

Überblick über MPEG-4 Audio Version 2

Heiko Purnhagen

Laboratorium für Informationstechnologie
Universität Hannover

Gliederung

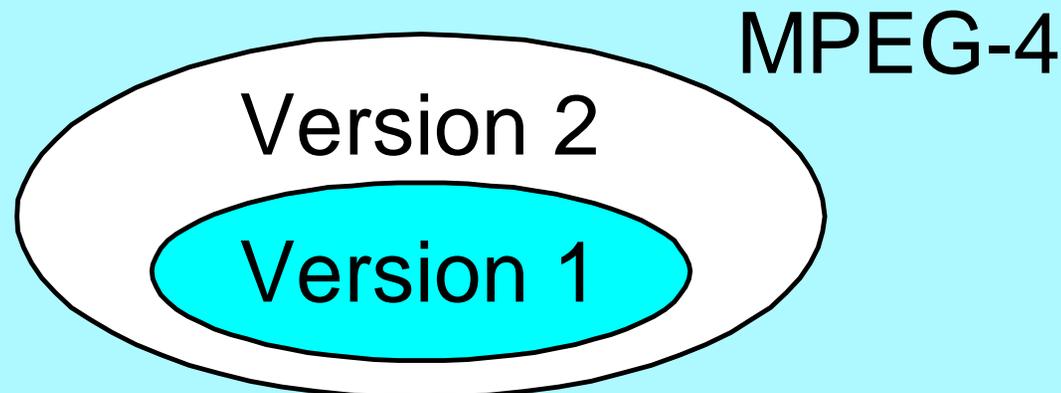
- Einleitung
- Rückblick: MPEG-4 Audio Version 1
- Neue "Tools" in MPEG-4 Audio Version 2
- Ausblick

Einleitung: MPEG-4

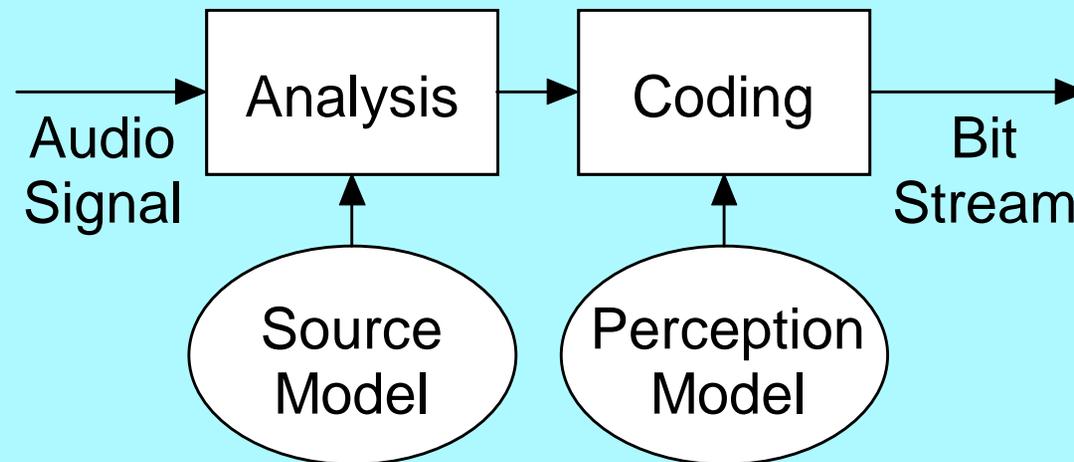
- Neue Multimedia-Anwendungen benötigen ...
 - effiziente & flexible Codierung von natürlichen & synthetischen audiovisuellen Daten
- ⇒ Entwicklung des MPEG-4 Standard:
"Coding of audiovisual objects"
- MPEG-4 Audio Version 1 (Okt. 1998)
 - Audio- und Sprachcodierung @ 2 .. 64+ kbit/s/ch
- MPEG-4 Audio Version 2 (Dez. 1999)
 - zusätzliche Funktionalitäten

Einleitung: MPEG-4 Versionen

- Problem: enger Zeitplan für MPEG-4
=> mehrere interessante "Tools" noch nicht reif
- Lösung: MPEG-4 Version 2 (Amendment 1)
 - rückwärts-kompatible Erweiterung von Version 1
 - neue "Tools" => zusätzliche Funktionalitäten



