Localização de Fontes e Ouvintes em Salas de Escuta*

Luciana Dias[1], **Marcelo Queiroz**[1]

[1]Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

Abstract. *This work consists in the study and implementation of geometrical acoustics and optimization techniques for the automatic search of an optimal placement of sound sources and listeners in listening rooms, aiming at a minimal harmonic distortion. The implementation is part of the ACMUS toolkit which provides computational tools for measurement of acoustical parameters as well as other simulation packages. Original contributions are the use of global optimization strategies and the release of open-source implementations.*

Resumo. *Este trabalho consiste no estudo e implementação de técnicas de acústica geométrica e otimização para a busca automática de localizações de fontes e ouvinte em salas de escuta, visando obter o mínimo de distorção harmônica possível. Esta implementação faz parte do conjunto de ferramentas de simulação do projeto ACMUS, que possui também ferramentas para a medição de parâmetros acústicos e outras ferramentas de simulação. São contribuições originais o uso de estratégias de otimização global e a disponibilização de implementações em código-aberto.*

1. Introdução

Uma *sala de escuta* é um ambiente caracterizado por possuir superfícies com diferentes características de reflexão e absorção do som, sendo portando um recinto *reverberante*. Quando uma fonte sonora é situada em um ambiente como esse, o volume de ar encerrado entre as superfícies limites da sala é excitado não somente pelas ondas sonoras provenientes diretamente da fonte, mas também pelas ondas que são refletidas pelas superfícies [Beranek, 1993, López, 2001].

A forma mais apropriada para obtenção de parâmetros acústicos de uma sala é a execução de medições no próprio local; quando isto não é possível, pode-se construir uma réplica em miniatura e utilizar as medições na réplica para obter aproximações dos parâmetros referentes ao ambiente original. A utilização de réplicas é viável quando não se pretende fazer modificações na sala de escuta, ou quando o número de modificações é muito pequeno. Ao considerar o problema de *projeto* de salas de escuta observa-se a dificuldade em utilizar medições no local ou em réplicas, pois durante a fase de projeto se deseja testar uma enorme quantidade de combinações dos materiais e de posições de fontes ou ouvintes, ou mesmo as dimensões da sala, entre outros parâmetros.

*Projeto financiado pela FAPESP, bolsa 04/01184-9 e projeto temático 02/02678-0.

Para simular computacionalmente o comportamento acústico de uma sala e quantizar o nível de distorção harmônica na posição de escuta, é necessário obter uma aproximação de sua *resposta em freqüência* a partir de uma quantidade mínima de informações que correspondem ao modelo da sala. A resposta em freqüência está associada à distorção harmônica pois quantifica a atenuação ou amplificação sofrida por cada freqüência, para uma certa disposição de fonte sonora e ouvinte. Essas diferenças no comportamento de cada freqüência são conseqüência da ressonância do recinto, das diferentes velocidades de amortecimento dos sinais e dos modos normais de vibração. Dois dos principais modelos geométricos teóricos empregados para a simulação acústica computacional são o *modelo de traçado de raios* e o *modelo de fontes virtuais* [Rindel, 1997, Rindel, 2000, López, 2001]. Estes modelos permitem a obtenção de forma aproximada da resposta impulsiva da sala, que é convertida em resposta em freqüência através da transformada rápida de Fourier (FFT). Tais simulações computacionais são imprescindíveis para uma busca automática de localizações ótimas para fontes sonoras e ouvinte.

Uma das alternativas para efetuar uma busca automática de localizações ótimas de fonte sonora e ouvinte é utilizar técnicas de otimização [Warusfel, 1995, D'Antonio and Cox, 1997, Queiroz, 2003]. Para tal, é necessário definir matematicamente qual é o objetivo de tal processo de busca. Neste trabalho considera-se a minimização da distorção harmônica, que pode ser expressa através de uma medida de distância entre a resposta em freqüência simulada e uma resposta em freqüência ideal; tal resposta ideal poderia ser uma resposta plana ou outra função pré-definida. A cada passo do algoritmo de busca, calcula-se através de uma simulação por acústica geométrica a resposta em freqüência associada a uma certa localização da fonte sonora e do ouvinte e se obtém a medida de distorção correspondente, que é a função objetivo a ser minimizada. Ao final do algoritmo são produzidas as localizações mais adequadas para o posicionamento da fonte sonora e ouvinte, isto é, as posições que minimizam a distorção da resposta em freqüência dentre as posições visitadas. O uso de técnicas de otimização global, não mencionadas em trabalhos relacionados, é de grande relevância pois a medida de distorção em função da localização das fontes e ouvinte possui muitos mínimos locais com valores bastante diversos.

