Module-5
Audio and Video Coding

Dr. Markkandan S

School of Electronics Engineering (SENSE)
Vellore Institute of Technology
Chennai

Human Speech Production Mechanism

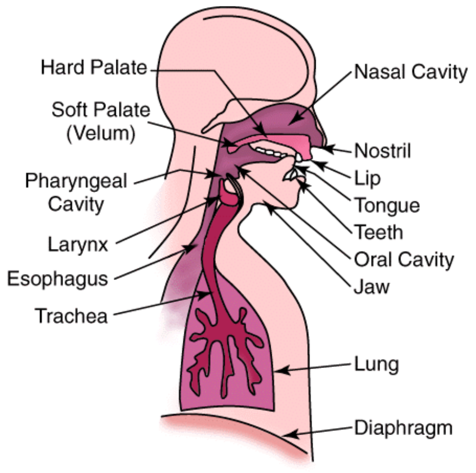

Figure 1: Human Speech Production Mechanism

Introduction to Audio Compression

- Audio compression refers to the process of reducing the amount of data required to represent audio signals while maintaining acceptable quality.
- Two main types of compression:
  - Lossy compression: Reduces data by removing some audio information, which may result in a loss of quality (e.g., MP3).
  - Lossless compression: Reduces data without any loss of quality (e.g., FLAC).
- Compression reduces file size, transmission time, and storage requirements.
- Applications: Digital media, streaming, telecommunication, and storage.

Introduction to Audio Coding

- Audio coding: compression of audio signals for efficient storage/transmission
- Objectives:
  - Reduce bit rate while maintaining perceptual quality
  - Enable efficient storage and transmission of audio
- Key approaches:
  - Waveform coding: Directly encode audio samples
  - Parametric coding: Encode model parameters (e.g., LPC)
  - Perceptual coding: Exploit human auditory system limitations
- Trade-offs:
  - Compression ratio vs. audio quality
  - Computational complexity vs. performance
  - Delay vs. coding efficiency
- Applications: Digital audio broadcasting, VoIP, music streaming, etc.

Types of Audio Coding Techniques

- Common audio coding techniques:
  - Pulse Code Modulation (PCM)
  - Differential Pulse Code Modulation (DPCM)
  - Adaptive Differential PCM (ADPCM)
  - Linear Predictive Coding (LPC)
  - Code-Excited Linear Prediction (CELP)
  - Perceptual Audio Coding (e.g., MPEG Audio)
- PCM: Direct sampling and quantization of the audio signal.
- DPCM and ADPCM: Reduces redundancy in the signal by encoding differences between samples.
- LPC and CELP: Use models of human speech production to achieve efficient compression.

Linear Predictive Coding(LPC)

Overview of Linear Predictive Coding (LPC)

- LPC: Efficient parametric coding technique for speech
- Core idea: Model speech production process
- Key components:
  - Excitation source model
  - Vocal tract filter model
- Process:
  1. Analyze speech to extract model parameters
  2. Transmit parameters (not raw audio)
  3. Synthesize speech at receiver using parameters
- Advantages:
  - Very low bit rate (2.4 kbps - 4.8 kbps)
  - Good intelligibility for speech
- Limitations:
  - "Robotic" sound quality
  - Not suitable for non-speech audio

Linear Predictive Coding (LPC): Overview

- LPC is widely used for speech signal compression. The basic idea is to model the vocal tract as a linear filter and represent speech as the output of this filter.
- Equation for LPC model:

$$y_n = \sum_{i=1}^{p} a_i y_{n-i} + Ge_n \tag{1}$$

  where:
  - y_n is the current sample,
  - a_i are the LPC coefficients,
  - e_n is the excitation signal,
  - G is the gain factor.
- Applications: Speech coding, synthesis, and recognition.

LPC: Speech Production Model

- Models speech as output of a time-varying linear system
- Two main components:
  - Excitation source: Models airflow from lungs
  - Vocal tract filter: Models acoustic properties of vocal tract
- Excitation types:
  - Voiced: Quasi-periodic pulses (e.g., vowels)
  - Unvoiced: White noise (e.g., fricatives)
- Vocal tract modeled as an all-pole filter:

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{2}$$

where G is gain, p is filter order, a_k are filter coefficients

- Time-domain representation:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{3}$$

