BASIC COMPRESSION TECHNIQUES

N. C. State University

CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lectures # 05

Questions / Problems / Announcements?

- Matlab demo of
  - DFT
  - Low-pass windowed-sinc filter design
  - Filtering
- Matlab files are online – try them!
- Compressing the "castle"
- ?

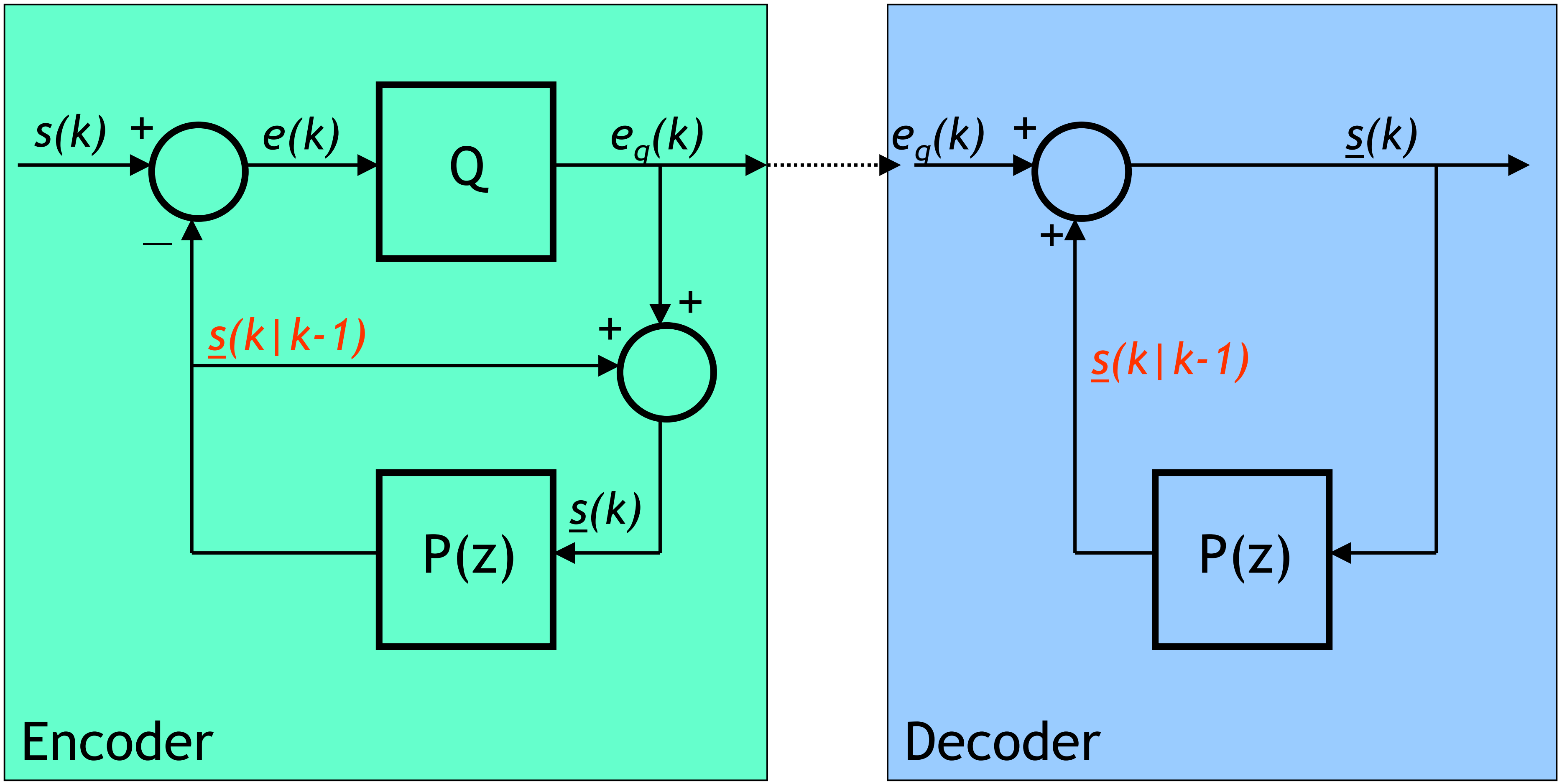
③ Prediction-Based Encoding

- Suppose specific sequences of symbols are predictable, according to some algorithm
- After *i-1* symbols have been encoded, predict the *i*th *symbol*
- Principles involved
  - Redundancy removal → subtract a predicted value from the input value
    - *predictor*
    - In other words, successive input samples are correlated
  - Entropy reduction → quantize difference between input sample and predicted value
    - *quantizer*

DPCM (cont.)

