# Optimal Listening Room Design

M. Queiroz  L. Dias

Department of Computer Science
University of São Paulo

X Brazilian Symposium on Computer Music, 2005

# Outline

# Acoustical Parameters

- Room Geometry
  here: cuboid room, given dimensions
- Surface materials
  here: given absorption coefficients

# Acoustical Parameters

- Room Geometry
  here: cuboid room, given dimensions
- Surface materials
  here: given absorption coefficients

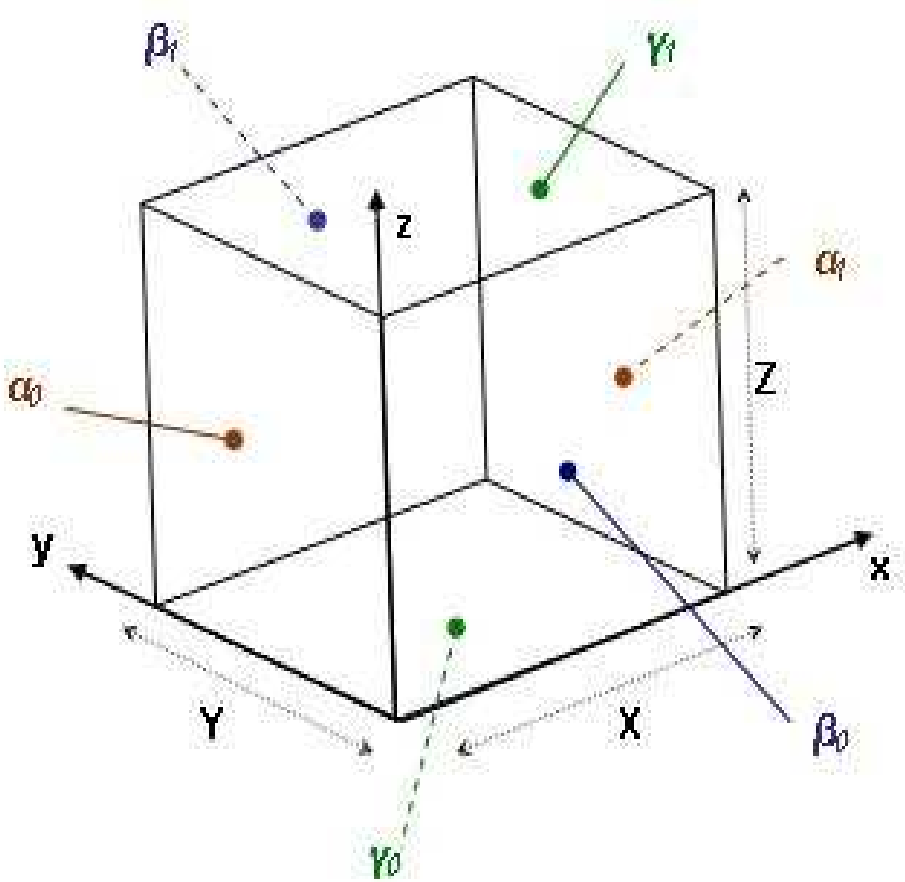
# Acoustical Parameters

- Room Geometry
  here: cuboid room, given dimensions
- Surface materials
  here: given absorption coefficients

# Acoustical Parameters

- Room Geometry
  here: cuboid room, given dimensions
- Surface materials
  here: given absorption coefficients

# Cuboid Room Model



# Acoustical Goals

- Define Sound Sources and Listener Positions
- Minimize Harmonic Distortion

# Acoustical Goals

- Define Sound Sources and Listener Positions
- Minimize Harmonic Distortion

# Geometrical Simulation

- Given Sound Source and Listener Positions
- Simulate Impulse Response
- Obtain Frequency Response
- Measure Harmonic Distortion

# Geometrical Simulation

- Given Sound Source and Listener Positions
- Simulate Impulse Response
- Obtain Frequency Response
- Measure Harmonic Distortion