LPC: Encoder and Decoder

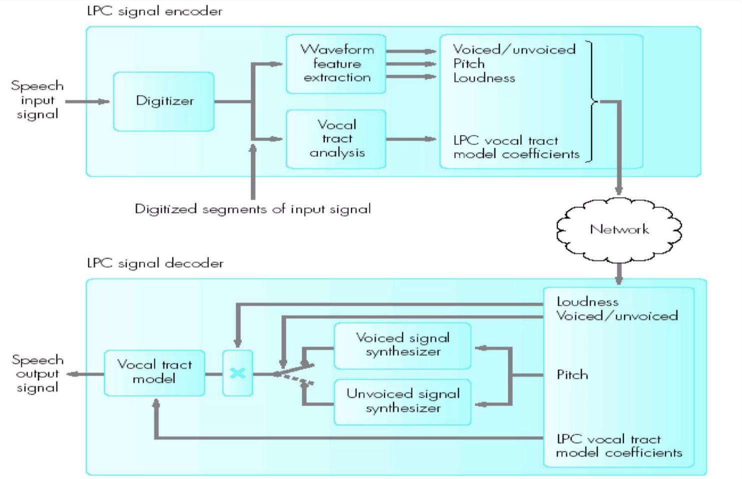

Figure 2: LPC encoder and Decoder

LPC: Speech Signal Modeling

- The speech signal is modeled by a filter that represents the vocal tract.
- The excitation signal e_n drives this filter.
- LPC analyzes the speech into frames and estimates filter coefficients for each frame.

Figure 3: Speech synthesis model

LPC: Vocal Tract Filter

- All-pole filter approximates vocal tract resonances (formants)
- Transfer function:

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{4}$$

- Typical filter order: 10-12 for 8 kHz sampled speech
- Estimation of filter coefficients:
  - Minimize mean squared prediction error
  - Autocorrelation method:

$$R_a = r \tag{5}$$

    where R is the autocorrelation matrix, a is the coefficient vector, r is the autocorrelation vector
  - Solved efficiently using Levinson-Durbin recursion
- Stability ensured by converting to reflection coefficients
- Quantization: Non-uniform quantization of reflection coefficients

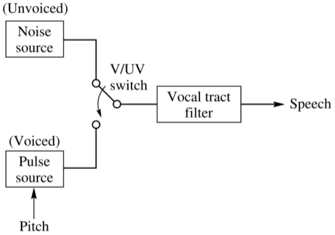

FIGURE 18.5

A model for speech synthesis.

Figure 4: Model for speech synthesis with vocal tract filter

LPC: Excitation Source

- Two types of excitation:
  1. Voiced excitation:
     - Quasi-periodic pulse train
     - Parameters: Pitch period, voiced/unvoiced decision
     - Pitch detection algorithms:
     - Time-domain: Autocorrelation, AMDF
     - Frequency-domain: Harmonic peak detection
  2. Unvoiced excitation:
     - White noise generator
     - No additional parameters needed
- Voiced/Unvoiced decision:
  - Based on features like:
    - Short-term energy
    - Zero-crossing rate
    - First reflection coefficient
- Excitation gain:

LPC: Excitation Source

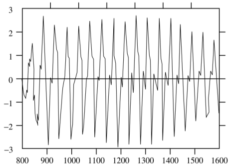

FIGURE 18.2

The sound /e/ in *test*.

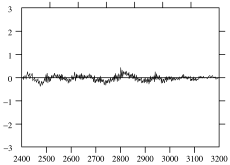

FIGURE 18.3

The sound /s/ in *test*.

LPC: Voicing and Pitch Detection

- Voicing decision: Determines if the segment is voiced or unvoiced.
- Voiced speech has a periodic structure, unvoiced speech resembles noise.
- Pitch period estimation: A critical step for voiced speech.
- The pitch period is extracted using autocorrelation or average magnitude difference function (AMDF):

$$AMDF(P) = \frac{1}{N} \sum_{i=1}^{N} |y_i - y_{i-P}| \tag{6}$$

- Voicing and pitch information helps generate the excitation signal for LPC.

LPC: Voicing and Pitch Detection - AMDF

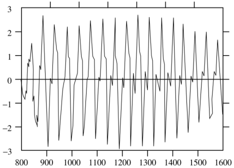

FIGURE 18.2
The sound /e/ in *test*.

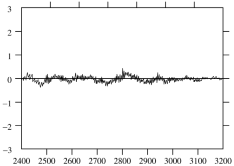

FIGURE 18.3
The sound /s/ in *test*.

Pitch Detection

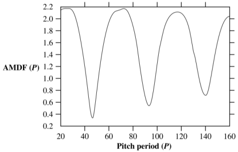

FIGURE 18.6
AMDF function for the sound /e/ in *test*.

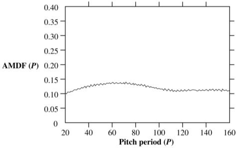

FIGURE 18.7
AMDF function for the sound /s/ in *test*.