Version 1: Überblick



- MPEG-4 Audio -- Zielsetzung:
 - effiziente Codierung (belieb. Signale / Datenraten)
 - weitere Funktionalitäten (z.B. Skalierbarkeit)
- => Kombination von Codierungsverf. notwendig.
 - mehrere Quellen- und Wahrnehmungsmodelle

Version 1: Tools

- MPEG-4 Version 1

- Audio Tools: Codierung von Audio-Objekten

	Speech	General Audio
Natural	<ul style="list-style-type: none">• HVXC (param.) (2 .. 4 kbit/s)• CELP (NB+WB) (4 .. 24 kbit/s)	<ul style="list-style-type: none">• TwinVQ (6 .. 16 kbit/s/ch)• AAC (+scalable) (16 .. 64+ kbit/s/ch)
Synthetic	<ul style="list-style-type: none">• TTS-Interface	<ul style="list-style-type: none">• SA (incl. MIDI)

- Systems Tools: Mischen von Audio-Objekten
=> Audio-Szene (Effekte: Structured Audio)

Version 2: Überblick

- Neue Audio Tools
 - Fehler-Robustheit
 - Audiocodierung mit geringer Verzögerung
 - Feinstufige Skalierbarkeit
 - Parametrische Audiocodierung
 - "Silence Compression" für Sprachcoder
- Neue Systems Tools
 - 3-D Audio: "Environmental Spatialisation"
 - MP4 Dateiformat

