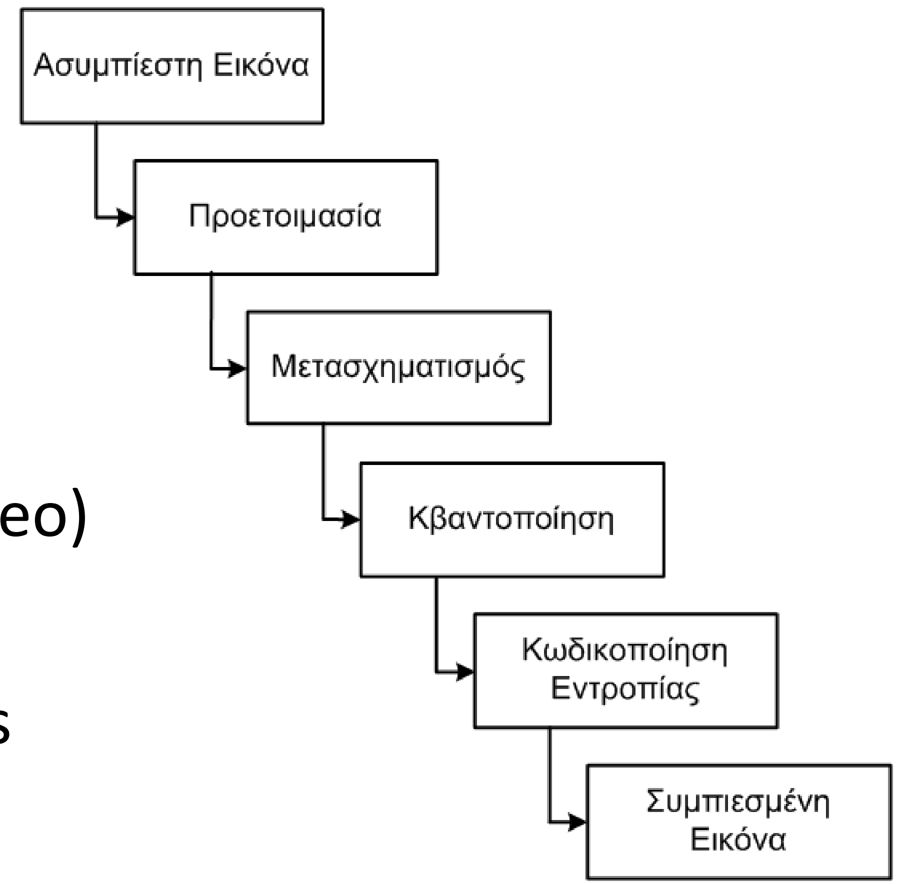
- Entropy coding
  - Disregards media content
  - Lossless
- Source coding
  - Understands media content
  - Lossy or lossless
- Hybrid coding
  - Usually, first source and then entropy coding

Types of coding (2 of 2)

Category	Example
Entropy	Run Length
	Huffman
	Arithmetic
	LZx
Source	Differential/predictive
	Transform
	Layered
	Vector quantization
Hybrid	JPEG
	MPEG-x
	H.26x

Coding flow (JPEG)

- Preparation
  - RGB to YUV
- Processing
  - DCT/FFT (transform)
  - Motion prediction (in video)
- Quantization
  - To (fewer) discrete values
- Entropy coding
  - Lossless compression of resulting flow



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Entropy coding

Class: Multimedia Technology, **Section # 5:** Principles of Coding

Instructor: George Xylomenos, **Department:** Informatics

Removing redundancy (1 of 3)

- Run length encoding (RLE)
  - Replaces sequences of the same symbol
  - Flag “~”, run length – 4, Byte
    - Does not make sense for smaller runs
  - Byte stuffing if the flag is in the data (~~)
  - Good (mostly) for drawings
    - Long runs of pixels with the same color
    - Very rare in photographs

Removing redundancy (2 of 3)

- Example
  - Input: BBBACCCCCC~A
  - Output: BBBA~2C~~A
- May even grow the file!
  - Few long runs, many appearances of ~
- Also possible with symbol groups
  - As long as there is periodicity
  - Example: RGB values (triples)