AMDF for 'e' and 's'

LPC: Parameter Estimation

- Goal: Estimate LPC model parameters from speech signal
- Process:
  1. Pre-emphasis: High-pass filter to flatten spectral slope
  2. Framing: Divide speech into 20-30 ms frames
  3. Windowing: Apply window function (e.g., Hamming) to each frame
  4. Autocorrelation: Compute autocorrelation coefficients
  5. Levinson-Durbin recursion: Solve for LPC coefficients

LPC: Parameter Estimation

- Levinson-Durbin algorithm:

  1. Initialize: $E_0 = R(0)$
  2. For $i = 1$ to p:

$$k_i = \frac{R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)}{E_{i-1}} \tag{7}$$

$$a_i^{(i)} = k_i \tag{8}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}, \quad 1 \leq j < i \tag{9}$$

$$E_i = (1 - k_i^2) E_{i-1} \tag{10}$$

- Output: LPC coefficients a_i and reflection coefficients k_i

LPC: Transmission of Parameters

- Parameters to transmit:
  - Reflection coefficients (instead of direct LPC coefficients)
  - Pitch period (for voiced frames)
  - Voiced/unvoiced decision
  - Gain
- Quantization:
  - Reflection coefficients: Non-uniform quantization

$$g_i = \frac{1 + k_i}{1 - k_i} \tag{11}$$

  - Pitch: Logarithmic quantization
  - Gain: Logarithmic quantization
- Bit allocation example (LPC-10, 2.4 kbps):

LPC: Transmission of Parameters

  - Reflection coefficients: 41 bits
  - Pitch and V/UV: 7 bits
  - Gain: 5 bits
  - Synchronization: 1 bit
- Frame duration: 22.5 ms (180 samples at 8 kHz)
- Resulting bit rate: 54 bits / 22.5 ms 2400 bps

LPC: Speech Synthesis at Receiver

- Process:
  1. Decode received parameters
  2. Generate excitation signal
  3. Synthesize speech using all-pole filter
- Excitation generation:
  - Voiced: Impulse train with decoded pitch period
  - Unvoiced: White noise generator
- All-pole filter implementation:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{12}$$

- Overlap-add successive frames to reduce discontinuities

LPC: Speech Synthesis at Receiver

- Post-processing:
  - De-emphasis filter (inverse of pre-emphasis)
  - Adaptive postfiltering to enhance formants

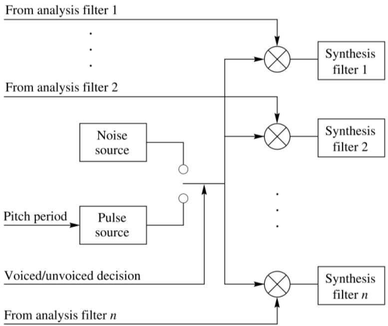

FIGURE 18.4

Limitations of LPC and Need for CELP

- Limitations of basic LPC:
  - "Buzzy" or "robotic" sound quality
  - Poor representation of unvoiced and transient sounds
  - Binary voiced/unvoiced decision too simplistic
  - Limited pitch resolution
  - Sensitive to transmission errors
- Reasons for limitations:
  - Oversimplified excitation model
  - All-pole filter may not capture all spectral details
  - Frame-based analysis loses some temporal resolution

Limitations of LPC and Need for CELP

- Need for improvement:
  - Better excitation model
  - Finer pitch and spectral representation
  - Improved perceptual quality at low bit rates
- CELP as a solution:
  - Addresses LPC limitations
  - Uses codebook of excitation vectors
  - Incorporates perceptual weighting
  - Achieves better quality at similar bit rates

Code Excited Linear Prediction (CELP)

Introduction to Code Excited Linear Prediction (CELP)

- Key idea: Use codebook of excitation vectors
- Components:
  - LPC filter (as in traditional LPC)
  - Adaptive codebook (for pitch structure)
  - Fixed (stochastic) codebook (for residual excitation)
  - Perceptual weighting filter
- Process:
  1. LPC analysis to obtain filter coefficients
  2. Search codebooks for best excitation
  3. Minimize perceptually weighted error
  4. Transmit codebook indices and gains

Introduction to Code Excited Linear Prediction (CELP)

- Advantages:
  - Improved speech quality compared to LPC
  - Efficient at low bit rates (4.8-16 kbps)
  - Handles both voiced and unvoiced speech well
- Challenges:
  - Computationally intensive codebook search
  - Requires larger codebooks for higher quality

Introduction to Code Excited Linear Prediction (CELP)

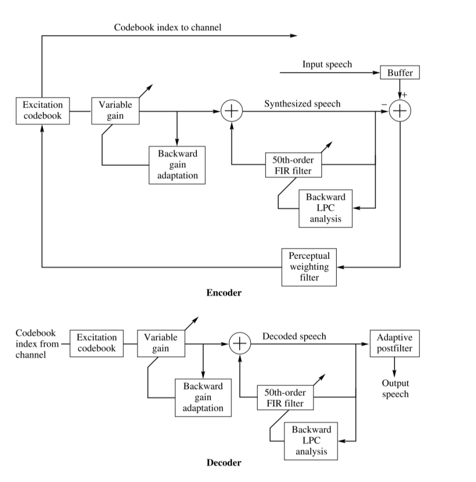

Figure 6: Block diagram of the ITU-T H.261 CELP Encoder and Decoder

CELP: Codebook Structure(1/2)

