3-D Audio and Virtual Acoustics

Riitta Väänänen, D.Sc, Senior Research Engineer
Nokia Research Center
Multimedia Laboratory
Riitta.Vaananen@nokia.com
(Partly based on slides of Jyri Huopaniemi
Nokia Research Center
jyri.huopaniemi@nokia.com)

Outline

• Introduction:
  • Components of acoustic communication chain taken into account in the modeling of virtual acoustics
  • Principles of virtual acoustics
• Virtual acoustics modeling:
  • Sound source modeling
  • Acoustic environment modeling
  • Receiver (=listener) modeling
• Digital reverberation:
  • Reproducing room acoustic effect by means of DSP
• 3-D audio and HRTFs:
  • Modeling of listener properties (with respect to sound source position)
  • Demos
  • Conclusions

Acoustical communication

Introduction - Principles of virtual acoustics
Virtual Acoustics - Building Blocks

Virtual acoustic modeling – computer and DSP based approach for recreating the effect of source, acoustic space and the listener

**Source Modeling: Sound production**

- From the viewpoint of virtual acoustics, sound sources and their corresponding models can be divided into two main classes:
  - **Natural audio** (sampled and recorded i.e. waveform-based audio)
    - Speech
    - Music
  - **Synthetic audio**
    - Sound synthesis
    - Speech synthesis
- Also hybrids exist!
  - e.g. wavetable synthesis
- Spatial properties of sound:
  - Source directivity
Source Modeling: Directivity & Radiation

- **Motivation:**
  - Natural sound sources never radiate sound equally to all directions
  - E.g., musical instruments, human head etc. all have very different sound radiation patterns that are also frequency dependent
- **Two main methods**
  - Direction-dependent filtering; source modeled as a single point source
  - Set of elementary point sources:
    - Also solves the defects of point source modeling
    - Not (yet) suitable for real-time applications
  - Direction-dependent filtering more widely used
    - Attach a directivity filter to each desired output direction
    - Suitable for real-time computation and dynamic situations

Sound source \[ R(\theta_1) \] \[ \Rightarrow \] \[ y_1(n) \]

Sound source \[ R(\theta_2) \] \[ \Rightarrow \] \[ y_2(n) \]

---

Acoustic environment modeling

- **Acoustic environment modeling:**
  - Free-field propagation of sound in the transmission medium:
    - Distance-dependent attenuation (usually according to 1/r law)
    - Air absorption
    - Finite propagation speed in the medium (e.g., 340 m in air):
      - Cause of Doppler effect in dynamic situations
      - Delay of reflected sounds with respect to the direct sound
  - **Interference with objects:**
    - Sound reflections off surfaces
    - Sound propagation through objects; Obstruction and occlusion
  - Both of the above factors are present in detailed room acoustic modeling

---

Example case: Spherical Head Model Directivity

- Magnitude
- Azimuth angle
- Frequency

---

Room Acoustics Modeling: Methods

- **Modeling of Room Acoustics**
  - Computational Modeling
  - Acoustic-scale Modeling
    - Wave-Based Modeling
    - Ray-Based Modeling
    - Statistical Modeling
    - Direct Modeling
    - Indirect Modeling
      - Difference Methods
      - Element Methods
      - Ray-Tracing
      - Image-Source Method
      - SEA
    - FEM
    - BEM
Room Impulse Response properties

- Room impulse response describes a transfer function between two points in a room.
- Basis for room acoustic parameters such as reverberation time (RT60) and early decay time (EDT).
- Different for all listener – source position combinations.
- Basis for real-time room acoustic modeling and auralization!
- Typical time-domain decomposition of room impulse response:
  - **Direct sound**
    - Delay of the direct sound depends on the source – listener distance.
    - Speed of sound ≈ 340 m/s; typical delay is 20...200 ms.
  - **Early reflections**
    - About 50...100 ms after the direct sound.
  - **Late reverberation**
    - Starts with ~100 ms delay with respect to the direct sound.
    - Late reverberation field usually considered diffuse (reflections arrive from all directions with equal probability).
    - Also often considered exponentially decaying.

