A Framework for the Development of Accurate Acoustic Calculations for Games

By
Panagiotis Charalampous & Panos Economou

Mediterranean Acoustics Research & Development Ltd
Historical Development

Late 1970’s – Early 1980’s: Slow Processing Times
(Naïve algorithms, slow CPUs)

- Prediction
- Accuracy

2013: Fast Processing Times (Advanced Algorithms, multicore CPUs, Programmable GPUs)

- Speed
- Interactivity

Fast and accurate predictions for both interactive and engineering applications
PEMAR’S FRAMEWORK

- Diffraction Calculations
- Distances Attenuation
- Model Optimization
- Sound Propagation Algorithms
- Refections
- Finite Impedance
- Time Domain Calcs
- Auralization
What is a Framework

An abstraction in which a software offering generic functionality can be selectively changed by user code, resulting in a specific software
PEMARD Framework

- A software architectural model which outlines a pattern that can be used in sound propagation calculations and defines a process for the calculation of sound propagation in 3D environments.
Sound Rendering Process

- Optimization
- Preprocessing
- Path Detection
- Path Calculation
- Auralization
Sequence Diagram

3D Geometry

- Optimized 3D Geometry
- Preprocessed 3D Geometry
- Sound Paths
- Sound File

Optimizer

Preprocessor

Path Detector

Path Calculator

Sound Paths
Optimization

- Sound propagation algorithms performance is based on the model size.
- Most of accurate propagation algorithms have a complexity of $O(n^k)$
- 3D CAD or game models usually contain information relevant to graphics rendering which could be irrelevant to sound rendering.
- The optimization step is the step where the model’s information is reduced to the geometrical detail necessary for acoustical calculations.
Optimization
Preprocessing

- Preprocessing is the step where we extract required metadata about the model.
Sound Path Detection

- Sound path detection refers to the process of finding the sound paths from source to receiver.
Sound Path Detection

- Sound path detection refers to the process of finding the sound paths from source to receiver.
Sound Path Calculation

- Sound path calculation is the step where the contribution of each sound path and the total contribution at each source are calculated.
- We use the following expression

\[ p_{total} = \sum_{i=1}^{n} p_i \frac{e^{jkR_i}}{R_i} \prod_{j=1}^{m} C_j \]

Where:
- \( p_{total} \) is the total sound pressure at a receiver, of all sound propagation paths from all sources,
- \( p_i \) is the total sound pressure at a receiver, of all sound propagation paths from one source
- \( n \) is the number of sound propagation paths from source to receiver
- \( k \) is the wavenumber
- \( R_i \) is the path length between a source and receiver
- \( C_j \) is any coefficient that represents a sound phenomenon e.g. reflection, diffraction, atmospheric absorption etc.
- \( m \) is the number of coefficients.
Auralization

\[ g(t) = \int_{-\infty}^{+\infty} s(\tau)f(t - \tau) \, d\tau \]
// We assign an optimizer object to the geometry
// object's Optimizer property
_geometry.Optimizer = new Optimizer();
// We assign an preprocessor object to the geometry
// object's Preprocessor property
_geometry.Preprocessor = new Preprocessor();
// We assign an path detector object to the Geometry
// object's Detector property
_geometry.Detector = new PathDetector();
// We add calculations in the the Geometry object's
// Calculations list
_geometry.Calculations.Add(new ReflectionCalculation());
_geometry.Calculations.Add(new HPSSDiffractionCalculation());
// We optimize and preprocess the Geometry
_geometry.Optimizer.Optimize(_geometry);
_geometry.Preprocessor.Preprocess(_geometry);
// We get the sound paths of the geometry
var paths = _geometry.GetAllPaths();
// We calculate these paths
_geometry.Calculate(paths);
// We iterate through the sound receivers if the
// Geometry and see the results
foreach (var receiver in _geometry.GetAllReceivers())
{
    // The impulse response at the receiver
    var ir = receiver.PreciseResults.IR;
}
## Framework Application

<table>
<thead>
<tr>
<th>Optimization</th>
<th>Preprocessing</th>
<th>Path Detection</th>
<th>Path Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Unnecessary triangles removal</td>
<td>• Distinct edges determination</td>
<td>• Reflections detection with visibility tracing</td>
<td>• Sound diffraction coefficients.</td>
</tr>
<tr>
<td></td>
<td>• Edge to triangles association</td>
<td>• Diffractions detection</td>
<td>• Spherical wave reflection coefficient.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reflection - diffraction detection</td>
<td>• Geometrical spreading.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Atmospheric absorption.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Atmospheric turbulence.</td>
</tr>
</tbody>
</table>
Application Results

We have implemented and tested the above design on the following

Implementation
• C# and VS 2012

Hardware
• Core 2 Duo T6600 processor at 2.20 GHz

Geometries
• Geometry 1 – 122 Triangles
• Geometry 2 – 72 Triangles
Application Results

Geometry 1 – 122 Triangles

Geometry 2 – 72 Triangles
# Application Results

Table 1: Results for Geometry 1 – 122 triangles

<table>
<thead>
<tr>
<th>Reflections Order</th>
<th>Diffractions Order</th>
<th>Paths Considered for Calculation</th>
<th>Time ms.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>8</td>
<td>362</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>8</td>
<td>3452</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>16</td>
<td>3636</td>
</tr>
</tbody>
</table>

Table 2: Results for Geometry 2 - 72

<table>
<thead>
<tr>
<th>Reflections Order</th>
<th>Diffractions Order</th>
<th>Paths Considered for Calculation</th>
<th>Time ms.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>8</td>
<td>355</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>8</td>
<td>1687</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>16</td>
<td>1755</td>
</tr>
</tbody>
</table>
PEMARD Framework in Commercial Applications
Framework Benefits

The benefits of our framework approach are the following

a) It outlines a pattern of a calculation process for acoustics simulations based on the principles of geometrical acoustics.

b) It provides an infrastructure for the acoustic simulation process by defining distinct steps and clear. It separates the concerns of the problem.

c) Enables research collaboration.
Q & A

• For more info, contact me at Panagiotis@pemard.com

Thank you!