A Framework for the Development of Accurate Acoustic Calculations for Games

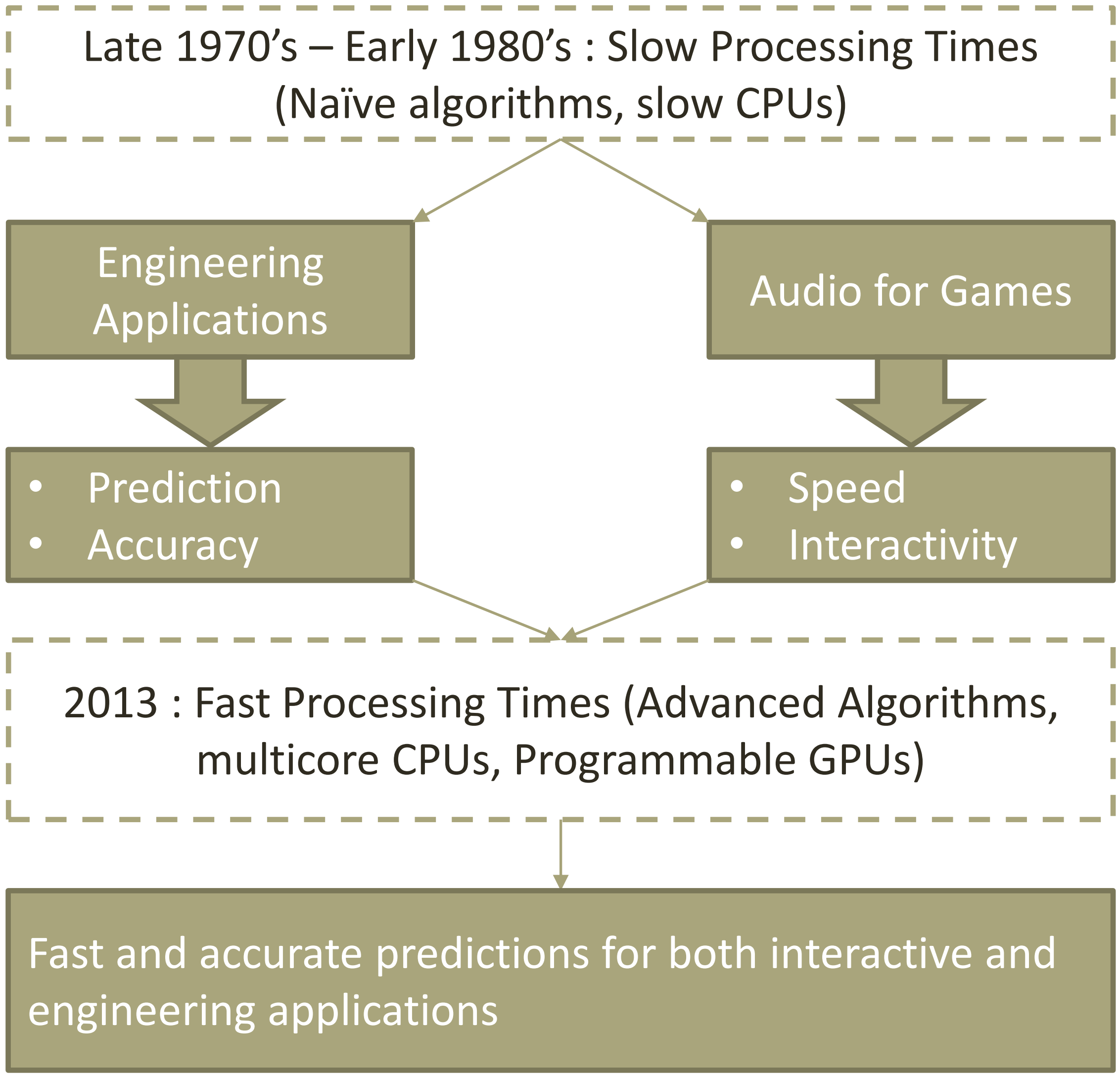
By

Panagiotis Charalampous &

Panos Economou

Mediterranean Acoustics Research & Development Ltd

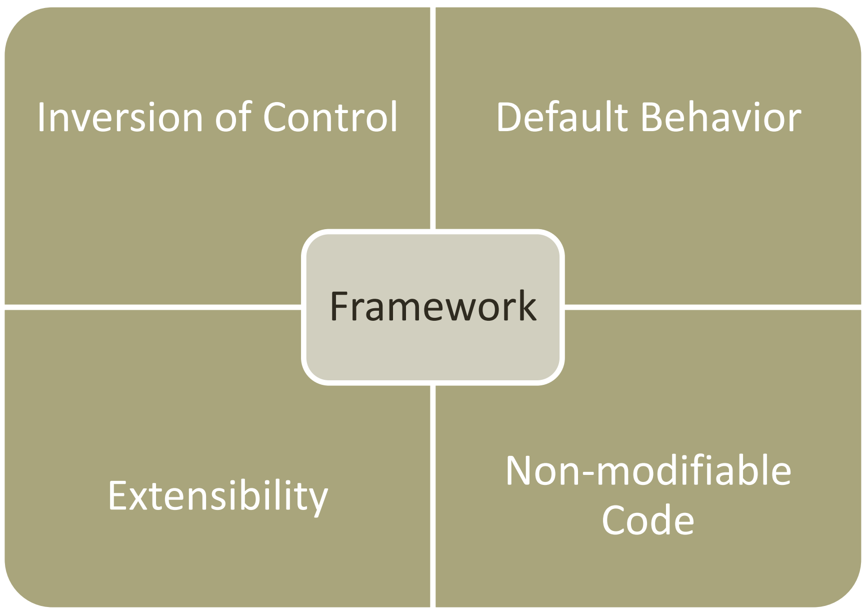
Historical Development




DIFFRACTION CALCULATIONS
DISTANCE ATTENUATION
MODEL OPTIMIZATION
SOUND PROPAGATION ALGORTHMS
PEMARD'S FRAMEWORK
REFLECTIONS
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