Real (measured) room impulse response: Examples

- Short source – listener distance -> shorter delay, strong direct sound level compared to room effect:
- Long source – listener distance -> longer delay of direct sound, strong room effect energy compared to direct sound:

Real-Time Room Acoustics Modeling approaches

- **Direct room impulse response (DRIR) rendering**
  - Convoluted with a room impulse response (RIR).
  - RIRs calculated/measured a priori for predefined listening points.
  - E.g., image-source or ray-tracing method.
  - Binaural synthesis needed for 3D sound field experience.
  - Computational and memory requirements high.
  - Full RIR must be stored for each source and receiver position.
  - Computational requirement directly proportional to the length of the RIR (longer RIR needed for longer reverberation times).
  - For example: The computational complexity increases with the reverberation time (e.g., at 44.1 kHz sampling frequency, and 2.0 s reverberation time, for stereo reverb -> 2.0 * 44100 * 2 = 176400 multiplications per stereo sample -> 7.8 * 10^9 multiplications per second).
  - Real time modification of the reverberation properties (RT60 etc) -> Not suitable for interactive applications such as virtual reality and gaming.
- **Parametric room impulse response rendering (PRIR)**
  - Separate modeling of:
    - Direct sound
    - Early reflections: delays and directions derived from e.g., image source method.
    - Late reverberation: Usually produced with IIR (Infinite impulse response) DSP filter structures.
  - Use recursive filter structures (IIR = Infinite Impulse Response filters) to produce diffuse late reverberation.
  - Computationally more efficient.
  - Allows for true interactivity (dynamic source & listener movement).
  - Detailed modeling of acoustics computationally heavy (diffraction, diffusion).

Room Acoustics Modeling based on direct convolution

- **Real-time sound synthesis/use of waveform sound data**
- **Non-real-time analysis of room impulse response**

![Diagram of room acoustics modeling](image)
**Parametric Room Acoustics Modeling**

- **Real-time Synthesis of early reflections**
  - Image-Source Method

- **Non-real-time analysis of late reverberation**
  - Difference methods
  - Ray-tracing
  - Measurements

- **Acoustical parameters of the space (Reverberation time, energy of late reverberation)**

- **Direct sound and early reflections** + **Artificial late reverberation**

---

**Reverberation Algorithms**

- One of the most important pieces of studio equipment is a digital reverb!
- **The aim and challenge** is to create artificial but natural-sounding room effect (practically RIR) to a sound signal with a reasonable computational effort
- **Two different goals:**
  - Add ambience in a “dry” recording:
    - **Perceptual approach** -> User-controllable, adjustable perceptual parameters (reverb time, reverb level, presets)
    - Used in digital reverberators, for music post processing
  - Simulate the acoustics of a room or hall:
    - **Physical approach** -> Environment/geometry-controllable
    - For precise room acoustic design, modeling, virtual reality modeling
- **In both cases,** must imitate propagation delays of sound, reflections of sound, and attenuation caused by air absorption
- **Typical properties of a “good reverb”** (music and speech applications):
  - High echo/reflection density (no flutter echo)
  - High modal density (flat frequency response, no coloration)
  - Efficient parametrization (change reverb parameters interactively)
  - Some applications may have different quality criteria (e.g., computer games)

---

**How to simulate full RIR?**

- The time-domain decomposition allows for separate implementations
  - **Direct sound**
    - Different implementations when considering a musical effect or virtual reality rendering
  - **Early reflections**
    - Discrete “echos” coming from defined locations in space
  - **Late reverberation**
    - Exponentially decaying white or filtered white noise sequence
  - Direct sound implementation:
    - Either unmodified, original signal (e.g., stereo sound track) e.g., in music processing
    - In virtual reality rendering undergoes positional 3D processing that gives a precise direction of arrival to the sound
  - The early reflections can be implemented as a “sparse” FIR filter (Schroeder 1970)
    - Many coefficients of the FIR filter may be zero (between the reflections)
    - A long delay line with some taps to model the reflections (so called tapped delay line)
  - The tail of the RIR is approximated with a recursive digital filter structure
    - Comb filters with long delay lines, allpass-comb filters, ...
    - Still remains the most challenging/non-trivial part of the RIR modeling regarding computational performance (MIPS consumption) and memory consumption vs. quality

---

**Schroeder Algorithm**

- Schroeder (1962) developed the 1st reverb algorithm
  - Several comb filters in parallel
    - Different delay-line lengths, loop gain < 1
  - A few comb-allpass filters in cascade
  - To spread the impulse response in time -> simulation of diffuseness

**Impulse response of schroeder reverberator**
Comb filter response

- Comb filter response is exponentially decaying and has a comb-like structure of frequency-domain modes.
- Can be thought to simulate a sound reflecting between two parallel walls.
- Parallel connection of comb filters:
  - Increased reflection and modal density.
  - Delay line lengths must be carefully selected to avoid coinciding of reflections in the time domain and modes in the frequency domain.

