Digital Signal Processing for Music

Part 24: Redundancy Coding

alexander lerch

Georgia Tech Center for Music Technology

redundancy coding
introduction

- **goals**
  - minimize bitrate
  - ensure that content can be reconstructed without error/loss

- **objective**
  - aim for a symbol pdf is as non-uniform as possible

  ⇒ minimize variance of signal to encode

- from a user perspective like zipping a file

redundancy coding

introduction

- **goals**
  - minimize bitrate
  - ensure that content can be reconstructed without error/loss
- **objective**
  - aim for a symbol pdf is as non-uniform as possible

  ⇒ minimize variance of signal to encode
- from a user perspective like zipping a file

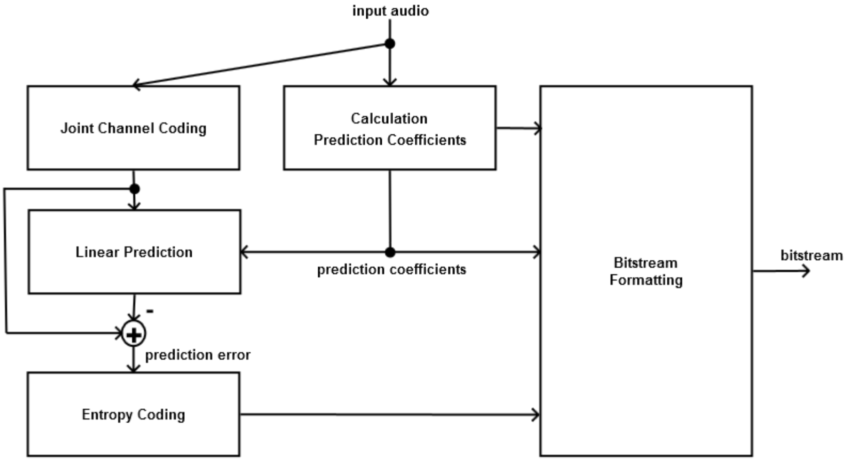
redundancy coding
introduction

- **goals**
  - minimize bitrate
  - ensure that content can be reconstructed without error/loss
- **objective**
  - aim for a symbol pdf is as non-uniform as possible

  ⇒ minimize variance of signal to encode
- from a user perspective like zipping a file

redundancy coding

standard system flow chart



redundancy coding
overview

- **properties**
  - *lossless*: perfect signal reconstruction
  - bitrate (reduction) *depends on input signal*
    - typical ballpark gain (stereo, 48k): factor 2
  - *no constant bitrate* → streaming only practical with large buffers and constraints on content

- **common applications/algorithms**

name	sampling rates	channels	word length
Shorten	all	2	8/16
FLAC	1-1048k	8	4-32
MLP	44.1k-192k	63	1-24
ALS	all	65536	1-32 (int), 32(float)

redundancy coding
overview

- **properties**
  - *lossless*: perfect signal reconstruction
  - bitrate (reduction) *depends on input signal*
    - typical ballpark gain (stereo, 48k): factor 2
  - *no constant bitrate* → streaming only practical with large buffers and constraints on content

- **common applications/algorithms**

name	sampling rates	channels	word length
Shorten	all	2	8/16
FLAC	1-1048k	8	4-32
MLP	44.1k-192k	63	1-24
ALS	all	65536	1-32 (int), 32(float)