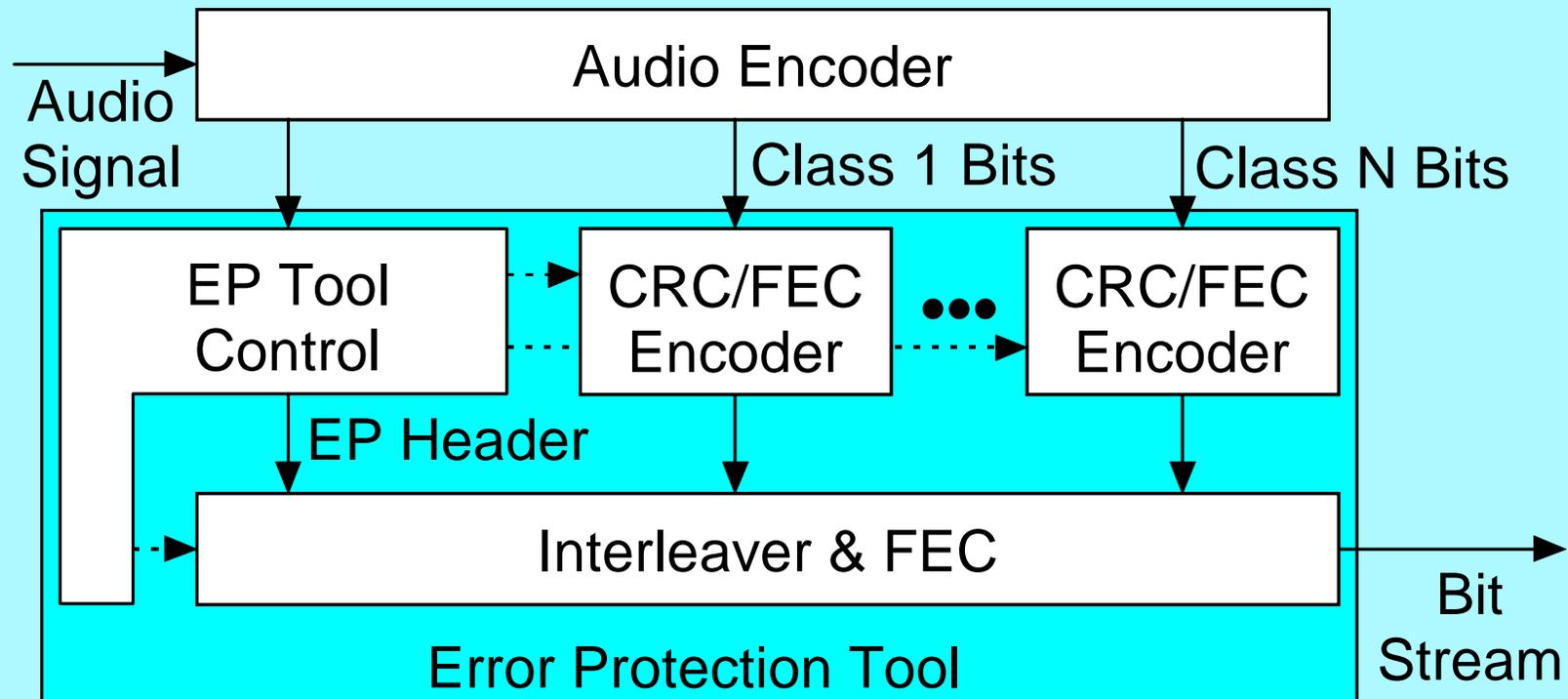
Version 2: Fehlerrobustheit

- Ziel: Übertragung über fehleranfällige Kanäle
- Ansatz: "Error Robustness" Tools
 - Fehlerschutz-Tool
 - Fehlerunempfindlichkeit der Codierungs-Tools

Version 2: Fehlerrobustheit

● Fehlerschutz-Tool

- Ungleicher Fehlerschutz: Fehlerempfindlichkeits-Klassen
- Cyclic Redundancy Check / Forward Error Correction



Version 2: Fehlerrobustheit

- Fehlerunempfindlichkeit der Codierungs-Tools
 - Fehlerunempfindlichkeit für AAC
 - z.B. Huffman Codeword Reordering
 - => reduzierte Fehlerfortpflanzung
 - Fehlerunempfindlichkeit für andere Tools
 - => geringere Qualitäts-Beeinträchtigung bei Übertragungsfehlern
- Verbergen von Übertragungsfehlern im Decoder (nicht normativ)

Version 2: Fehler-Robustheit

- **Demo:** AAC 48 kHz mono @ 64 kbit/s
Bitfehlerrate: 10^{-3}
 - keine Fehler-Robustheit
(nur Synchronisation)
 - Fehlerschutz und -unempfindlichkeit
(19% höhere Datenrate)

Version 2: Audiocod. mit geringer Verz.

- Ziel: Audiocod. mit geringer Verzögerung
 - bidirektionale Echtzeit-Kommunikation
 - AAC algorithmische Verzögerung:
z.B. 24 kHz @ 24 kbit/s
=> 110 ms + max. 210 ms (Bit-Reservoir)
- Ansatz: "Low-Delay Audio Coder"
 - abgeleitet aus AAC

Version 2: Audiocod. mit geringer Verz.

- Low-Delay Audiocoder (modifizierter AAC)