Schroeder Algorithm (2)

- Schroeder (1962) invented the digital comb allpass filter:
  - Add a negative feedforward path to flatten the spectrum of a comb filter.
  - Long impulse response, such as that of a comb filter, and “almost” exponentially decaying.

Moorer Algorithm

- Moorer (1979) inserted a one-pole lowpass filter inside the comb filters:
  - Lowpass-comb filter
  - Long impulse response, where high frequencies decay faster than low frequencies.
  - More natural behavior.
  - Less “metallic” sound.

Moorer Algorithm (2)

- Moorer developed the algorithm further based on rectangular room modeling.
  - More comb filters is better (min. 6) - > resembles room reflections.
Other Structures

- Gardner has proposed nested allpass filters (Gardner 1998)
  - Replace part of delay line with an allpass filter
  - The overall structure is still allpass, but more dense impulse response

- Dattorro has published a “plate reverb” algorithm (Dattorro 1997)
  - Based on ideas used in reverb units by Lexicon
  - A cascade of allpass-comb filters with time-varying delay lines and some feedback paths

Feedback Delay Network

- A generalized comb filter structure that uses a feedback matrix (Jot and Chaigne 1991)
- Researchers at IRCAM developed a perceptual parameter mapping from psychoacoustic listening tests to this reverb structure
  - IRCAM Spatialisateur
  - MPEG-4 Advanced BIFS, perceptual approach (see next slide)

Monophonic vs. multichannel reverbs

- Room responses that humans hear are binaural -> the task of a reverb is actually to model a Binaural Room Impulse Response (BRIR)
- From the computational point of view it is not optimal to produce a stereo signal using “brute force”:
  - BRIR convolution
  - Run two identical late reverb modules
- Most current reverbs use a “pseudostereo” approach
  - Take uncorrelated outputs from a monophonic late reverb network
- Schroeder (1962) proposed a means to produce uncorrelated multiple outputs by computing linear combinations of the comb filter outputs
- Jot (1992) has proposed an interaural cross-correlation (IACC) post-processing algorithm to control the amount of cross-correlation

Perceptual Reverb Parametrization

- Aim is to give user control over the quality of the acoustic effect based on a perceptual, intuitive set of parameters
- Time and frequency-domain decomposition:
  - Direct sound (R0)
  - Directional early reflections (R1)
  - Diffuse early reflections (R2)
  - Diffuse late reverberation (R3)
  - Three frequency bands (low-mid-high)
- Source parameters:
  - Source presence (energy of the direct sound and early room effect)
  - Brilliance and warmth (variation of early energy with frequency)
- Room effects:
  - Room presence (energy of late room effect)
  - Running reverberance (early decay time)
  - Envelopment (energy of early room effect relative to direct sound)
- Late reverberation:
  - Late reverberance (reverb time)
  - Heaviness, Liveness (decay time as a function of frequency)
Physical Reverb Parametrization

- Use physical modeling principles to model sound propagation
  - Typical methods: ray-tracing, image-source modeling
- Typical properties “automatically” derived from a physical model
  - Source directivity
  - Distance dependent attenuation
  - Air absorption, Doppler effect
  - Modeling of room reflections, transmission and late reverberation based on measurements or models of existing spaces

Modeling of Boundary Reflections

- Simple models of reflection and transmission
  - octave-band absorption coefficient data (125, 250, 1000, 2000, 4000 Hz)
  - used in room acoustic prediction (geometrical acoustics models)
- Examples:
  - a) plasterboard on frame and glass panel (2 materials), 1st order IIR pole-zero model
  - b) plasterboard and fibreboard (2 materials), 3rd order IIR pole-zero model