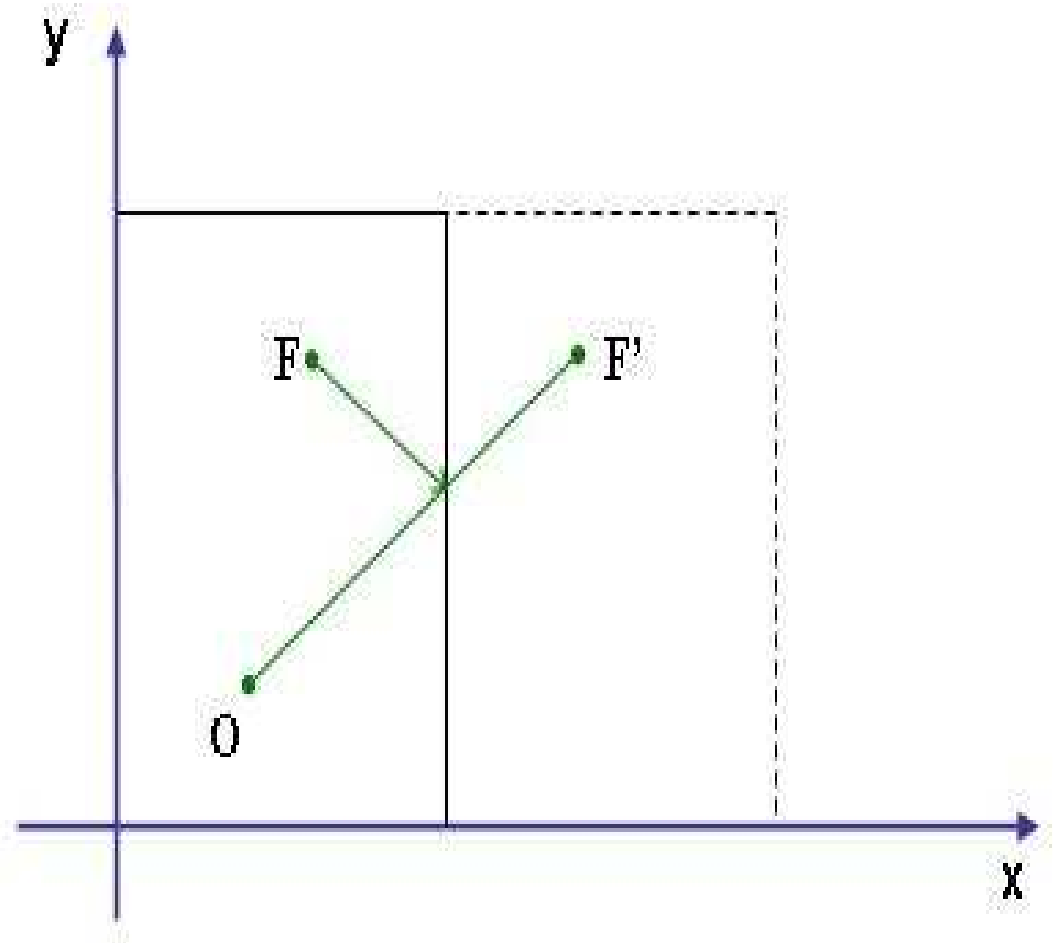
# Geometrical Simulation

- Given Sound Source and Listener Positions
- Simulate Impulse Response
- Obtain Frequency Response
- Measure Harmonic Distortion