Removing redundancy (3 of 3)

- Coding of frequent words
  - Replaced by flag and word index
  - Needs a dictionary of common words
  - Good for programming language
    - Part of the input is unpredictable
    - But we have language keywords
    - May also exploit symbol table
      - E.g., variables and methods

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

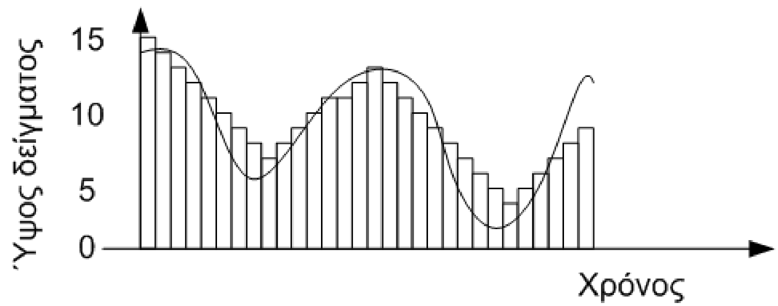
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Differential / predictive coding

Class: Multimedia Technology, **Section # 5:** Principles of Coding

Instructor: George Xylomenos, **Department:** Informatics

Differential coding (1 of 4)



- Encodes difference between values
  - Appropriate when values change slowly
    - Example: audio samples
  - Small differences between values…
  - …large differences from zero

Differential coding (2 of 4)

- Delta coding
  - One bit per value (+1 or –1)
- Non-adaptive differential coding
  - Assume that we are limited to 4 bits per value
  - How can we best use them?
  - Encode small differences: [-8,+7] in single steps
  - Encode large differences: [-128,+127] in 16 steps
  - Either way, there may be some loss

Differential coding (3 of 4)

- Why is it also called predictive?
  - Every value is a predictor for the next one
  - We encode the change from the prediction
- Generalized in various ways
  - Adaptive to large/small changes
  - Use multiple values as predictor
- Coding requires memory
  - One or more past values for prediction
  - Same for decoding

Differential coding (4 of 4)

- Synchronization of predictors
  - Encoding may lose some information
    - The available bits may not be enough
    - The encoded value is distorted
  - The decoder only sees the encoded values
    - Its prediction is based on the distorted value
  - The encoder must also decode the values!
    - To make predictions based on the distorted ones

ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Transform coding

Class: Multimedia Technology, **Section # 5:** Principles of Coding
Instructor: George Xylomenos, **Department:** Informatics

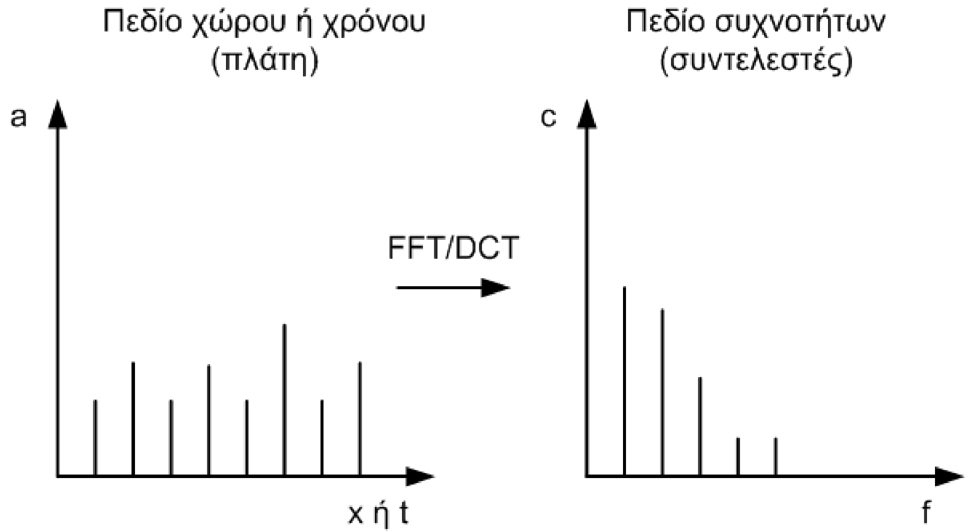
Why transforms?