- Zeitrahmen & Filterbank delay

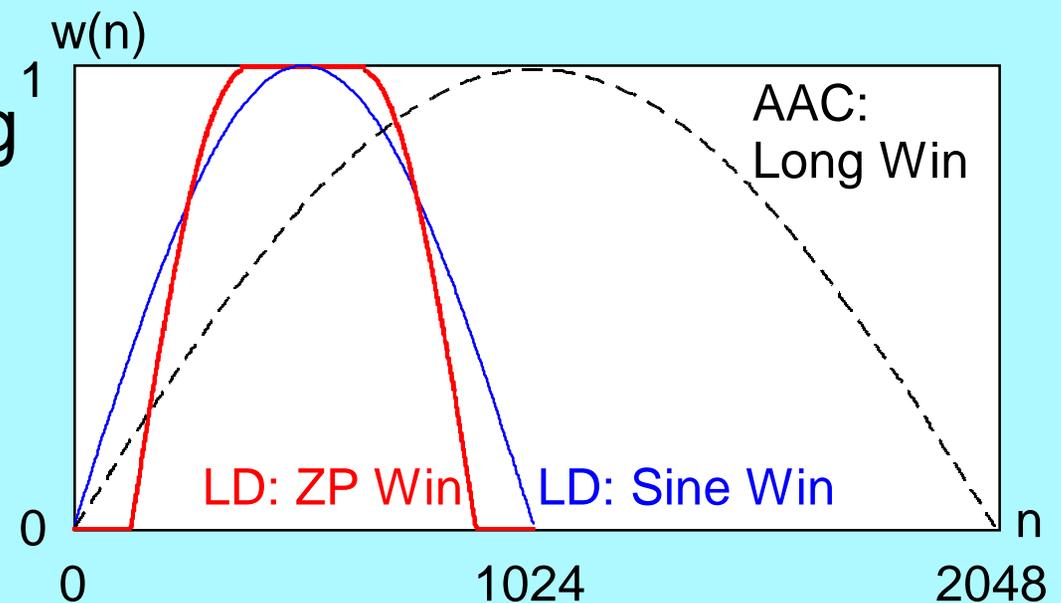
- ⇒ 1/2 Fensterlänge

- "look-ahead" für Fensterumschaltung

- ⇒ Zero-Padded Fenster für Transienten

- Bit-Reservoir

- ⇒ nicht genutzt



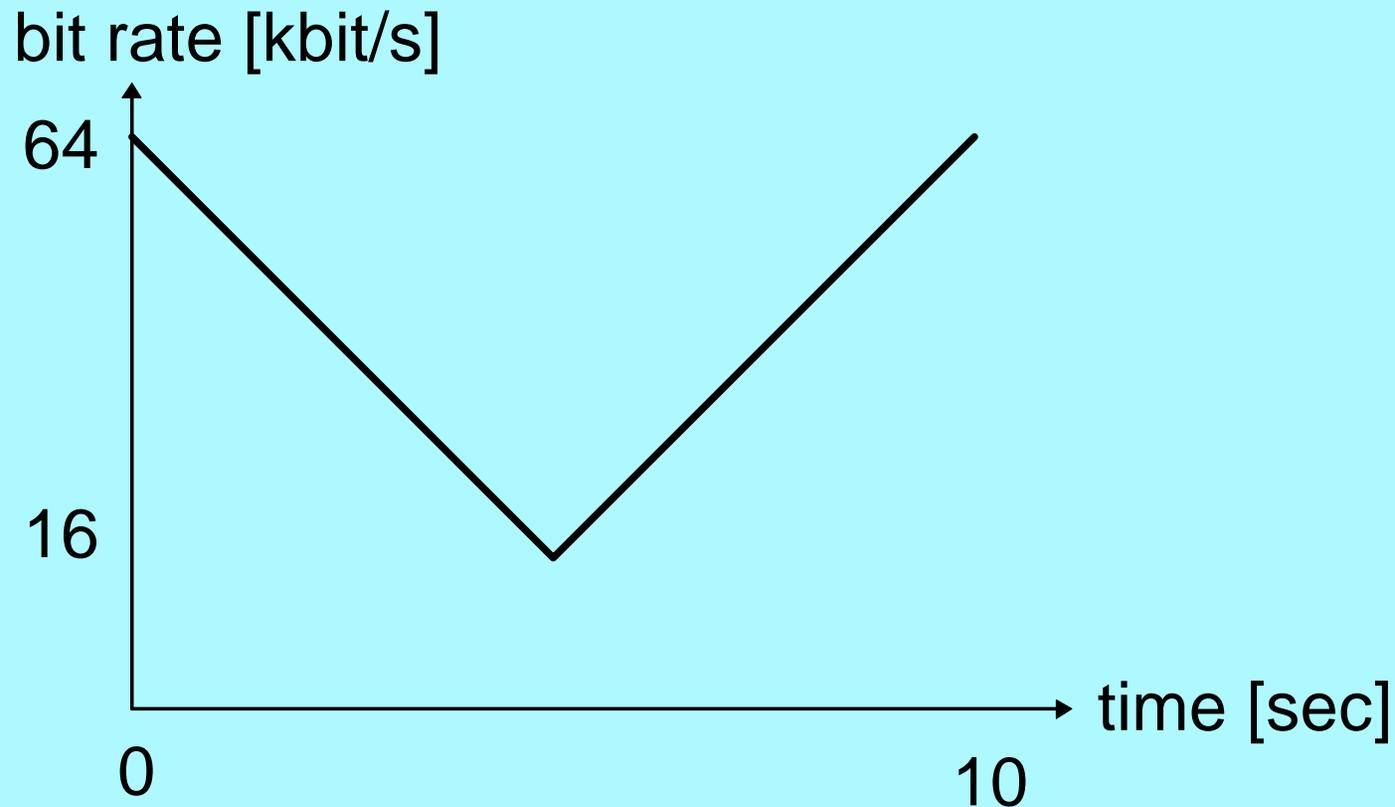
⇒ 20 ms algorithmische Verz. (48 kHz)