Distance Effects

- Distance attenuation
  - Sound attenuation inverse proportional to distance
- Air absorption
  - function of temperature, humidity, and distance
  - analytical methods exist for determining air absorption
- Distance dependent delay
  - Doppler effect

Considerations for Audio Playback

- Headset / headphones
  - Personalized listening, suitable for small devices
  - Enhanced experience using 3D processing
- 2 Speakers
  - Community listening
  - Stereo & 3D audio with loudspeakers (e.g. stereo dipole)
  - Virtual surround for reproducing multichannel material
- Multichannel audio
  - 5.1 Dolby Digital, MPEG-2 and MPEG-4
  - DVD, DVB, home theatre systems
**Spatial sound; Three-dimensional (3-D) sound**

- Hearing is omnidirectional!
- Detecting the angle of arrival of sound depends on the interaural time and level differences between the ears and on the location-dependent spectral filtering
- Three-dimensional sound illusion can be achieved using headphones (or a pair of loudspeakers). We can "cheat" the auditory system by replicating spatial audio cues!

**Head-related transfer function processing**

- Torso, shoulders, head and pinnae modify the perceived spectrum as a function of the incoming angle of sound
- Room acoustics (early reflections), head movements and vision also contribute to spatial hearing sensation
- HRTF (head-related transfer function) = Free-field impulse response from a point in space to a point in the listener's ear canal

**Headphone and 2-speaker reproduction of 3D sound**

- Practical approaches for 3D sound in mobile devices
- Improvements: head tracking to keep the virtual sound sources still

**Multi-loudspeaker reproduction of 3D sound**

- Aim is to reproduce a sound field that either physically or perceptually resembles the sound field caused by real sound sources
- Multi-loudspeaker methods for 3D sound reproduction:
  - Amplitude panning (e.g., Vector-Base Amplitude Panning)
  - Ambisonics, holophony
  - Wave-field synthesis

Wave-field synthesis example: (Carrouso project)
Doppler effect, obstruction and occlusion

- **Doppler effect:**
  - Perceived when there's relative velocity between the sound source(s) and the listener.
  - Essentially a pitch shift that varies as a function of the relative speed between the source and the listener.
  - Pitch raises when distance decreases and vice versa.

- **Obstruction:**
  - The (frequency-dependent) attenuation of direct sound when there is an object on the direct path between the listener and the sound source.
  - Filters only the direct sound, the reverb/room effect remains unchanged.

- **Occlusion:**
  - The (frequency-dependent) attenuation perceived when the source is in a different room from the listener.
  - Affects (=filters) both direct sound and the room response (reverb).

Combining the spatial sound DSP into virtual acoustic modeling scheme

- High-level blocks needed in virtual acoustic modeling (2-channel reproduction example):
  - Distance attenuation and direction filtering
  - Doppler effect
  - Occlusion
  - Obstruction
  - 3D sound processing (positioning of source)

- In dynamic applications such as computer games time-variance is present in all the processing phases.
- Coefficient interpolations, delay line length interpolations momentarily require more processing than the whole DSP network in a static situation.

Headphone or Loudspeaker Playback?

- **Headphone binaural reproduction**
  + No crosstalk, HRTFs usable "as is" (with proper equalization).
  + Headphone placement stable, head-tracking easily integrated.
  + Listening environment does not interfere with virtual sound image.
  - In-head localization, front/back confusions frequent.
  - Difficulty to externalize sound image on the median plane.
  - Individual performance.

- **Loudspeaker binaural reproduction**
  + Good frontal imaging and out-of-head localization.
  + Not as listener-dependent as headphone listening.
  + Stereo Dipole (closely-spaced loudspeakers) suitable for high-quality binaural playback, applicable for multimedia monitor use.
  - Cross-talk canceling highly dependent on listening position (the "sweet spot" problem).
  - Off-axis listening results in image collapse, colored sound.
  - Difficult to create 360 degree imaging.
  - Maximum imaging area normally approx. 240 degrees.
  - Listening room acoustics (early reflections) may deteriorate sound imaging.