- Transform an input field to an output field
  - Reversed transform for decoding
  - Example: FFT, DCT (intensities to frequencies)
  - The transform does NOT compress data
  - But the new field is better for compression
    - Separates more from less important values
    - More loss for less important values
    - Less loss for more important values

Fast Fourier Transform

- Fast Fourier Transform (FFT)
  – Input: signal at each time point f(t)
  – Output: coefficient of each frequency g(l)
  – Breaks down f(t) to harmonics weighed by g(l)
  – Isolates most important coefficients
    - They are all at one end of the spectrum
  – Allows compression based on importance
    - Less loss for more important coefficients

Discrete Cosine Transform



- Discrete Cosine Transform (DCT)
  - Input: intensity at each point f(x,y)
  - Output: frequency coefficients g(s,y)
    - Two-dimensional transform
  - Used in JPEG and video

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Layered coding

Class: Multimedia Technology, **Section # 5:** Principles of Coding

Instructor: George Xylomenos, **Department:** Informatics

Layers (1 of 2)

- Layered coding
  - Splits information in multiple layers
  - Each layer is treated differently
- Sub-band coding
  - Splits signal to frequency sub-bands
  - Each sub-band is encoded separately
  - Approximates signal with a few sub-bands

Layers (2 of 2)

- Sub-sampling
  - Different sample rate for each component
    - More important components -> more samples
  - Approximates signal with basic components
- Heterogeneity
  - Each layer sent independently
  - Better approximation with more layers
  - Receiver chooses which layers to receive

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Vector quantization

Class: Multimedia Technology, **Section # 5:** Principles of Coding

Instructor: George Xylomenos, **Department:** Informatics

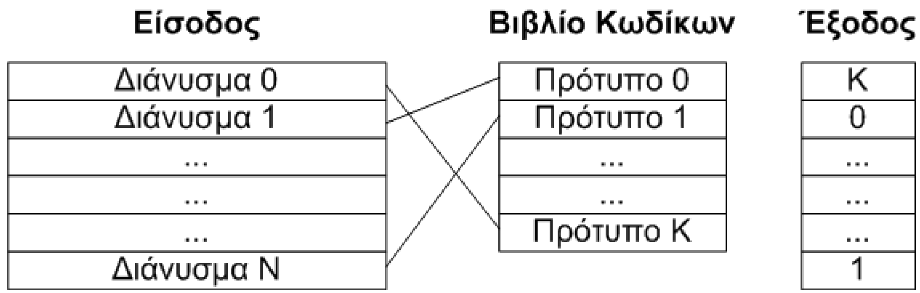
Code book (1 of 2)

- Code book contains vectors
  - Vectors: representative bit sequences
  - Encoding matches inputs to vector indexes
  - "Quantizes" input to closest matching vectors
- A good code book needs
  - Few code words
    - So that indexes are small
  - Code words that match inputs well
    - Small quantization losses

Code book (2 of 2)

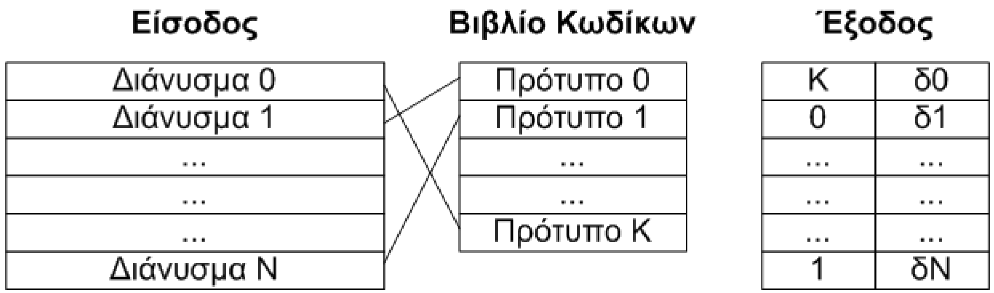
- Adaptive code books
  - We start with a basic set of code words
  - We add new code words as we proceed
  - Allows adaptation to source contents

