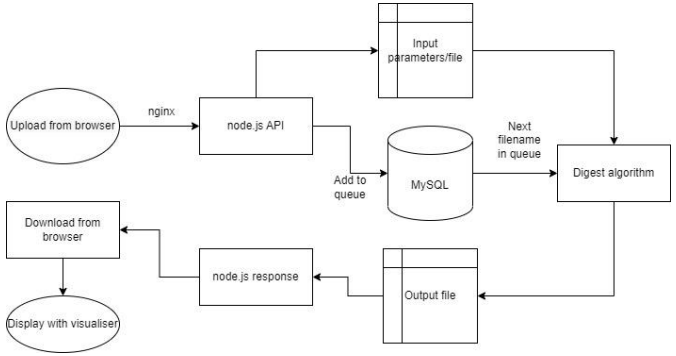
# Visualising Audio in an Enclosed Reverberant Space

Joe Davison – u1958945@unimail.hud.ac.uk

## A cloud processing system for calculating and visualising reverberation in an enclosed space.

- This system allows a user to upload a 16 bit stereo wav file and calculates the correlation between the original audio and the resulting audio at multiple points throughout room.
- The user can then download a file which contains the reverberated audio at a set listening point, as well as the correlation data for all points, which can then be played via an in browser visualiser.

### System overview




### SQL ticketing system

- Due to the length of time the calculations take, a queueing system is needed to sequence incoming requests. To do this the system uses a SQL database which tells both the digest algorithm and the node.js server who has a process order and whether it has been completed.

### Web-based

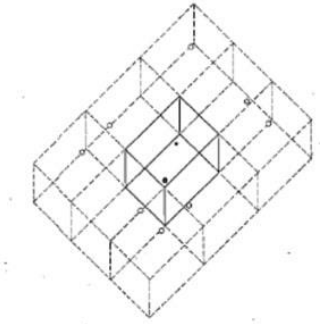
- The idea behind this project was to provide a simple and free service to audio engineers which allows them to get a greater understanding of a venue they might not have other equipment to measure.
- To provide that service I have chosen to host a public website using node.js, the website both serves as a place to upload audio files for processing as well as being the visualiser for playing files.

### Visualiser

- The visualiser represents the room as a net, each point in one of the cross sectional views is the average of the correlation at the points behind it going into the room.

### Simulation algorithm

- The system calculates reflections by getting delay times using the distance from the speaker to a virtual point representing the path the ray takes as a straight line going through the walls.
- It also calculates the total absorption of each wall by calculating the total count of and also which walls the ray passes through.



### GPU acceleration

- To accelerate the simulation process GPU processing is used to parallelise the calculations for each sample of each ray.
- Using GPU processing speeds up calculation for each point by about 30x-40x.
- The Nvidia Cuda compiler (nvcc) was used along with C++ to program the simulation algorithm.