- Two main codebooks in CELP:
    1. Adaptive codebook
    2. Fixed (stochastic) codebook
- Adaptive codebook:
    - Models pitch periodicity and long-term correlations
    - Constructed from past excitation signals
    - Updated each subframe
    - Typically 128-256 entries

CELP: Codebook Structure(2/2)

- Fixed codebook:
  - Models residual excitation after pitch prediction
  - Designed to cover range of possible excitations
  - Typically 512-1024 entries
  - Various structures: Binary, ternary, sparse algebraic
- Excitation signal:

$$e(n) = \beta v(n) + \gamma c(n) \tag{13}$$

where $v(n)$ is from adaptive codebook, $c(n)$ is from fixed codebook, β and γ are respective gains

CELP: Stochastic Codebook

- Purpose: Model residual excitation not captured by adaptive codebook
- Types of stochastic codebooks:
  - Random codebook: Gaussian random sequences
  - Algebraic codebook: Structured sparse vectors
- Random codebook:
  - Entries are Gaussian random sequences
  - Typically quantized to +1, -1, or 0
  - Large storage requirement
- Algebraic codebook (ACELP):
  - Sparse vectors with few non-zero pulses
  - Pulse positions and signs determined by algebraic structure

CELP: Adaptive Codebook

- Purpose: Model pitch periodicity and long-term correlations
- Structure:
  - Contains past excitation signals
  - Updated each subframe
  - Typically 128-256 entries
- Adaptive codebook vector:

$$v(n) = e(n - T + i), \quad i = 0, 1, ..., N - 1 \tag{16}$$

where T is pitch lag, N is subframe size

- Fractional pitch:
  - Allows finer pitch resolution (e.g., 1/3 or 1/4 sample)
  - Requires interpolation of past excitation

CELP: Perceptual Weighting

- Purpose: Shape quantization noise to be less perceptible
- Concept: Exploit masking properties of human auditory system
- Perceptual weighting filter:

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} \tag{18}$$

  where $A(z)$ is LPC filter, typically $\gamma_1 = 0.9$, $\gamma_2 = 0.5$
- Effect:
  - Attenuates error in formant regions
  - Amplifies error in spectral valleys

CELP: Perceptual Weighting

- Implementation:
  - Apply $W(z)$ to error signal in codebook search
  - Minimize weighted error:

$$E_w = \sum_{n=0}^{N-1} [x_w(n) - \hat{x}_w(n)]^2 \tag{19}$$

- Benefits:
  - Improved subjective quality
  - Better allocation of bits to perceptually important regions

CELP: Encoder Operation

- CELP encoding steps:
  1. LPC analysis: Compute coefficients, convert to LSP
  2. Subframe processing (typically 4 per frame)
  3. Adaptive codebook search: Find best pitch lag and gain
  4. Fixed codebook search: Find best index and gain
  5. Parameter quantization: LSPs, pitch, indices, gains
  6. Update memories for next frame
- Computational complexity:
  - Codebook search most intensive
  - Fast search algorithms used in practice
- Bit allocation example (FS1016 4.8 kbps):
  - LSP: 34 bits/frame
  - Pitch: 8 bits/subframe
  - Fixed codebook: 9 bits/subframe
  - Gains: 5 bits/subframe

CELP: Encoder Operation

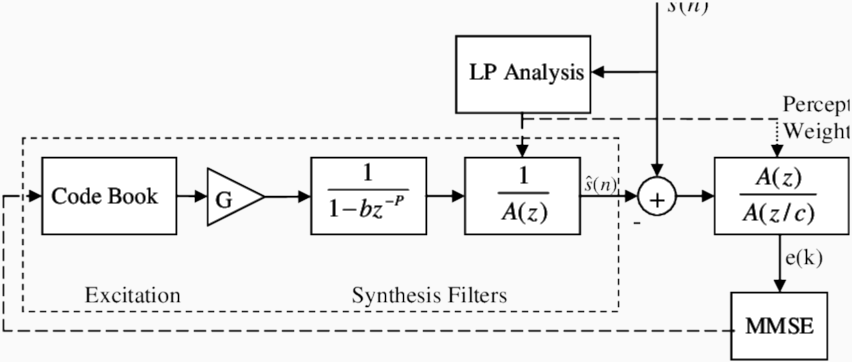

Figure 7: Block diagram of CELP encoder

Image source: `https://www.researchgate.net/figure/Block-diagram-of-CELP-encoder_fig1_264423574`

CELP: Decoder Operation

- Steps in CELP decoding:
  1. Parameter decoding
  2. LSP to LPC conversion
  3. For each subframe:
     - Adaptive codebook contribution
     - Fixed codebook contribution
     - Excitation reconstruction
     - LPC synthesis filtering
  4. Post-processing