É objetivo deste trabalho contribuir com as pesquisas do projeto *ACMUS*, que visa o desenvolvimento de um software para cálculo, análise e simulação de acústica de salas para prática musical [F. H. Iazzetta and Silva, 2001, AcMus, 2005]. Uma das fortes contribuições deste projeto consiste em disponibilizar os códigos-fonte de todas as ferramentas implementadas. Utilizou-se a linguagem *Java*, por sua robustez, versatilidade e portabilidade. As seções a seguir detalham as técnicas de acústica geométrica e otimização global utilizadas na implementação, e apresentam os resultados obtidos.

2. Modelo de Acústica Geométrica

O Modelo de Fontes Virtuais

Segundo o modelo de fontes virtuais, o sinal da fonte sonora refletido por uma parede é idêntico ao de uma onda direta criada por uma fonte virtual situada do outro lado

da parede, como uma imagem especular da fonte sonora real. Ou seja, de acordo com este modelo, as fontes virtuais irradiam ondas sonoras sincronizadas com a fonte real, e apenas caminhos diretos entre fontes e ouvinte são considerados. Seguindo o mesmo processo para todas as fontes virtuais nas salas virtuais adjacentes, pode-se encontrar todas as reflexões de primeira ordem que passam por um determinado ponto do ambiente. Esse processo se repete para reflexões de segunda ordem, terceira, etc.

Este modelo é aplicado a ambientes de superfícies planas e usa o modelo de reflexões especulares, que é o caso dos ambientes tratados neste projeto, e é mais indicado pra salas de geometrias simples, como as de forma cubóide, já que para geometrias mais complexas é muito difícil identificar as posições das fontes virtuais. Modelos geométricos que tratam a propagação do som por raios são bastante acurados em altas freqüências, mas sua qualidade cai consideravelmente nas freqüências mais baixas. Por outro lado estes modelos têm como principal vantagem a sua eficiência computacional, visto que toda informação processada diz respeito à localização do ouvinte e às reflexões que atingem aquele ponto. Outras técnicas de simulação, como a solução da equação de onda por elementos finitos, são computacionalmente inviáveis por necessitarem da informação da variação de pressão em uma grande quantidade de pontos (que fornecem uma aproximação de todo o espaço da sala).

Na modelagem para o problema em questão, cada fonte virtual está localizada em uma sala virtual, identificada por um índice (i, j, k), que indica a distância da sala virtual em relação à sala real em número de salas, sobre um eixo tridimensional. Por exemplo, a sala virtual imediatamente à direita da sala real na direção do eixo x possui índice $(1, 0, 0)$. Para efeito de cálculo computacional, considera-se apenas as salas virtuais situadas a uma distância menor do que um certo raio máximo da sala real, onde a medida de distância é dada por $|i| + |j| + |k|$.

Obtenção da Resposta Impulsiva

A resposta impulsiva de um ambiente de escuta musical corresponde à resposta produzida por ele para um impulso produzido pela fonte sonora. De acordo com o modelo geométrico adotado e considerando-se a simulação de um pulso ideal (delta de Dirac), a resposta impulsiva fornece um *mapa temporal* das reflexões, registrando o intante e a intensidade com que cada reflexão chega ao ponto onde está situado o ouvinte.

Para obter o *retardo* com que a reflexão da sala de índice (i, j, k) chega ao ouvinte, calcula-se a distância da fonte sonora real (x_F, y_F, z_F) à fonte sonora virtual

$$\left(2X\lceil i/2\rceil + (-1)^{|i|}x_F \ ,\ 2Y\lceil j/2\rceil + (-1)^{|j|}y_F \ ,\ 2Z\lceil k/2\rceil + (-1)^{|k|}z_F\right)$$

onde (X, Y, Z) são as dimensões da sala. A partir da posição de cada fonte virtual e da posição do ouvinte, é fácil calcular o tempo gasto pelas reflexões até atingir o ouvinte, dividindo-se a distância percorrida por cada reflexão pela velocidade do som no ar.