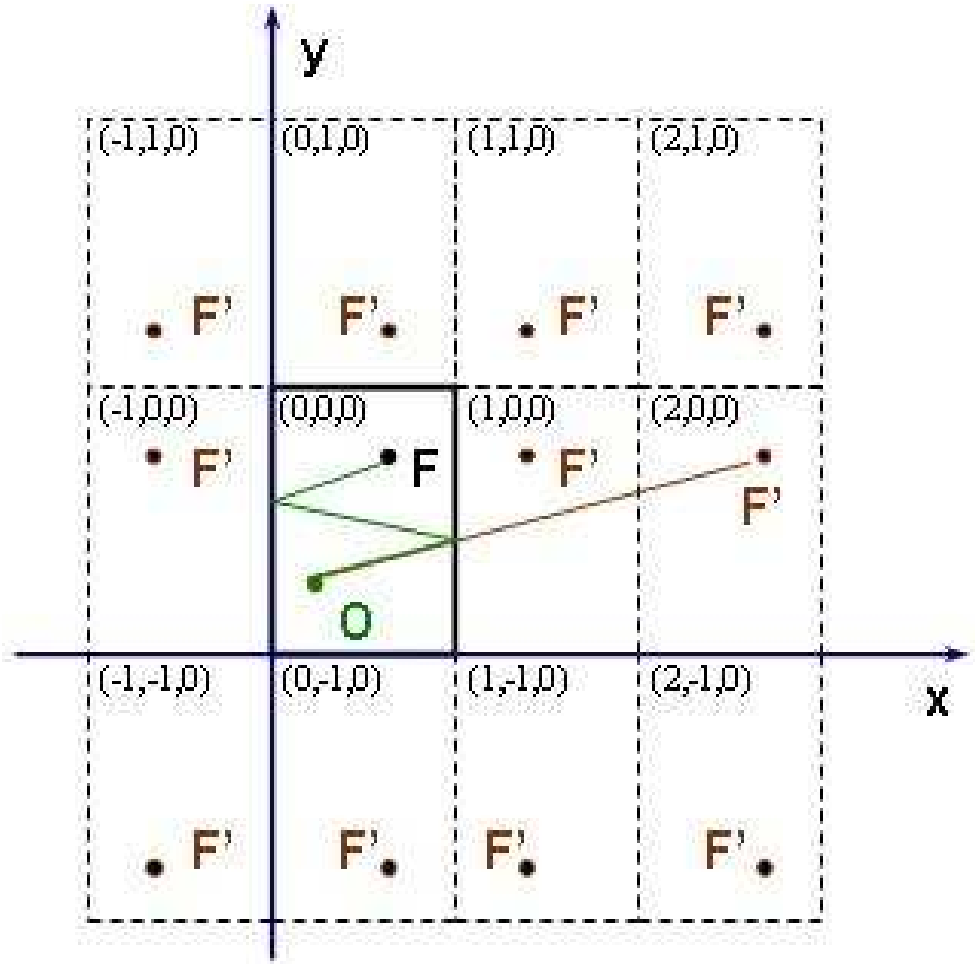
# Geometrical Simulation

- Given Sound Source and Listener Positions
- Simulate Impulse Response
- Obtain Frequency Response
- Measure Harmonic Distortion

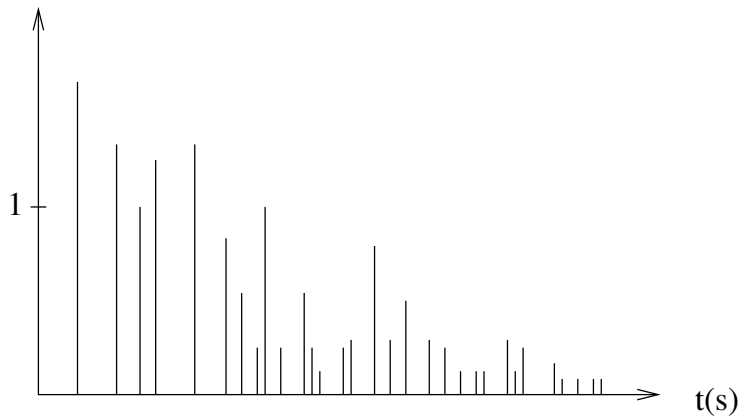
# Image Sound Source Model

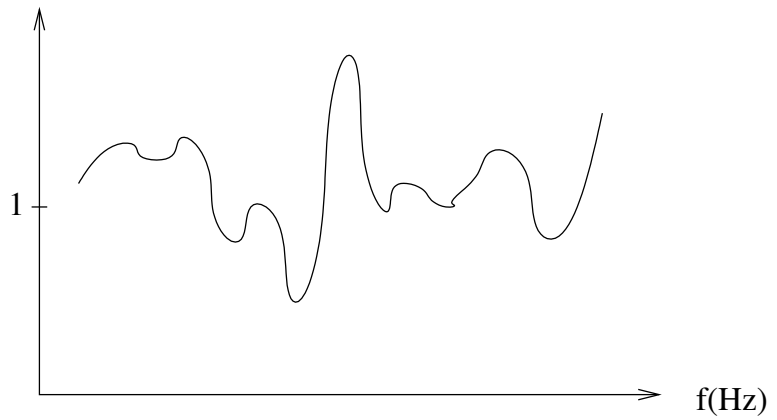


# Generating Impulse Response



# Obtaining Frequency Response by FFT



# Distortion as Deviation from Ideal Frequency Response

$$d(F) = \sqrt{\sum_{i=0}^{N-1} (\log(|F[i]|) - \mu)^2}$$

# Grid Definition

- Grid Points

$$G^{(m)} = \{x \in \mathbb{R}^n : x = x_o^{(m)} + h^{(m)} \sum_{i=1}^{n} \eta_i v_i\}$$

- Positive Bases

$$V_+ = \{V_1, \ldots, V_n, -V_1, \ldots, -V_n\}$$

$$V_+ = \{V_1, \ldots, V_n, -\sum_{i=1}^{n} V_i\}$$

# Grid Definition

- Grid Points

$$G^{(m)} = \{x \in \mathbb{R}^n : x = x_o^{(m)} + h^{(m)} \sum_{i=1}^{n} \eta_i v_i\}$$