- **Multichannel (more than 2 channels) reproduction:**
  + Better localization behind the listener.
  - Less portable, non-standard systems.
Multimedia standards supporting 3D audio and virtual acoustics

- MPEG-4 multimedia standard
- Java3d API (not dealt with here)
- Java API's JSR-135 and JSR-234 for mobile devices

MPEG-4 Multimedia standard

- MPEG-4 is a standard for interactive multimedia applications such as digital television, efficient audiovisual compression, 3D virtual reality and games applications
- Synthetic and natural audio and video
- MPEG-4 consists of several parts, among them are the audio-visual coding tools:
  - Audio coding including synthetic and natural audio coding tools:
    - Generic Audio coding (incl. E.g., AAC), speech coding, parametric audio coding, text-to-speech synthesis interface, structured audio coding
  - Video coding:
    - Video compression (natural video coding)
    - Synthetic coding, incl 3D graphics animation, face and body animation
  - Scene description, Binary Format for Scenes (BIFS):
    - Enables hierarchical composition of MPEG-4 data (audio + visual) into interactive multimedia presentations
    - Interactivity is included in the encoded content, implemented locally when the scene/application/multimedia presentation is decoded
    - Representation format is based on the Virtual Reality Modeling Language (VRML); Hierarchically connected nodes define the structure of an MPEG-4 application
- Profiles are defined to enable subsetting of the functionalities

Introduction to Audio Scene Description

- Audio scene description in MPEG-4:
  - AudioBIFS: Sound API for generic mixing and advanced DSP processing of sounds, spatial positioning
  - Advanced AudioBIFS: Extension to AudioBIFS with more advanced spatial sound (virtual acoustic) processing capabilities
- AudioBIFS: Set of BIFS nodes used for:
  - Hierarchical, object oriented composition of audio data
  - Simple 3-D and 2-D presentation
  - Part of the 1st Edition of MPEG-4 Systems
- Advanced AudioBIFS = Extension to AudioBIFS for:
  - Advanced modeling of acoustic environment
  - Acoustic effects in 3-D scenes
  - Added to 2nd Edition of MPEG-4 Systems
Content Composition of Sound in BIFS

- **AudioBIFS** nodes used to build up an audio sub-tree:
  - Defines how the presented sound is combined of various audio elementary streams
- **AudioBIFS** allows SNHC (Synthetic-Natural Hybrid Coding) of sound:
  - Combining different sound streams (encoded with different coding tools) and including them in a single MPEG-4 presentation

Advantages of Content Composition

- **AudioBIFS** subtree ensures *high-quality playback* of sound streams (includes synchronization & sample rate conversion of streams)
- Enables *local effects processing and filtering* at the terminal by using the Structured Audio methods (**AudioFX** node)
- The resulting audio track may be coded at a *lower bitrate* than if a similar sound was encoded with wideband audio coding
- **AudioBIFS** also allows *local interaction from the user* (or other BIFS nodes) to manipulate the properties of BIFS nodes and thus the presentation of sound at the terminal

Example Audio BIFS scene for composition and presentation of sounds

- **AudioBIFS Composition**
  - Sound nodes: Attach sound to scene
  - Recorded wideband music
  - Physical instrument model (e.g., waveguide)
  - Speech
  - **AudioBIFS:** Sound can be spatially presented in a 2-D or 3-D audiovisual scene:
    - 3-D scenes: **Sound** node = VRML positional sound model (3-D position rendering not normative)
    - 2-D scenes: **Sound2D** = 2-D sound model in x-y plane
  - **Advanced AudioBIFS** in version 2 of MPEG-4
    - Enhanced presentation of sound sources in 3-D scenes with **DirectiveSound** node

Spatial Presentation

- **AudioBIFS:** Sound can be spatially presented in a 2-D or 3-D audiovisual scene:
  - 3-D scenes: **Sound** node = VRML positional sound model (3-D position rendering not normative)
  - 2-D scenes: **Sound2D** = 2-D sound model in x-y plane
- **Advanced AudioBIFS** in version 2 of MPEG-4
  - Enhanced presentation of sound sources in 3-D scenes with **DirectiveSound** node
Advanced AudioBIFS

- Modeling of sound source radiation:
  - Directivity (can be frequency dependent)
- Transmission in the medium:
  - Attenuation, air absorption, propagation delay (causing also Doppler effect)
- Modeling of acoustic effect ("room" acoustics) of the environment:
  - Sound reflections
  - Propagation through objects
  - Reverberation

Advanced AudioBIFS cont.