- Excitation reconstruction:

$$e(n) = \beta v(n) + \gamma c(n) \quad (20)$$

- LPC synthesis:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n) \quad (21)$$

CELP: Decoder Operation

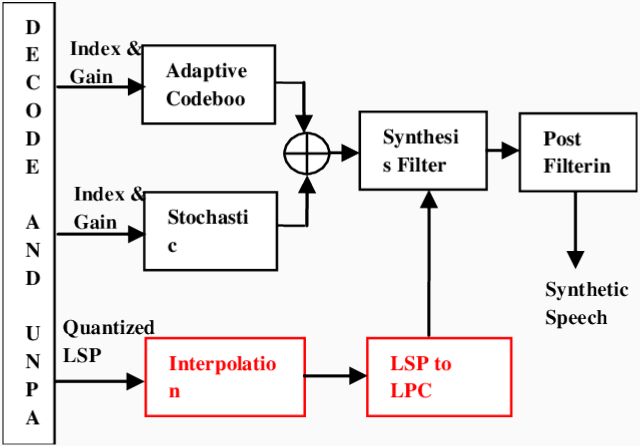

Figure 8: Block diagram of CELP decoder

CELP: Examples (FS1016, G.728)

Federal Standard 1016 (4.8 kbps)

- Frame size: 30 ms
- Subframes: 4 x 7.5 ms
- LPC order: 10
- Adaptive codebook: 128 entries
- Fixed codebook: 512 entries
- Bit allocation:
    - LSP: 34 bits/frame
    - Pitch: 8 bits/subframe
    - Fixed codebook: 9 bits/subframe
    - Gains: 5 bits/subframe

ITU-T G.728 (16 kbps)

- Frame size: 0.625 ms (5 samples)
- LPC order: 50
- Backward adaptive prediction
- No explicit pitch prediction
- Shape-gain vector quantization
- Bit allocation:
    - Shape codebook: 7 bits/frame
    - Gain codebook: 3 bits/frame
- Low delay: 0.625 ms

Perceptual Coding & MPEG

Introduction to Perceptual Coding

- Goal: Exploit limitations of human auditory system
- Key principles:
  - Auditory masking
  - Critical band analysis
  - Temporal masking
- General approach:
  1. Time-frequency analysis
  2. Psychoacoustic modeling
  3. Bit allocation based on perceptual importance
  4. Quantization and coding
- Advantages:

Introduction to Perceptual Coding

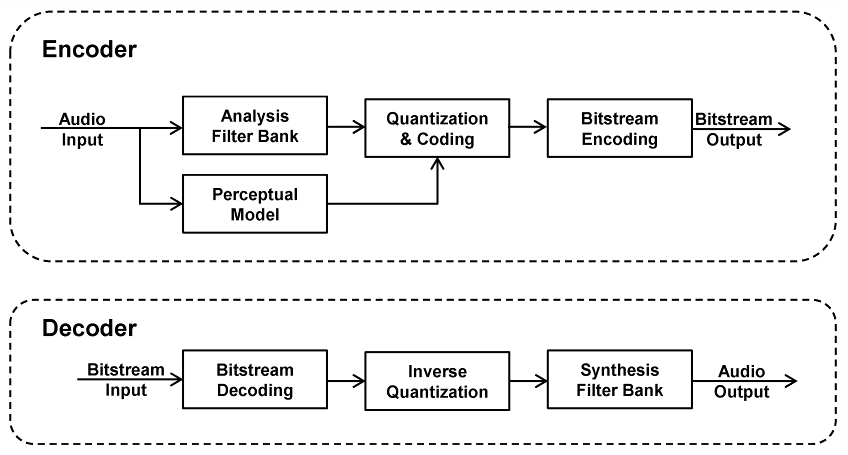

Figure 9: General structure of a perceptual audio coder

Image source: https://www.researchgate.net/figure/Basic-structure-of-perceptual-audio-coder_fig1_224345681

Psychoacoustic Principles in Audio Coding

- Auditory masking:
  - Louder sounds mask quieter sounds
  - Masking threshold varies with frequency
- Critical bands:
  - Non-uniform frequency resolution of ear
  - Approximated by bark scale
- Temporal masking:
  - Pre-masking: 20 ms before masker
  - Post-masking: up to 200 ms after masker
- Just Noticeable Distortion (JND):
  - Minimum perceivable change in sound
  - Varies with frequency and intensity

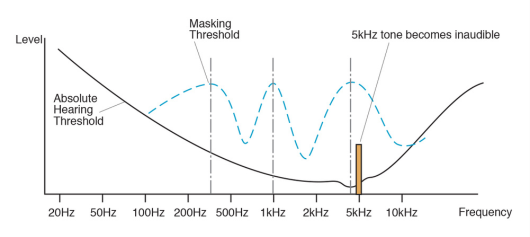

Figure 10: Frequency masking

Image source: `https://www.researchgate.net/figure/Illustration-of-the-masking-effect_fig2_220009783`