- Terminology
  - Input signal at time k = $s(k)$
  - Predicted value at time k = $\underline{s}(k|k\text{-}1)$
  - Prediction error at time k = $e(k) = s(k) - \underline{s}(k|k\text{-}1)$
  - Quantized prediction error at time k = $e_q(k)$
  - Digitized approximation to input at time k = $\underline{s}(k)$
- Quantization
  - Linear PCM, for example
  - Step size may be adjusted, or *adapted*, dynamically
    - (details omitted; see Gibson if interested)

Differential PCM (DPCM)



DPCM Predictor Adaptation

- Prediction accomplished via convolution (filtering)
  - $\underline{s}(k|k-1) = a_1\underline{s}(k-1)+a_2\underline{s}(k-2)+\ldots a_N\underline{s}(k-N)$
- Adapting the prediction: constantly strive to reduce the prediction error
- Example of "backwards adaptation"
  - $a_i(k+1) = \alpha a_i(k)+G(k+1)e_q(k+1)$
  - α is constant "weighting factor" near 1
  - $G(k+1)$ is vector gain computed to minimize the prediction error
    - (details omitted; see Gibson if interested)
- There is no need to transmit the a_i's, since the decoder can run exactly the same prediction adaptation algorithm

Example "Media-Specific" Predictors

- Images: the *j*th Pixel in a row will have the same color as the (*j-1*)th pixel
- Video: the *j*th frame will have the same content as the (*j-1*)th frame

④ LZW Compression

- Given: an alphabet of input "symbols"
  - Pixel colors, ascii characters, …
- Goal: Construct a "dictionary" of input symbol sequences (ISS's) and output (replacement) symbols
  - desired: output symbols should be shorter than the ISS's they replace
- Initially, the dictionary contains only the individual input symbols
  - "sequences of length 1"
- Scanning input from start to finish
  - add each newly-encountered ISS to the dictionary
  - replace each ISS found in the dictionary with the matching output symbol
  - then *skip over* the matched ISS, and continue

Example (LZW Encoding)

- Assume 12-bit output symbols, 8-bit (ASCII) input symbols
- First 256 entries in dictionary = the alphabet of input symbols
  - 'A'=65,'B'=66,'C'=67,...
- Example of input data to compress: **BABAABAAA**

  (=9*8=72 bits)

Encoder Input	**Encoder Output**	**CodeWord Added**	**String**
B	66	256	BA
A	65	257	AB
BA	256	258	BAA
AB	257	259	ABA
A	65	260	AA
AA	260		

LZW Decoding

- Given the encoded (output) data, and the initial dictionary values
  - Do not need the entire dictionary!
- For each output symbol encountered
  - Replace with the input sequence from the dictionary
  - And, create new entry in the dictionary
- If an output symbol is not found in dictionary, it can still be recovered
  - It must be the same as the immediately-previous sequence, with the last symbol repeated

Example (LZW Decoding)

- Encoded string =
  - 66 65 256 257 65 260

Decoder Input	Decoder Output	CodeWord Added	String
66	B		
65	A	256	BA
256	BA	257	AB
257	AB	258	BAA
65	A	259	ABA
260	AA	260	AA

LZW Dictionary Problems

- Dictionary fills up during construction?
- Options
  - Expand table size dynamically
  - Purge table and start over if compression ratio too low

Lossy Compression Methods

❺ (We've already seen DPCM, ADPCM)

❻ (We'll do transform coding later when we talk about audio and image compression)

❼ Source coding

Source Coders (Synthesis)

- Complex data may be produced by simple mechanism. Examples:
  - speech from the vocal tract
  - bitmap from a drawing tool
  - musical sound from an instrument
  - image that is a flowchart or block diagram

Source Coders (cont.)

- Compression idea: record the state of the mechanism, instead of the output it produces
- Decompression: *synthesize* the data by modeling the mechanism

Example (Source Coding)

- Compressing a performance on the piano
- Record the keys pressed, when, how long, and how forcefully
- Decompression: synthesize a realistic-sounding piano note
  - with the right pitch
  - at the right time
  - with the right volume
- MIDI recording!

Source Coding Tradeoffs

- Huge amount of compression
- Quality loss because synthesis is not perfect

Sources of Information

- [Crane97] *A Simplified Approach to Image Processing*
  - Chapter 9, section on LZW
- [Gibson98] *Digital Compression for Multimedia*
  - Section 5.3, highlights only