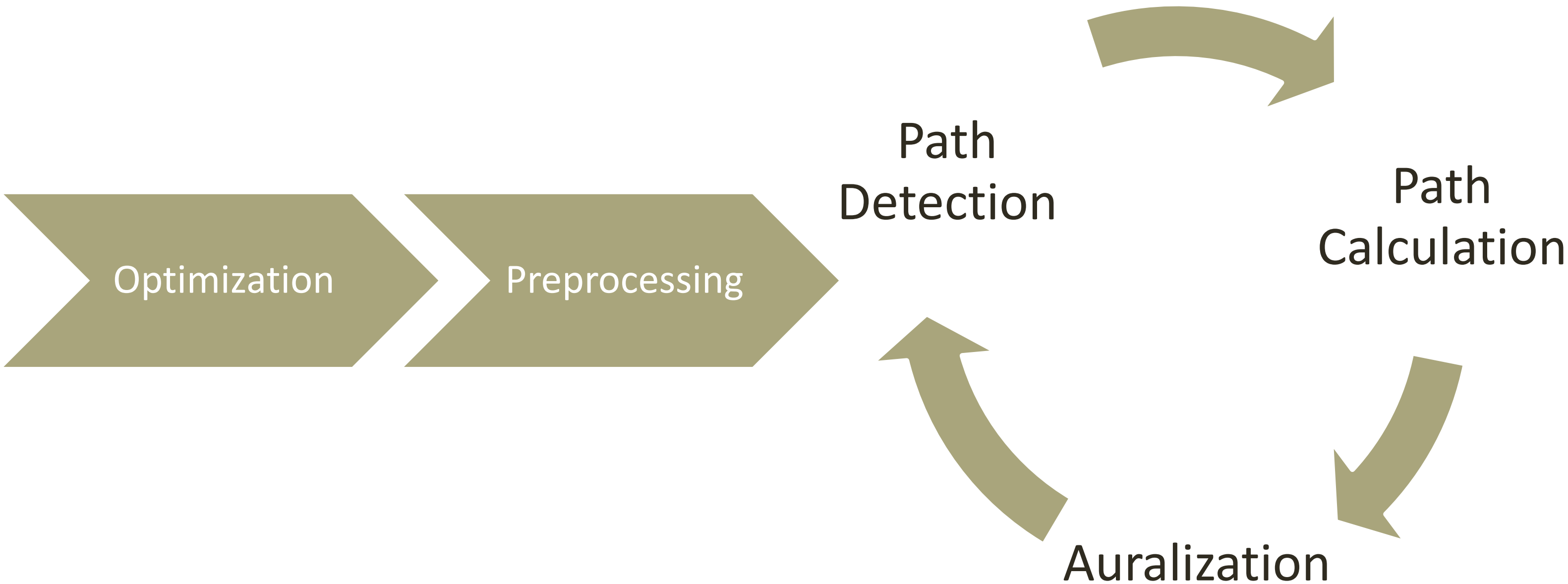
PEMARD Framework

- 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.