MPEG Audio Coding: Overview

- MPEG: Moving Picture Experts Group
- Audio coding standards:
  - MPEG-1 Audio (1992)
  - MPEG-2 Audio (1994)
  - MPEG-4 Audio (1999)
- MPEG-1 Audio Layers:
  - Layer I: Simplest, lowest compression
  - Layer II: Improved compression
  - Layer III (MP3): Highest compression

MPEG Audio Coding: Overview

- Key features:
  - Perceptual coding principles
  - Filterbank for time-frequency mapping
  - Psychoacoustic model
  - Dynamic bit allocation coding

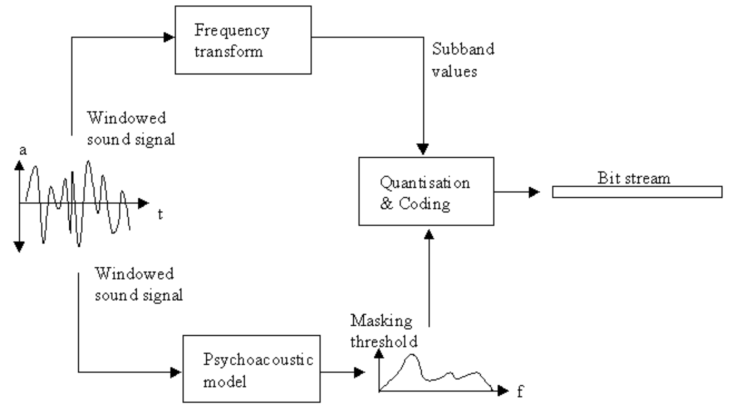

Figure 11: General structure of MPEG audio encoder

Image source: `https://www.researchgate.net/figure/Block-diagram-of-a-generic-MPEG-1-Layer-I-II-encoder_fig1_224245681`

LPC: Numerical Problem 1

Problem

Given an LPC model of order 4 with the following reflection coefficients: $k_1 = 0.5$, $k_2 = -0.3$, $k_3 = 0.2$, $k_4 = 0.1$ Calculate the g-parameters for transmission using the equation: [$g_i = \frac{1+k_i}{1-k_i}$]

Solution

Calculating for each coefficient:

$$g_1 = \frac{1+0.5}{1-0.5} = \frac{1.5}{0.5} = 3$$

$$g_2 = \frac{1+(-0.3)}{1-(-0.3)} = \frac{0.7}{1.3} \approx 0.538$$

$$g_3 = \frac{1+0.2}{1-0.2} = \frac{1.2}{0.8} = 1.5$$

$$g_4 = \frac{1+0.1}{1-0.1} = \frac{1.1}{0.9} \approx 1.222$$

Therefore, the g-parameters are approximately: 3, 0.538, 1.5, and 1.222.

CELP: Numerical Problem 2

Problem

In a CELP coder, the excitation signal is given by: [e(n) = $\beta v(n) + \gamma c(n)] If \beta = 0.8$, $\gamma = 0.5$, $v(n) = 1, -1, 0.5, -0.5$, and $c(n) = 0.2, 0.3, -0.1, 0.1$ for a subframe of 4 samples, calculate the excitation signal $e(n)$.

Solution

Calculating sample by sample:

$$e(0) = 0.8(1) + 0.5(0.2) = 0.8 + 0.1 = 0.9$$
$$e(1) = 0.8(-1) + 0.5(0.3) = -0.8 + 0.15 = -0.65$$
$$e(2) = 0.8(0.5) + 0.5(-0.1) = 0.4 - 0.05 = 0.35$$
$$e(3) = 0.8(-0.5) + 0.5(0.1) = -0.4 + 0.05 = -0.35$$

Therefore, $e(n) = 0.9, -0.65, 0.35, -0.35$

LPC: Numerical Problem 3

Using the Levinson-Durbin algorithm, calculate k_2 and $a_1^{(2)}$ given: $R(0) = 1$, $R(1) = 0.5$, $R(2) = 0.2$ Recall the relevant equations:

$$k_i = \frac{R(i) - \sum_{j=1}^{i-1} a^{(i-1)} jR(i-j)}{Ei - 1}$$

$$a^{(i)} j = a^{(i-1)} j - k_i a^{(i-1)} i - j, \quad 1 \leq j < i$$

$$E_i = (1 - k_i^2) Ei - 1$$

Solution

First, calculate k_1: [$\mathrm{k}_1 = \frac{R(1)}{R(0)} = \frac{0.5}{1} = 0.5] Then,$ E_1:[$\mathrm{E}_1 = (1 - k_1^2) E_0 = (1 - 0.5^2) 1 = 0.75$]
Now, calculate k_2:

$$k_2 = \frac{R(2) - a_1^{(1)} R(1)}{E_1} \qquad = \frac{0.2 - 0.5(0.5)}{0.75} = \frac{0.2 - 0.25}{0.75} = -\frac{1}{15} \approx -0.0667$$