Version 2: Feinstufige Skalierbarkeit

- Ziel: Feinstufige Bitraten-Skalierbarkeit
 - AAC Skalierbarkeit: typ. 16 kbit/s Bitraten-Stufen
- Ansatz: "Bit-Sliced Arithmetic Coding"
 - Kombination mit AAC:
BSAC ersetzt AAC Huffman-Codierung
=> 1 kbit/s/ch Bitraten-Stufen
 - Funktionsweise:
erst "Bit-Scheiben" mit wichtiges Bits übertragen
dann: weniger signifikante Bits (feinere Quant.)
höhere Frequenzbänder

Version 2: Feinstufige Skalierbarkeit

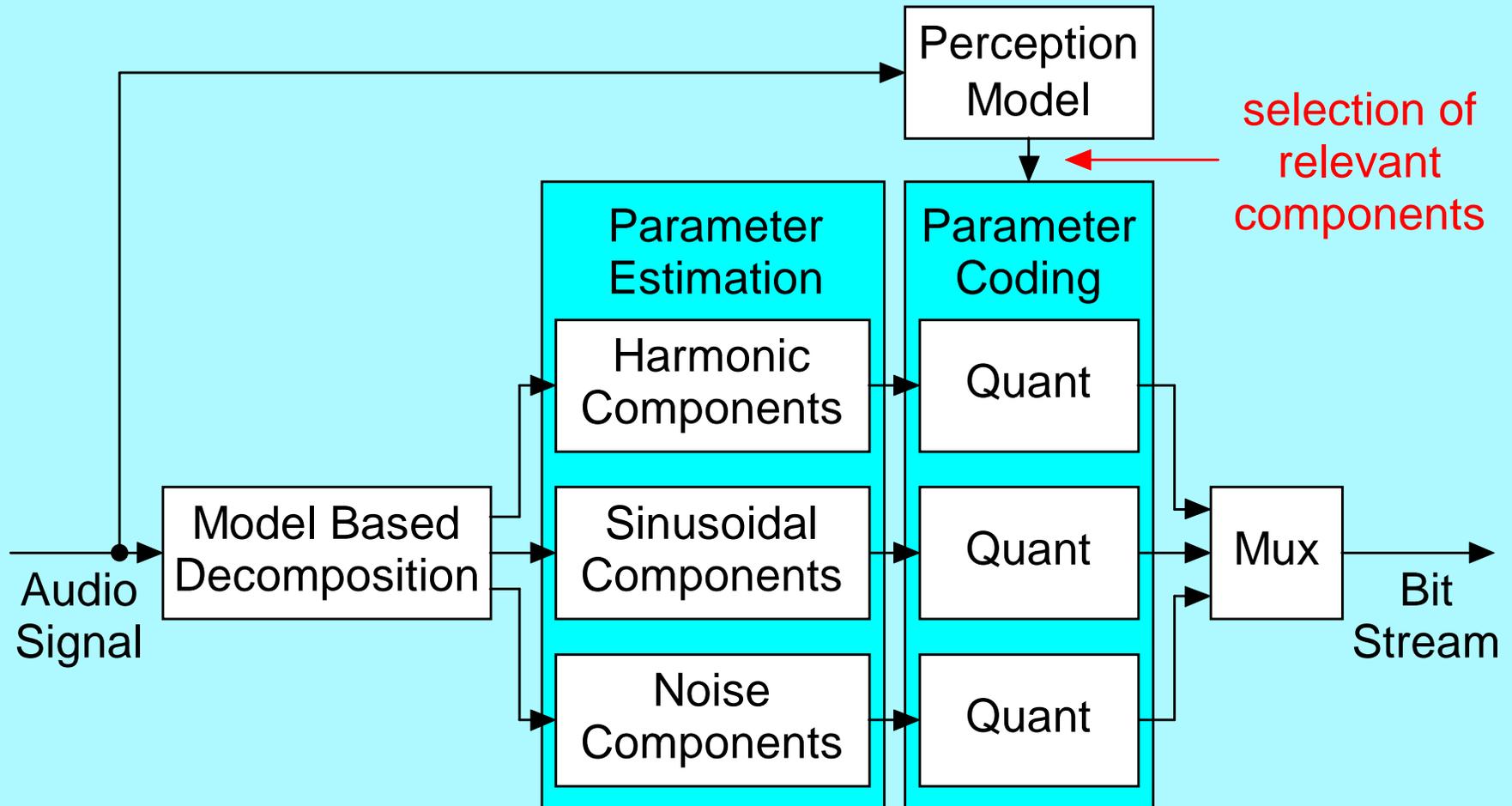
- **Demo:** BSAC 48 kHz mono @ 64 kbit/s