- **Physical** and **perceptual** approaches for modeling an acoustic effect
- **Physical** approach:
  - Based on defined geometry of acoustically responding objects
- **Perceptual** approach:
  - Allows adding a spatial sound effect based on set of perceptual room acoustic parameters

Advanced AudioBIFS cont.

- **Physical**:
  - Immersive audiovisual VR systems: Reflections, obstruction, reverberation are in synchrony with the visual scene
  - The spatial effect automatically updated according to changes in the scene
- **Perceptual**:
  - Room acoustic effect (reverberation) implemented more efficiently and dependent only on the relative positions of the source and the listener
  - Good for, e.g., in audio-only applications, post-production of music
Shape { 
appearance { 
material AcousticMaterial { 
diffuseColor 0.2 0.2 0.2 
reffunc 0.4 
transfunc 0.2 
} } 
group { 
indexedFaceSet { 
coord ... 
coordIndex ... 
} } 
} 

Perceptual approach

- Room response characteristics can be defined in 4 time sections and in 3 frequency bands:

```
PerceptualParameters {
  sourcePresence 1.0
  sourceWarmth 1.0
  sourceBrilliance 1.0
  roomPresence 1.0
  runningReverberance 1.0
  envelopment 0.0
  lateReverberance 1.0
  heavyness 1.0
  liveliness 1.0
  omniDirectivity 1.0
  directFilterGains 1.0, 1.0, 1.0
  inputFilterGains 1.0, 1.0, 1.0
  refDistance 1.0
  freqLow 250.0
  freqHigh 4000.0
  timeLimit1 0.02
  timeLimit2 0.04
  timeLimit3 0.1
  modalDensity 0.8
}
```

General Java background

- Why is Java so attractive for mobile devices?
  - It is cross-platform ("write once, run anywhere")
  - It is an open standard
  - It is manufacturer- and region-independent
  - It is a widely accepted programming language

Which version of Java do mobile phones use today?
- J2ME + CLDC1.0 + MIDP1.0
- Also MIDP 2.0 (+ CLDC 1.1) phones have been introduced to the market (such as Nokia 6600, 7610)

Useful acronyms:
- J2ME = Java2 Micro Edition
- CLDC = Connected, Limited Device Configuration
- MDP = Mobile Information Device Profile
- KVM = K(=kilobyte) virtual machine (small runtime environment for Java/CLDC)
- JCP = Java Community Process (Java standardization process)
- JSR = Java Specification Request (Java standard)
- Mobile device APIs:
  - JSR-110 = MIDP 2.0
  - JSR-135 = Mobile Media API (MM API)
  - JSR-184 = Mobile 3D Graphics API
  - JSR-226 = Scalable 2D Vector Graphics API
  - JSR-234 = Advanced Multimedia Supplements (AMMS)
Java multimedia technology overview

Specification efforts

Existing platform in 2003

MIDP 2.0
JSR-135
JSR-234

CLDC

- CLDC = Connected, Limited Device Configuration
- Minimum footprint Java platform for small, resource constrained, connected devices
- KVM = K Virtual Machine ~160kb - 512kb memory
- 16- or 32-bit processor
- Often battery powered, limited bandwidth (wireless) connection
- Targets to cell phones, communicators, PDA's,... that are manufactured in large quantities
- Defines API's for
  - Core Java libraries (java.lang.*, java.util.*)
  - Input/Output (java.io.*)
  - Networking (generic connection framework)
  - Security
  - Internationalization
- Pre-verification, applications checked before installed in the device
- Missing e.g. floating points, JNI, finalize(), ...
  - Floating point support targeted for CLDC 1.1

MIDP

- Mobile Information Device Profile
  - Version 1.0 finalized in 2001
  - Version 2.0 finished in fall 2002, already in products
- Memory requirements:
  - MIDP 1.0
    - Non-volatile memory: 128 kB for MIDP, 8 kB for application-created persistent data
    - 32 kb runtime memory
  - MIDP 2.0
    - Non-volatile memory: 256 kB for MIDP, 8 kB for application-created persistent data
    - 128 kB runtime memory
- MIDP 1.0 - Defines API's for:
  - Application (semantics and controls of app)
  - UI
  - Persistent storage (RMS)
  - Networking (HTTP supported)
  - Timers
- MIDP 2.0 - Enhancements over MIDP 1.0:
  - Enhanced UI, better support for multimedia and games
  - Sound
  - Better connectivity
  - Security model

What is JSR-135?