Finally, calculate $a_1^{(2)}$:

$$a_1^{(2)} = a_1^{(1)} - k_2 a_1^{(1)} \qquad = 0.5 - (-0.0667)(1) = 0.5 + 0.0667 = 0.5667$$

Therefore, $k_2 \approx -0.0667$ and $a_1^{(2)} \approx 0.5667$

Video Coding

Introduction to Video Compression

- Video: time sequence of images
- Huge data rates: e.g., CCIR 601 format
  - 30 frames/second, 16 bits/pixel
  - 168 Mbits/second
- Goal: Reduce data rate while maintaining quality
- Key approach: Exploit temporal correlation between frames
- Challenges:
  - Motion video perception differs from still images
  - Artifacts may be more/less noticeable in motion

Video Compression: Basic Concept

- Use previous frame to predict current frame
- Encode and transmit prediction error (residual)
- Receiver reconstructs frame using:
  - Prediction from previous frame
  - Received prediction error
- Key technique: Motion-compensated prediction

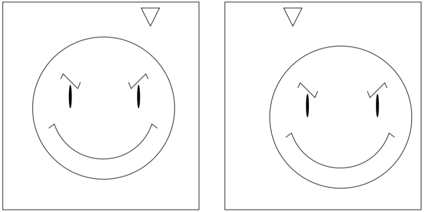

FIGURE 19.1

Two frames of a video sequence.

Two frames of a video sequence

Video Compression: Basic Concept

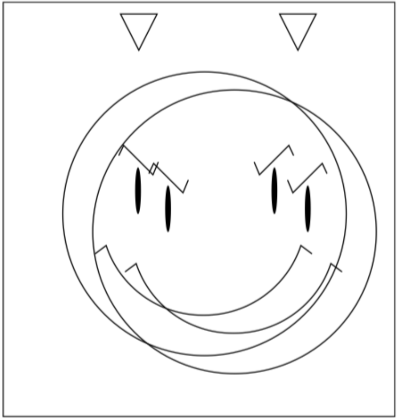

FIGURE 19.2

Difference between the two frames.

Video Coding: Motion Estimation and Compensation

- Problem: Objects move between frames
- Solution: Block-based motion compensation
  - Divide frame into blocks (e.g., 16x16 pixels)
  - Search previous frame for best matching block
  - Transmit motion vectors instead of pixel differences
- Advantages:
  - Exploits temporal redundancy
  - Significantly reduces data to be transmitted

Motion Estimation: Block Matching

- Search area typically ± 15 pixels
- Matching criterion: Sum of Absolute Differences (SAD) $SAD = \sum_{i=0}^{15} \sum_{j=0}^{15} |C_{ij} - R_{ij}|$ where C_{ij} and R_{ij} are pixels in current and reference blocks
- Trade-off: Block size vs. prediction accuracy

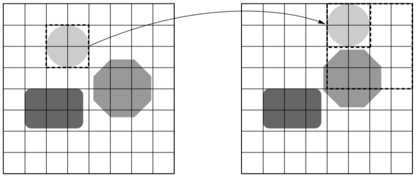

FIGURE 19.11

Effect of block size on motion compensation.

Effect of block size on motion compensation

Types of Frames in Video Coding

- I-frames (Intra-coded)
  - Encoded without reference to other frames
  - Provide random access points
- P-frames (Predictive-coded)
  - Use motion-compensated prediction from previous I or P frame
- B-frames (Bidirectionally predictive-coded)
  - Use prediction from both past and future frames
  - Highest compression, but introduce delay

Group of Pictures (GOP) Structure

- GOP: Sequence of I, P, and B frames
- Typical pattern: IBBPBBPBBPBB
- I-frame: Start of each GOP
- P-frames: Predicted from previous I or P
- B-frames: Bidirectional prediction
- Benefits:
  - Efficient compression
  - Flexible access to video

Group of Pictures (GOP) Structure

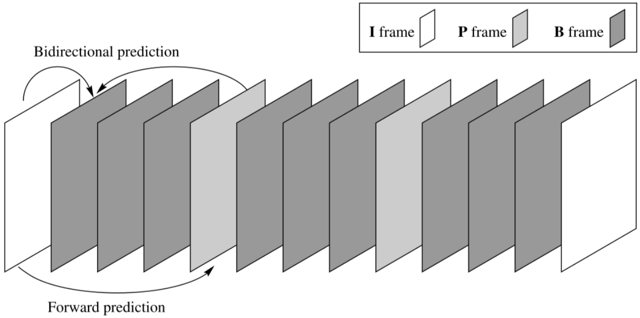

FIGURE 19.14

A possible arrangement for a group of pictures.