Version 2: Parametrische Audiocodierung

- Ziel: Audiocodierung für sehr niedrige Bitraten
 - Quellenmodell für Musik ?
- Ansatz: Parametrische Signaldarstellung
 - Zerlegen des Signals in Komponenten:
"Harmonic and Individual Lines plus Noise" (HILN)
 - Funktionalitäten:
sehr niedrige Bitrate (4 .. 16 kbit/s)
Tonhöhen- und Geschwindigkeits-Modifikation
Bitraten-Skalierbarkeit

Version 2: Parametrische Audiocodierung

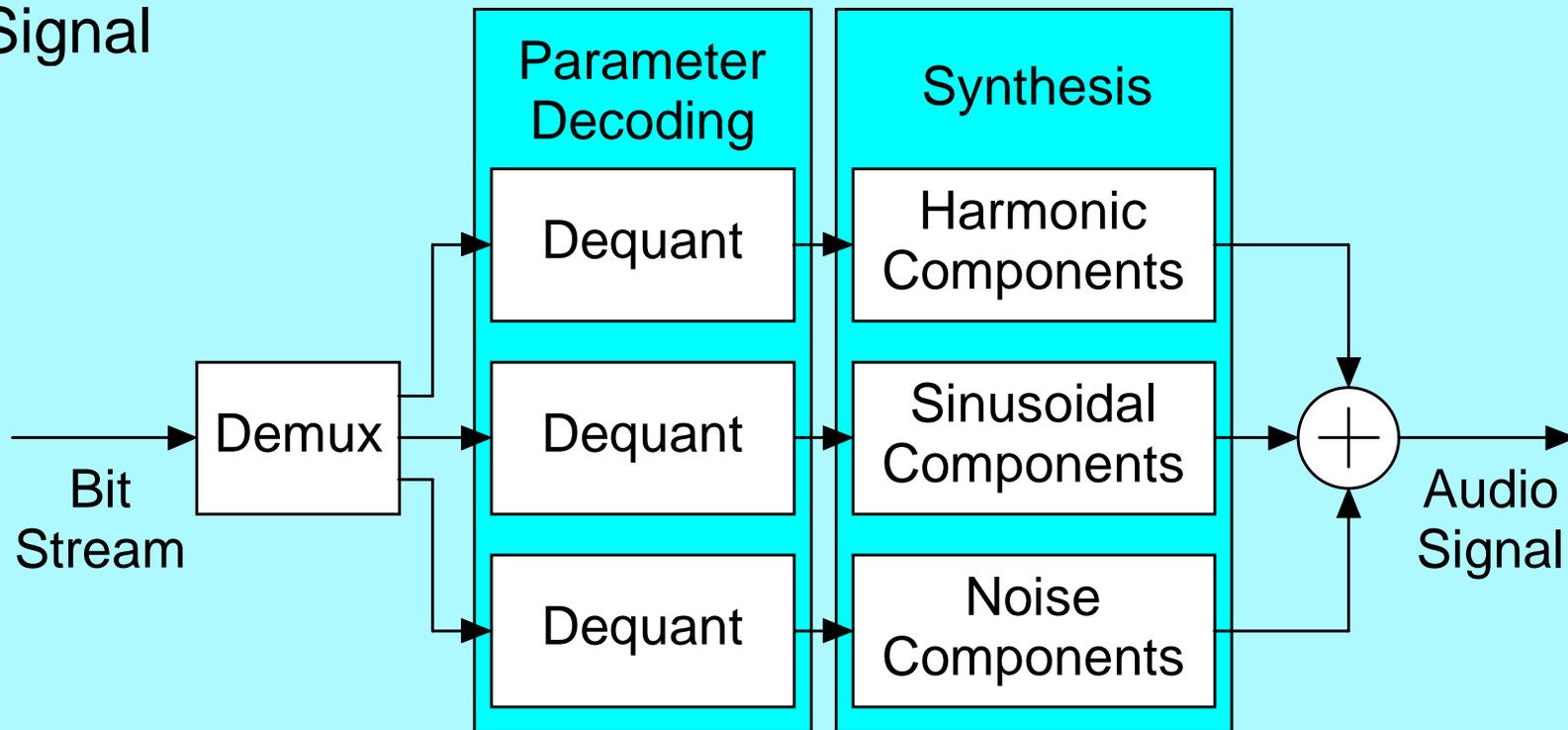


Parametric Audio Encoder (HILN)

Version 2: Parametrische Audiocodierung

- **Demo:** HILN 16 kHz mono @ 6 kbit/s

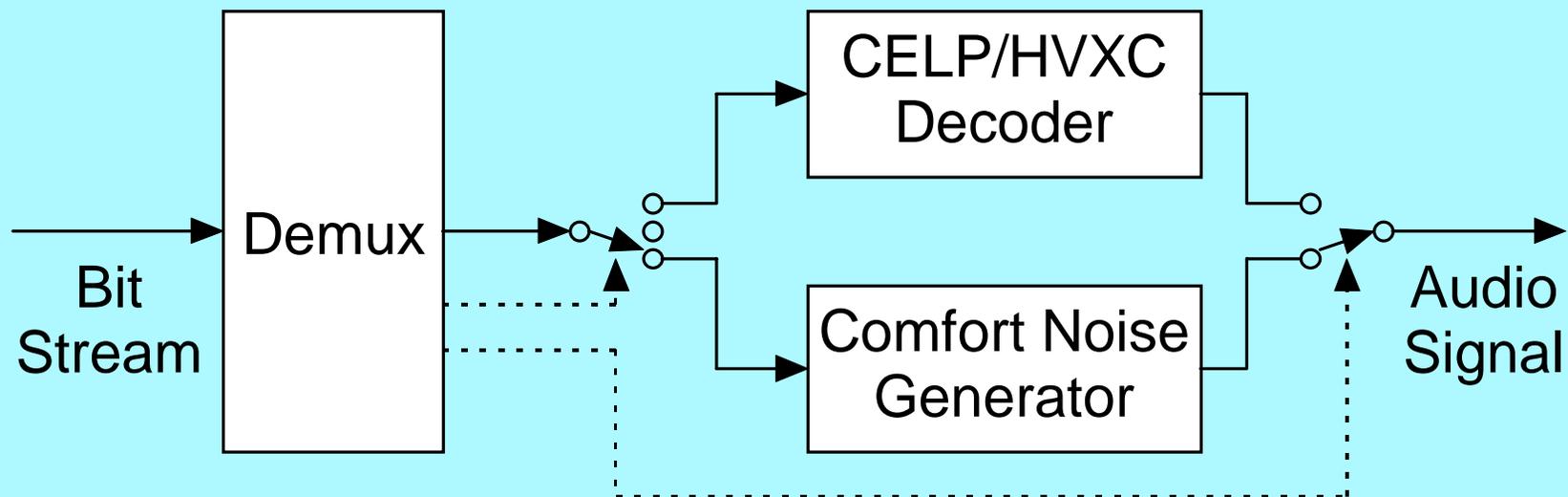
Original
Signal



Parametric Audio Decoder (HILN)

Version 2: "Silence Compression"