Para calcular a absorção sofrida pelo sinal sonoro saído da fonte localizada na sala virtual de índice (i, j, k) deve-se considerar todos os coeficientes de absorção das superfícies intersectadas pelo raio que liga a fonte virtual ao ouvinte. Se os

coeficientes das superfícies são $\alpha_0, \alpha_1, \beta_0, \beta_1, \gamma_0, \gamma_1$ então a absorção total devido às superfícies será

$$\alpha_0^{||\lfloor i/2 \rfloor||} * \alpha_1^{||\lceil i/2 \rceil||} * \beta_0^{||\lfloor j/2 \rfloor||} * \beta_1^{||\lceil j/2 \rceil||} * \gamma_0^{||\lfloor k/2 \rfloor||} * \gamma_1^{||\lceil k/2 \rceil||}$$

onde os expoentes indicam justamente quantas vezes o sinal proveniente da sala (i, j, k) é refletido em cada uma das superfícies. Além disto há também a atenuação devido à distância percorrida d, que corresponde a um fator de atenuação $\mathcal{O}(1/d)$.

Obtenção da Resposta em Freqüência

A resposta em freqüência de uma sala fornece uma quantificação da atenuação ou amplificação sofrida por cada freqüência, para uma certa localização de fonte sonora e ouvinte. Através da Transformada de Fourier, pode-se obter a resposta em freqüência através do espectro da resposta impulsiva. Este espectro é calculado através do algoritmo conhecido como *Transformada Rápida de Fourier (FFT)*, que permite obter N amostras da resposta em freqüência em tempo $\mathcal{O}(N \log N)$.

Estimativa de Distorção na Resposta em Freqüência

Tendo a resposta em freqüência amostrada para uma certa posição de fonte sonora e ouvinte, pode-se comparar a quantidade de distorção presente nessa resposta em relação a uma resposta em freqüência "neutra", como por exemplo a *resposta plana*, ou outro modelo adequado.

Uma resposta plana corresponde à situação hipotética em que nenhuma freqüência sofreria qualquer atenuação ou amplificação, a não ser aquela devida à absorção do ar. Esta situação hipotética só pode ser obtida num ambiente sem superfícies refletoras (ao ar livre ou em um ambiente com 100% de absorção), e não é sequer uma situação ideal, visto que a reverberação é de fato desejável em um ambiente de escuta musical.

Por outro lado a resposta plana fornece um bom parâmetro de comparação entre duas duas respostas em freqüência "reais", pois aquela que resultar em maior distorção estará também mais distante da resposta plana.

A distorção é calculada a partir da resposta logarítmica em freqüência, pois esta corresponde melhor à percepção de distorção: intuitivamente, pode-se pensar que um fator s (uma amplificação de s vezes) e um fator $1/s$ (uma atenuação de s vezes) são percebidos como distorções igualmente "ruins". Para uma dada resposta logarítmica em freqüência armazenada em um vetor F com N amostras, define-se a medida de distorção pela expressão

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

onde $\mu = \frac{1}{N} \sum_{i=0}^{N-1} F[i]$ é a média dos fatores de atenuação/amplificação, e normalmente está próximo de zero (que é o logarítmo do valor 1, que corresponde a nenhuma atenuação ou amplificação). Observe que esta expressão corresponde à distância entre o gráfico de F e o gráfico de resposta constante e igual a μ, utilizando a norma Euclidiana.

3. Técnicas de Otimização Global

Em geral um problema de otimização global corresponde a minimizar uma função $f(x)$ (chamada de função objetivo) onde $x \in S$ e S é o conjunto de soluções viáveis. Muitos problemas de otimização contínua possuem várias soluções ótimas locais. Para esses casos, principalmente quando as condições do escopo clássico de otimização contínua (diferenciabilidade e convexidade) não são satisfeitas ou verificáveis, usa-se estratégias de otimização global. Em princípio, o objetivo é encontrar um ótimo global, ou seja, um $x^* \in S$ tal que $f(x^*) \leq f(x), \forall x \in S$. Quando S é finito, poder-se-ia fazer uma busca exaustiva, embora isso seja computacionalmente inviável na maioria dos casos. Quando S não é finito, em geral são utilizados métodos iterativos, ou seja, métodos que geram uma seqüência de soluções que aproximam-se da solução ótima.