- Positive Bases

$$V_+ = \{V_1, \ldots, V_n, -V_1, \ldots, -V_n\}$$

$$V_+ = \{V_1, \ldots, V_n, -\sum_{i=1}^{n} V_i\}$$

# Grid-Based Search

- Local Grid Minimizers

$$f(x + h^{(m)} v_i) \geq f(x), \forall v_i \in V_+$$

- While not in a Local Minimizer ...
  ... find descent diretion ...
  ... in-depth search for better solution
- When in a Local Minimizer ...
  ... make grid thinner ...
  ... check again condition for Local Minimizer

# Grid-Based Search

- Local Grid Minimizers

$$f(x + h^{(m)} v_i) \geq f(x), \forall v_i \in V_+$$

- While not in a Local Minimizer ...
  - ... find descent diretion ...
  - ... in-depth search for better solution
- When in a Local Minimizer ...
  - ... make grid thinner ...
  - ... check again condition for Local Minimizer

# Grid-Based Search

- Local Grid Minimizers

$$f(x + h^{(m)} v_i) \geq f(x), \forall v_i \in V_+$$

- While not in a Local Minimizer ...
  ... find descent diretion ...
  ... in-depth search for better solution
- When in a Local Minimizer ...
  ... make grid thinner ...
  ... check again condition for Local Minimizer

# Grid-Based Search

- Local Grid Minimizers

$$f(x + h^{(m)} v_i) \geq f(x), \forall v_i \in V_+$$

- While not in a Local Minimizer ...
  ... find descent diretion ...
  ... in-depth search for better solution
- When in a Local Minimizer ...
  ... make grid thinner ...
  ... check again condition for Local Minimizer

# Grid-Based Search

- Local Grid Minimizers

$$f(x + h^{(m)} v_i) \geq f(x), \forall v_i \in V_+$$

- While not in a Local Minimizer ...
  ... find descent diretion ...
  ... in-depth search for better solution
- When in a Local Minimizer ...
  ... make grid thinner ...
  ... check again condition for Local Minimizer

# Grid-Based Search

- Local Grid Minimizers

$$f(x + h^{(m)} v_i) \geq f(x), \forall v_i \in V_+$$

- While not in a Local Minimizer ...
  ... find descent diretion ...
  ... in-depth search for better solution
- When in a Local Minimizer ...
  ... make grid thinner ...
  ... check again condition for Local Minimizer

# Grid-Based Search

- Local Grid Minimizers

$$f(x + h^{(m)} v_i) \geq f(x), \forall v_i \in V_+$$

- While not in a Local Minimizer ...
  ... find descent diretion ...
  ... in-depth search for better solution
- When in a Local Minimizer ...
  ... make grid thinner ...
  ... check again condition for Local Minimizer

# Grid-Search Properties

- Stop Criteria
  successive goal function values
  minimum grid thinness (# iterations)
- Convergence to stationary points
- Allow reuse of previous simulations

# Grid-Search Properties

- Stop Criteria
  successive goal function values
  minimum grid thinness (# iterations)
- Convergence to stationary points
- Allow reuse of previous simulations

# Grid-Search Properties

- Stop Criteria
  successive goal function values
  minimum grid thinness (# iterations)

- Convergence to stationary points
- Allow reuse of previous simulations

# Grid-Search Properties

- Stop Criteria
  successive goal function values
  minimum grid thinness (# iterations)
- Convergence to stationary points
- Allow reuse of previous simulations

# Grid-Search Properties

- Stop Criteria
  successive goal function values
  minimum grid thinness (# iterations)
- Convergence to stationary points
- Allow reuse of previous simulations

# Going from Local to Global Minimizers

- Meta-heuristic to estimate attraction regions

- Generate random feasible points
- Uses local minimization procedure here: Grid-Search Method

# Going from Local to Global Minimizers

- Meta-heuristic to estimate attraction regions
- Generate random feasible points
- Uses local minimization procedure here: Grid-Search Method

# Going from Local to Global Minimizers

- Meta-heuristic to estimate attraction regions
- Generate random feasible points
- Uses local minimization procedure
  here: Grid-Search Method

# Going from Local to Global Minimizers

- Meta-heuristic to estimate attraction regions
- Generate random feasible points
- Uses local minimization procedure
  here: Grid-Search Method