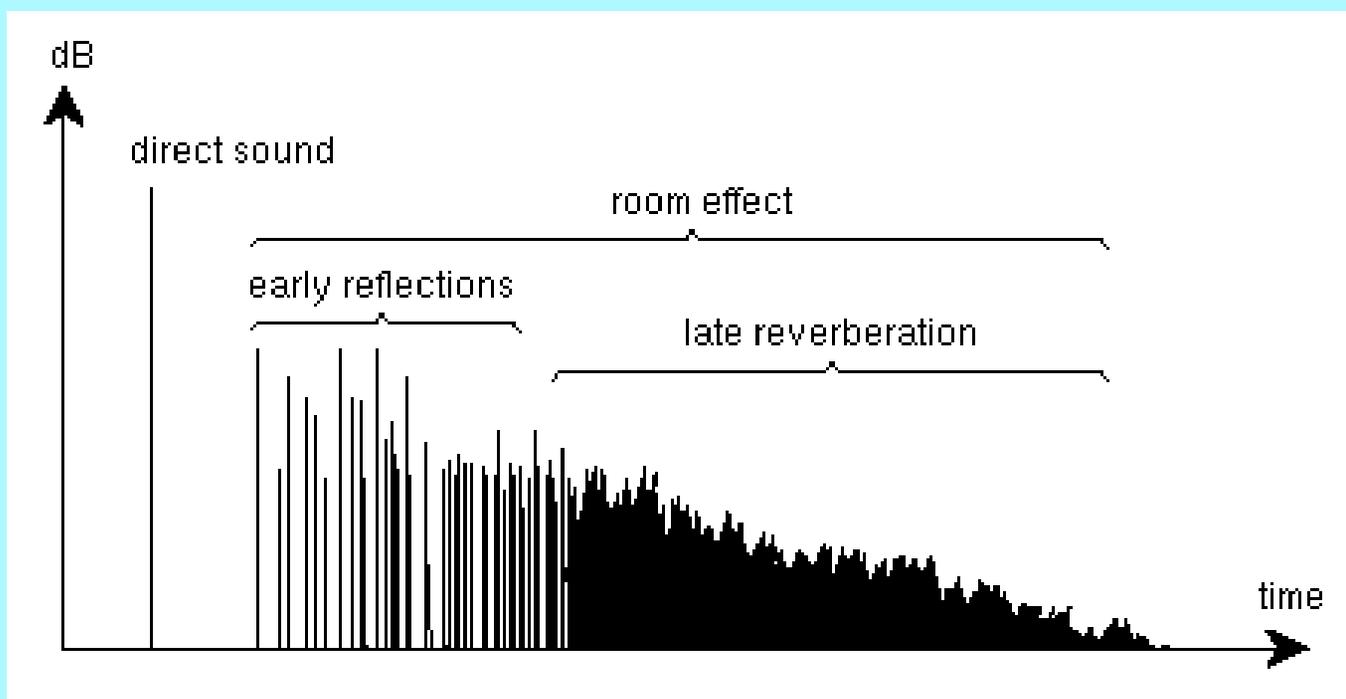
- Ziel: reduz. Bitrate wenn keine Sprachaktivität
 - bidirektionale Kommunikation: ~50% Ruhe
- Ansatz: HVXC/CELP "Silence Compression"
 - erzeugt "comfort noise" wenn keine Sprachaktivität



Version 2: "Environmental Spatialisation"

- Ziel: Effiziente und flexible "3-D" Audioszenen
 - SA-basierte Beschreibung umständlich für "3-D"
- Ansatz: physikalisch / wahrnehmungsorient.

typical
room
response



Version 2: "Environmental Spatialisation"

- Physikalischer Ansatz
 - Beschreibung der akustischen Eigenschaften (Raum-Geometrie, Position der Schallquelle, ...)
 - Audio- und Video-Szene korrespondieren z.B. "3-D virtual reality"
- Wahrnehmungsorientierter Ansatz
 - abstrakte, wahrnehmungsorientierte Beschreibung (Raumhall, Präsenz der Schallquelle, ...)
 - Audio- und Video-Szene unabhängig z.B. Kinofilm

Version 2: Weitere Tools

- MPEG-4 Dateiformat (MP4)
 - flexibles Format für:
Datenaustausch, Editierung, Vorführung
 - basiert auf QuickTime
- Rückkanal
 - z.B. für adaptives "Streaming"

Ausblick

- Version 2 Verifikations-Tests: Herbst 1999
- MPEG-4 Version 2 abgeschlossen: Dez. 1999
- Encoder-Optimierung möglich (nicht normativ)
- Neues Arbeitsgebiet: MPEG-7
"Multimedia Content Description Interface"
- MPEG Audio Web Page
<http://www.tnt.uni-hannover.de/project/mpeg/audio/>