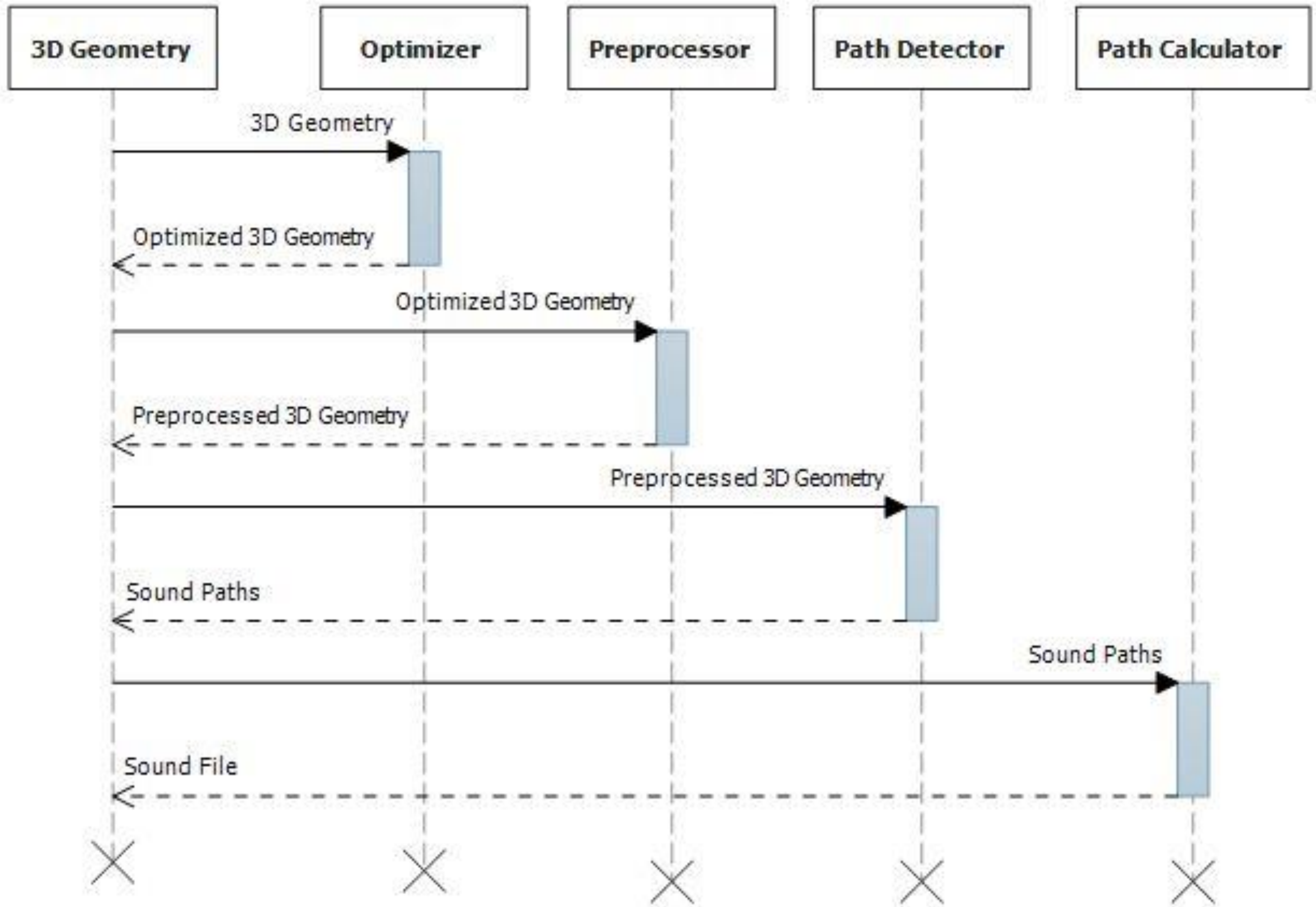
PEMARD Framework

- Geometrical Acoustics
- Loose Coupling
- Combination of Methodologies

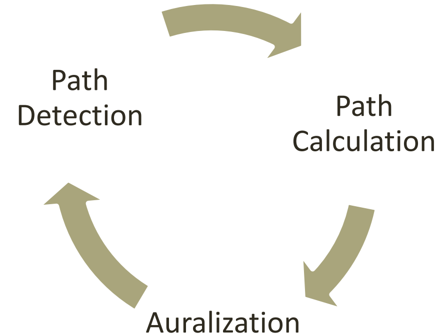
Sound Rendering Process



Sequence Diagram

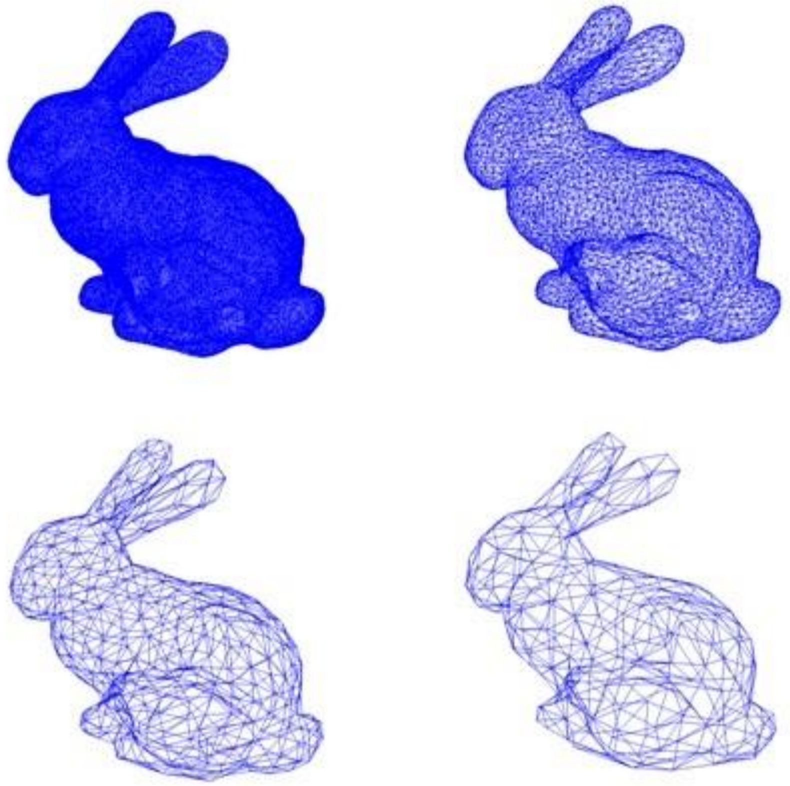


Optimization

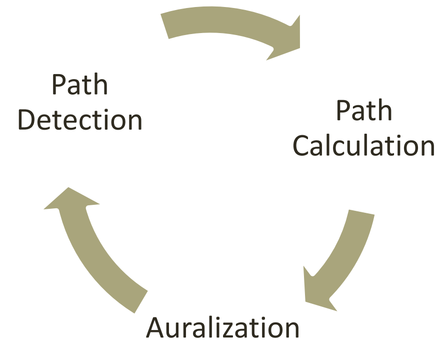


- Sound propagation algorithms performance is based on the model size.
- Most of accurate propagation algorithms have a complexity of $O(n^k)$
- 3D CAD or game models usually contain information relevant to graphics rendering which could be irrelevant to sound rendering.
- The optimization step is the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