# Constructing Clusters

- Find initial local minima
- Radii of clusters (2nd order approximation)

$$r_i(x) = \frac{1}{\pi}\left(i \cdot \Gamma\left(1+\frac{n}{2}\right) \cdot \sqrt{\det(H(x))} \cdot V \cdot \frac{\zeta \ln(N)}{N}\right)^{1/n}$$

- Dispose points within critical distance

# Constructing Clusters

- Find initial local minima
- Radii of clusters (2nd order approximation)

$$r_i(x) = \frac{1}{\pi}\left(i \cdot \Gamma\left(1+\frac{n}{2}\right) \cdot \sqrt{\det(H(x))} \cdot V \cdot \frac{\zeta \ln(N)}{N}\right)^{1/n}$$

- Dispose points within critical distance

# Constructing Clusters

- Find initial local minima
- Radii of clusters (2nd order approximation)

$$r_i(x) = \frac{1}{\pi}\left(i \cdot \Gamma\left(1+\frac{n}{2}\right) \cdot \sqrt{\det(H(x))} \cdot V \cdot \frac{\zeta \ln(N)}{N}\right)^{1/n}$$

- Dispose points within critical distance

# Symmetric Case: 2 sources, 1 listener

- Grid-search method
  Time: 87.509 s
  Best function value: 0.561700772947476 9
  Worst function value: 1.420227875343753 4
  Optimal Source location: (0.2109375, 3.5, 4.84375)
  Optimal Listener location: (1.5, 0.5, 0.0)
- Density Clustering Method
  Time: 1378.040 s
  Best function value: 0.576192159271777 6
  Worst function value: 0.983844720494565 2
  Optimal Source location: (1.49990844726562 5, 3.531616210937 5, 4.101257324218 75)
  Optimal Listener location: (1.5, 0.874877929687 5, 0.001525878906 25)

# Symmetric Case: 2 sources, 1 listener

- Grid-search method
  Time: 87.509 s
  Best function value: 0.561700772947476 9
  Worst function value: 1.420227875343753 4
  Optimal Source location: (0.2109375, 3.5, 4.84375)
  Optimal Listener location: (1.5, 0.5, 0.0)
- Density Clustering Method
  Time: 1378.040 s
  Best function value: 0.576192159271777 6
  Worst function value: 0.983844720494565 2
  Optimal Source location: (1.49990844726562 5, 3.5316162109375, 4.10125732421875)
  Optimal Listener location: (1.5, 0.874877929687 5, 0.00152587890625)

# Assymetric Case: 1 source, 1 listener

- Grid-search method
  Time: 61.775 s
  Best function value: 0.395826985767801 7
  Worst function value: 0.779715886901801 9
  Optimal Source location: (2.25, 0.0, 0.0)
  Optimal Listener location: (2.25, 2.0, 2.5)
- Density Clustering Method
  Time: 850.816 s
  Best function value: 0.365890521997611 13
  Worst function value: 1.797693134862315 7308
  Optimal Source location: (1.6875, 1.75, 2.5)
  Optimal Listener location: (1.6875, 1.75, 2.490234375)

# Assymetric Case: 1 source, 1 listener

- Grid-search method
  Time: 61.775 s
  Best function value: 0.395826985767801 7
  Worst function value: 0.779715886901801 9
  Optimal Source location: (2.25, 0.0, 0.0)
  Optimal Listener location: (2.25, 2.0, 2.5)
- Density Clustering Method
  Time: 850.816 s
  Best function value: 0.365890521997611 13
  Worst function value: 1.797693134862315 7308
  Optimal Source location: (1.6875, 1.75, 2.5)
  Optimal Listener location: (1.6875, 1.75, 2.490234375)

# Future Work

- Comparison among positive bases
  trade-off between freedom of movement and computational cost
- Acoustic simulation
  frequency-dependent absorption coefficients
  general geometries
  diffusion, difraction, ...
- Implementation within the ACMUS toolbox
  (Eclipse computational platform)

# Future Work

- Comparison among positive bases
  trade-off between freedom of movement and computational cost
- Acoustic simulation
  frequency-dependent absorption coefficients
  general geometries
  diffusion, difraction, ...
- Implementation within the ACMUS toolbox (Eclipse computational platform)

# Future Work

- Comparison among positive bases
  trade-off between freedom of movement and computational cost
- Acoustic simulation
  frequency-dependent absorption coefficients
  general geometries
  diffusion, difraction, ...
- Implementation within the ACMUS toolbox
  (Eclipse computational platform)

# Thanks

Thanks for your attention!