Figure 12: A possible arrangement for a group of pictures

Video Encoding Process

1. Motion estimation and compensation
2. Transform coding (usually DCT)
3. Quantization
4. Entropy coding

Video Encoding Process

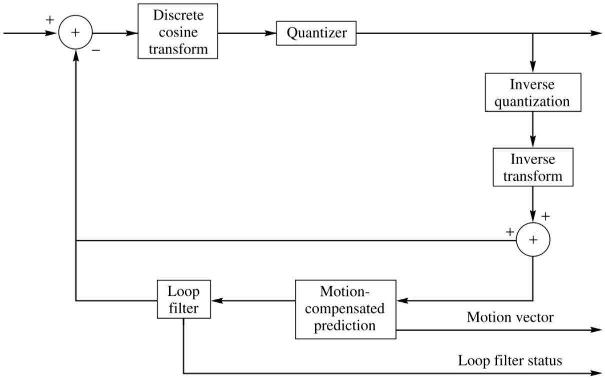

FIGURE 19.10

Block diagram of the ITU-T H.261 encoder.

Figure 13: Block diagram of a video encoder

Video Decoding Process

1. Entropy decoding
2. Inverse quantization
3. Inverse transform
4. Motion compensation

- Decoder performs inverse operations of encoder
- Reconstructs video frames from received data

Rate Control in Video Coding

- Purpose: Maintain constant output bit rate
- Methods:
  - Adjust quantization step size
  - Drop frames if necessary
- Buffer fullness guides rate control decisions
- Balances quality and bit rate

Video Coding Standard: MPEG-4

- Developed by Moving Picture Experts Group (MPEG)
- Object-oriented approach to multimedia coding
- Key features:
    - Object-based coding
    - Sprite coding for backgrounds
    - Scalability (temporal, spatial, and object)
- Applications:
    - Digital television
    - Interactive graphics applications
    - Streaming media

MPEG-4: Object-Based Coding

- Video scene as collection of objects
- Each object coded independently
- Objects can be:
    - Visual (e.g., background, talking head)
    - Aural (e.g., speech, music)
- Scene description using BIFS (Binary Format for Scenes)
- Allows flexible manipulation of objects

MPEG-4: Object-Based Coding

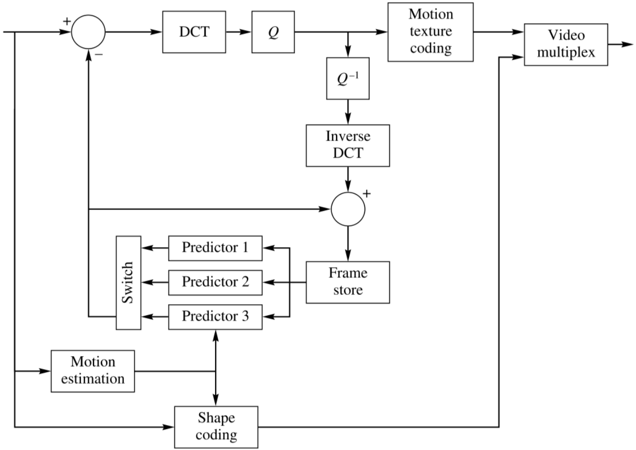

FIGURE 19.18

A block diagram for video coding.

MPEG-4: Sprite Coding

- Sprite: Large panoramic background image
- Transmitted once, reused in multiple frames
- Moving foreground objects placed on sprite
- Efficient for scenes with static backgrounds
- Equation for sprite warping:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where (x, y) is the original coordinate and $(x'/w', y'/w')$ is the warped coordinate

MPEG-4: Scalability

- Temporal Scalability:
    - Enhance frame rate
    - Base layer + enhancement layer(s)
- Spatial Scalability:
    - Enhance spatial resolution
    - Use upsampling of base layer
- SNR Scalability:
    - Enhance quality (Signal-to-Noise Ratio)
    - Refine quantization in enhancement layer
- Object Scalability:
    - Selectively transmit or decode objects

MPEG-4: Shape Coding

- Important for object-based coding
- Methods:
  - Bitmap-based
  - Contour-based
- Context-based Arithmetic Encoding (CAE) for binary alpha planes
- Equation for context number:

$$C_k = \sum_{i=0}^{9} c_i \cdot 2^i$$

where c_i are binary values of neighboring pixels

MPEG-4: Facial Animation

- Facial Definition Parameters (FDPs):
  - Define shape and texture of face
- Facial Animation Parameters (FAPs):
  - Control facial expressions
- 68 FAPs defined, e.g.:
  - Jaw rotation
  - Eye movement
  - Lip deformation
- Equation for FAP interpolation:

$$FAP(t) = FAP(t_1) + \frac{t - t_1}{t_2 - t_1}[FAP(t_2) - FAP(t_1)]$$

where t_1 and t_2 are key frames