Optimization

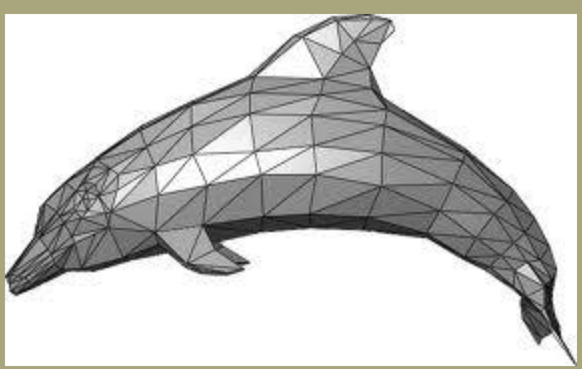


Preprocessing

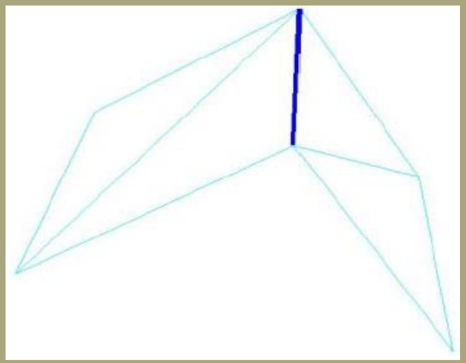


- Preprocessing is the step where we extract required metadata about the model

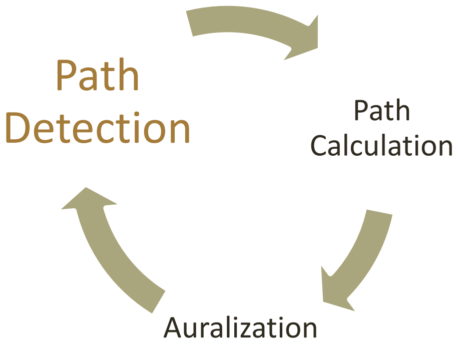
Raw Triangle Mesh



Triangles Forming an edge



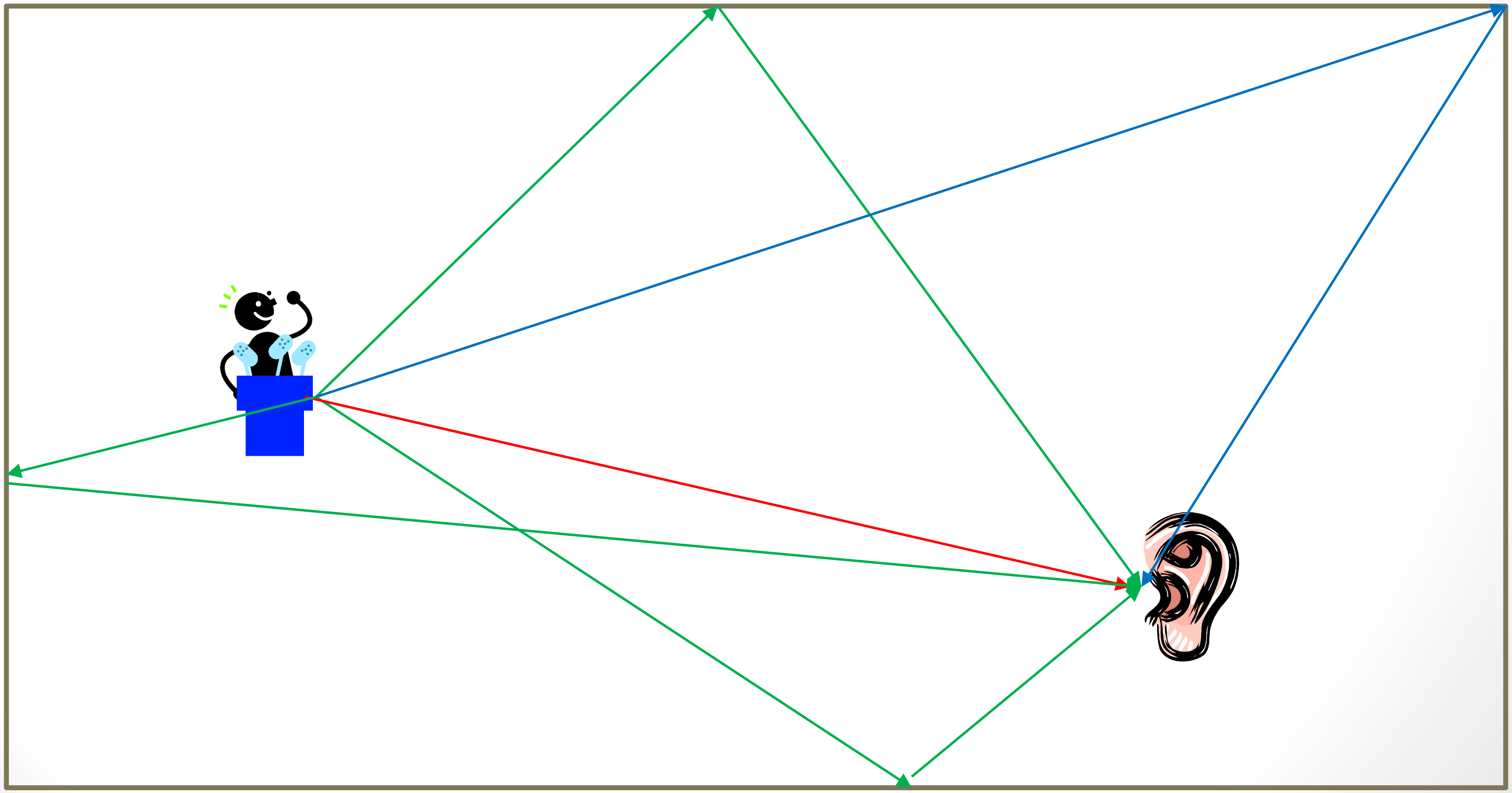
Sound Path Detection



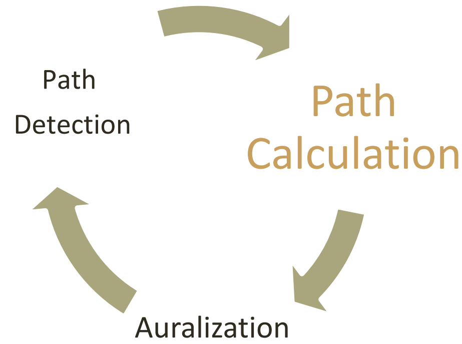
- Sound path detection refers to the process of finding the sound paths from source to receiver.

Sound Path Detection

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



Sound Path Calculation



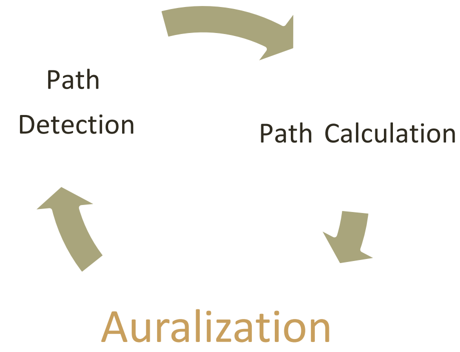
- Sound path calculation is the step where the contribution of each sound path and the total contribution at each source are calculated.
- We use the following expression

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

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



$$g(t) = \int_{-\infty}^{+\infty} s(\tau) f(t-\tau) \mathrm{d}\tau$$

Code Sample

```
// 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 HPSDiffractionCalculation());
// 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 responce at the receiver
    var ir = receiver.PreciseResults.IR;
}
```

Framework Application

pemard
mediterranean acoustics research & development

.