Lossy VQ



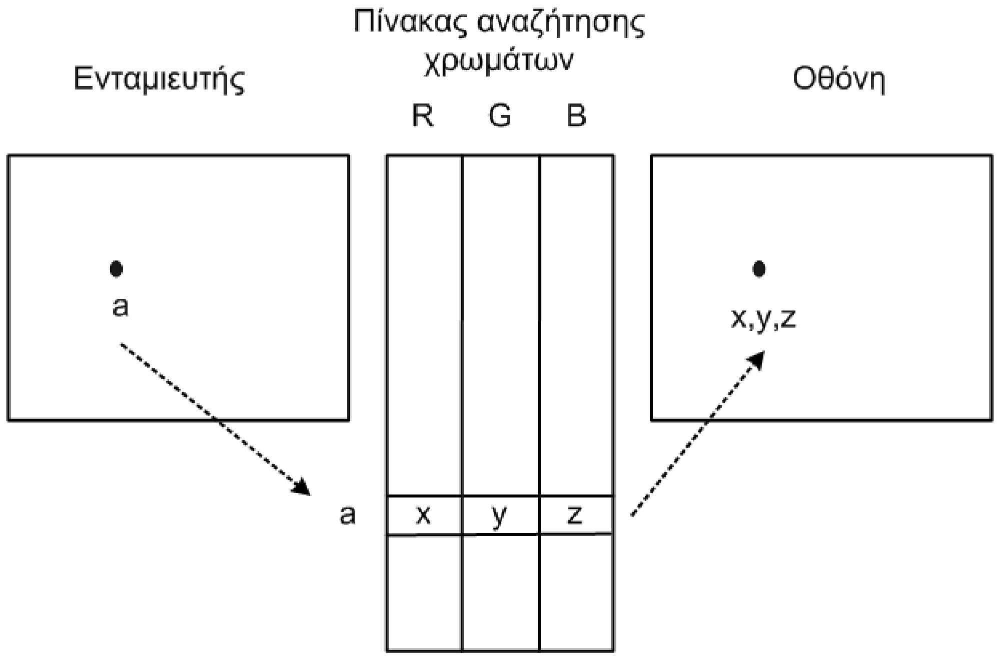
- Lossy variant
  - Code book with n byte code words
  - Input is broken into n byte vectors
  - Quantization to closest matching code word
    - The code book cannot contain all vectors
    - This leads to approximations (lossy)

Less lossy VQ



- Lossless variants
  - Same as lossy until the matching
  - Also sends difference from match
    - Requires more bits
  - Can reconstruct exact input
  - More often, just a less lossy version of it!

Example: CLUT



- Color Lookup Table (CLUT)
  - Each pixel encoded with an index
  - The index selects a color from a table
  - The table is the code book

How to choose colors? (1 of 3)

- Color space quantization
  - Example: from 24 bit to 8 bit color
    - Input: three 8 bit values (RGB)
    - Output: one of 256 colors (24 bits each)
    - Assume a resolution of 1920 x 1080
    - We start with 1920 x 1080 x 24 bit
    - We end with 1920 x 1080 x 8 bit
    - Plus a CLUT with 256 x 24 bit

How to choose colors? (2 of 3)

- Constructing a fixed CLUT
  - We can visualize colors as a 3D cube
  - Break the cube into equal rectangular blocks
    - Divide R and G by 32 (2^5) and B by 64 (2^6)
    - We basically create 8 x 8 x 4 blocks
    - Each block has a 3+3+2=8 bit index
    - Each rectangle is represented by its central color
    - Image content is disregarded
    - Color gradients are abrupt

How to choose colors? (3 of 3)

- Constructing an image-dependent CLUT
  - Sort pixels by R value and find their median *m*
    - Pixels with R < m are assigned 0, R >= m are assigned 1
  - In each R block, do the same (split) for G values
  - In each RG block, do the same for B values
  - Repeat for R, G, B, R and G
  - Every pixel matched by an 8 bit index
  - Represented by the central color of its block
    - But the blocks are NOT equal!

End of Section #5

Class: Multimedia Technology, **Section # 5:** Principles of Coding

Instructor: George Xylomenos, **Department:** Informatics