- JSR-135 = Java Mobile Media API (MMAPI) Standard
- A time-based multimedia API
- Optional package for MIDP/J2ME
- Designed by the following companies:

  Aplix Corp.
  Beatnik, Inc.
  France Telecom

  Insignia
  Mitsubishi
  Motorola

  Netdecisions
  Nokia (Spec Lead)
  NTT DoCoMo

  Openwave Systems
  PacketVideo
  Philips

  Siemens
  Smart Fusion
  Sun Microsystems

  Symbian
  Texas Instruments
  Vodafone

  Yamaha Corp.
  Zucotto Wireless


Requirements and Features of MMAPI

• Main features
  • Small footprint
  • Used with CLDC and MIDP
  • Provides sound and multimedia support for CLDC devices:
    • Tone generation and playback
    • Sampled and synthetic audio playback, streaming and recording
    • Interactive MIDI
    • Video playback, streaming, and recording
    • Media synchronization
  • Only features supported by the terminal must be implemented
• Compatible subset of JSR-135 included as a building block in MIDP 2.0
  • Tone generation (mandatory)
  • Support for sampled and synthetic audio playback (optional)

Java JSR-234; Advanced Multimedia Supplements

• Java Advanced Multimedia Supplements (AMMS) is a recent JSR API specification to be used with CLDC/MIDP 2.0
• AMMS is based on MMAPI (JSR-135) framework (extends its functionalities)
• New features include:
  • Advanced audio
  • Effect network
  • Media post-processing
  • Advanced digital camera
  • Format management and conversions
  • Radio control
• Advanced audio enables virtual acoustics and audio effects processing

Advanced audio with Java AMMS API

• Virtual acoustics:
  • Real-time simulation of sound propagation in an acoustic space
  • Varies along with listener/sound source movements
  • Requires modeling of the sound source, room acoustics, and the listeners hearing characteristics
• Audio Effects:
  • Audio algorithms implemented in DSP
  • Performed in real-time in the playback “terminal” (e.g., the mobile device)

Virtual Acoustics

Modelling

Source → Room → Listener
Virtual Acoustics

Modelling in Java

Controls from Player

Source

- Location
- Directivity
- Doppler
- Macroscopic
- Obstruction

- DistanceAttenuation

Global Controls from GlobalManager

Room

- Reverb
- Commit

listener Controls from Spectator

Listener

- Location
- Orientation
- Doppler

Audio Effects

- Equalizer
  - N-band equalizer
  - Bass and treble for convenience
- Chorus and flanger
- Reverberation
  - Set of environmental and musical presets mandated
  - Setting of the reverb time (RT60) and level
- Audio virtualizer
  - Stereo widening and virtual 5.1

Sound API links

- OpenAL: http://www.openal.org/
- DirectSound (Microsoft): http://msdn.microsoft.com/library/
- IASIG (Interactive Audio Special Interest Group) has defined common guidelines for 3D sound API's (I3DL2 = Interactive 3D audio rendering guidelines, Level 2): http://www.iasig.org/pubs/3dl2v1a.pdf
- MPEG-4: http://www.chiariglione.org/mpeg/

Summary

- Virtual acoustics modeling consists of modeling of the sound source, transmission medium, and the receiver (listener)
- Real-time interactive room acoustic modeling requires parametrization of the room impulse response
- Due to increasing popularity of interactive multimedia devices and applications
- Applying and design of digital audio effects is moving from studios (professional equipment) towards “light” consumer (mobile devices and home computers)
- The algorithms need to be efficient, modular and scalable:
  - DSP implementations must run in real time
  - Limitations of memory and processing power need to be taken into account in the algorithm implementation design
- They should be interconnectable (networking of effects)
- Interactive, real-time modification need to be taken into account in the implementation design
- Currently much effort is also put on API design:
  - Easy programmability of effects
  - Cross-platform independency of applications: Same APIs may have implementations with different complexities depending on the hw resources of the target device
  - 3D sound APIs are gaining interest and popularity. Needed particularly for gaming applications and enhanced sound reproduction
  - Examples: MPEG-4 Standard, DirectSound API, Java standard API JSR-234 (Advanced Multimedia Supplements)