Optimization	Preprocessing	Path Detection	Path Calculation
• Unnecessary triangles removal	• Distinct edges determination • Edge to triangles association	• Reflections detection with visibility tracing • Diffractions detection • Reflection - diffraction detection	• Sound diffraction coefficients. • Spherical wave reflection coefficient. • Geometrical spreading. • Atmospheric absorption. • Atmospheric turbulence

Application Results

We have implemented and tested the above design on the following

Implementation

- C# and VS 2012

Hardware

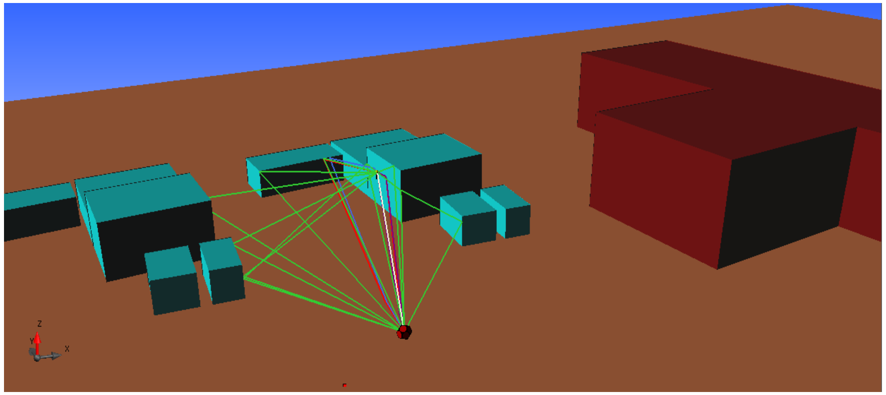
- Core 2 Duo T6600 processor at 2.20 GHz

Geometries

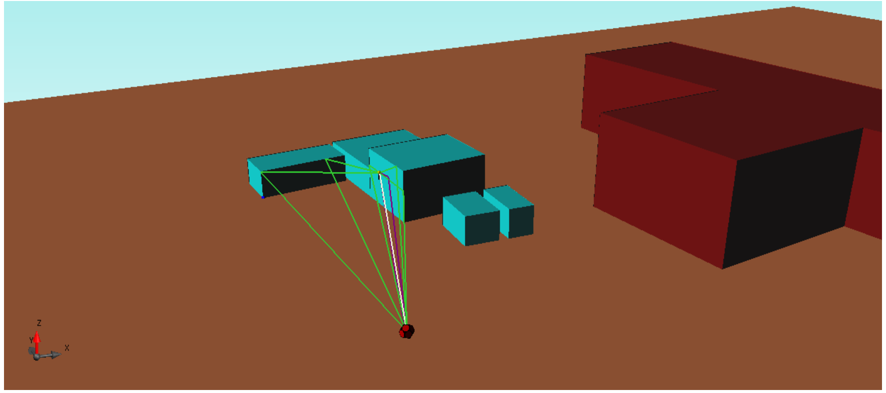
- Geometry 1 – 122 Triangles
- Geometry 2 – 72 Triangles

Application Results

Geometry 1 – 122 Triangles



Geometry 2 – 72 Triangles



Application Results

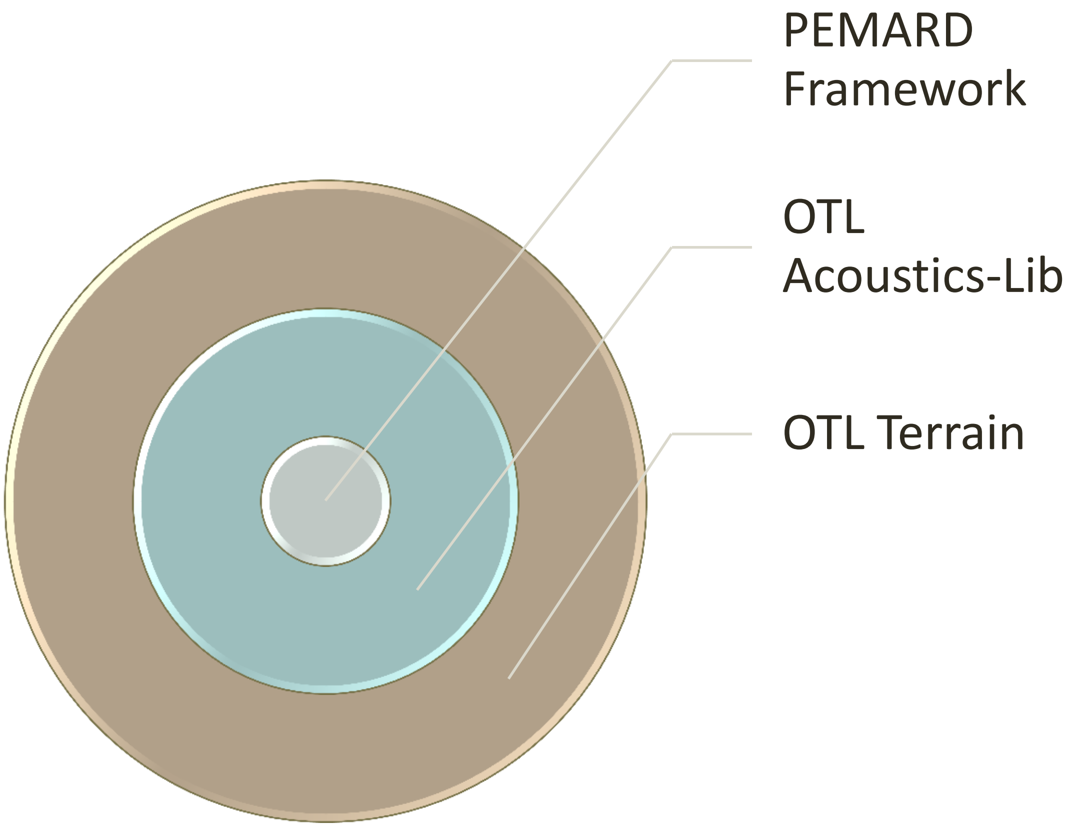
Table 1: Results for Geometry 1 – 122 triangles

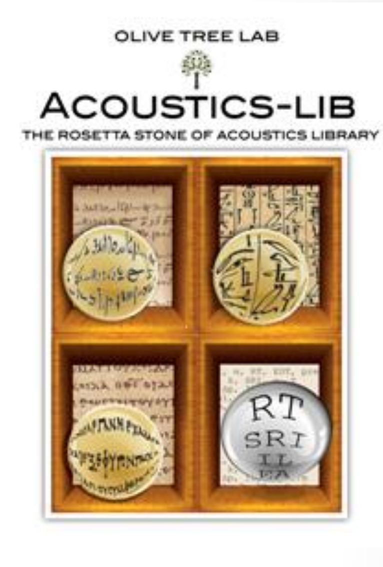
Reflections Order	Diffractions Order	Paths Considered for Calculation	Time ms.
1	1	8	362
2	2	8	3452
4	2	16	3636

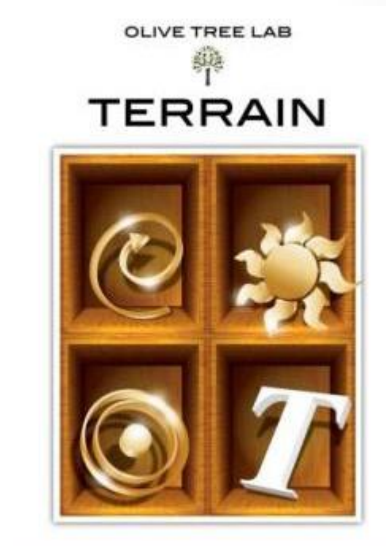
Table 2: Results for Geometry 2 - 72

Reflections Order	Diffractions Order	Paths Considered for Calculation	Time ms.
1	1	8	355
2	2	8	1687
4	2	16	1755

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.


DIFFRACTION CALCULATIONS
DISTANCE ATTENUATION
MODEL OPTIMIZATION
SOUND PROPAGATION ALGORTHMS
PEMARD'S FRAMEWORK
REFLECTIONS
FINITE IMPEDANCE
TIME DOMAIN CALCS
AURALIZATION

Q & A

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

Thank you!