Métodos de otimização global podem ser classificados como *determinísticos* ou *estocásticos*, bem como *heurísticos* ou *exatos*. Nos métodos determinísticos, a partir dos mesmos dados iniciais, chega-se à mesma solução; já os métodos estocásticos possuem, em geral, algum componente de busca global aleatória, mas podem e normalmente são utilizados em conjunto com estratégias determinísticas. A distinção entre métodos exatos e heurísticos tem relação com a existência de garantias teóricas de convergência a uma solução global. Poucos problemas de otimização global são suficientemente bem estruturados a ponto de admitirem um método de solução exata; muitos problemas reais e relevantes só podem ser resolvidos aproximadamente.

Uma característica fundamental do problema, no que concerne a escolha de um algoritmo específico, é o alto custo computacional envolvido no cálculo da função objetivo, que depende da simulação geométrica e do cálculo da FFT; deve-se, assim, garantir que o algoritmo não calcule a função objetivo em um número excessivamente grande de pontos, ou de forma desnecessária.

Para comparar duas abordagens heurísticas para o problema, foram implementados o método estocástico *Density Clustering* [Horst and Pardalos, 1995] e o método determinístico baseado em grades [Coope and Price, 2001].

Método *Density Clustering*

Este é um método em duas fases: a primeira fase (fase global) gera uma amostragem aleatória de pontos viáveis usando a distribuição uniforme sobre o conjunto viável, pontos estes que são avaliados de acordo com a função objetivo; a segunda fase (fase local) aplica um procedimento de otimização local L a partir de um subconjunto dessa amostragem, obtendo vários mínimos locais, dos quais o melhor é selecionado.

Duas técnicas para o pré-processamento da amostra são *Redução*, que consiste em tomar uma fração da amostra apenas com os pontos com melhor valor da função objetivo, e *Concentração*, que consiste em modificar a amostra aplicando alguns passos de Cauchy para cada ponto. Antes de aplicar L a cada ponto, são criados grupos (*clusters*) com pontos relativamente próximos, e aplica-se L apenas uma vez para cada grupo. O método básico para identificação de clusters é tomar o melhor ponto da amostra e adicionar outros pontos usando uma *regra de clustering*.

Uma destas regras baseia-se na hipótese de que a região de atração de um mínimo local pode ser aproximada por uma elipsóide, isto é, a função objetivo pode ser localmente aproximada por uma função quadrática. O algoritmo consiste basicamente no seguinte:

0 $k \leftarrow 1$, $X^* \leftarrow \emptyset$, $i \leftarrow 1$, $j \leftarrow 0$

1 Gere N pontos usando uma distribuição uniforme sobre o conjunto viável S e selecione os γkN melhores pontos, para um dado $\gamma \in (0,1)$ (método da redução).

2 Se todos os pontos selecionados estão em algum *cluster*, vá para o passo *4*.
Se $j \leq |X^*|$, escolha o j-ésimo mínimo local $x_* \in X^*$ como o próximo "ponto semente" e vá para o passo *3*.
Se $j > |X^*|$, escolha o melhor ponto $\overline{x}$ que não estiver em nenhum *cluster* e aplique o método de otimização local L a partir de $\overline{x}$. Se o mínimo local resultante x_* for um elemento de X^*, inclua $\overline{x}$ no *cluster* de x_* e volte para o passo *2*. Se $x_* \notin X^*$, adicione x_* a X^* e faça x_* ser o próximo "ponto semente".

3 Adicione todos os pontos sem *cluster* da amostra reduzida que estejam dentro de uma distância $r_i(x_*)$ do ponto semente ao *cluster* iniciado por x_*. Se nenhum ponto tiver sido adicionado, faça $j \leftarrow j+1$ e volte para o passo *2*; senão, faça $i \leftarrow i+1$ e volte para o passo *3*.

4 Faça $k \leftarrow k+1$ e volte para o passo *1*.

A distância crítica $r_i(.)$ sugerida em [Rinnoykan and Timmer, 1987] é dada pela equação

$$r_i(x) = \frac{1}{\pi}\left(i\Gamma\left(1+\frac{d}{2}\right)\cdot\sqrt{-detH(x)}\cdot m(S)\cdot\frac{\zeta ln(kN)}{kN}\right)^{1/d}$$

onde d é a dimensão do problema, $m(S)$ é a medida de Lebesgue, $H(x)$ é a Hessiana no ponto x, $\zeta > 0$ é um parâmetro positivo do algoritmo e $\Gamma(\cdot) = \int_{t=0}^{\infty} t^{z-1}e^{-t}dt$.

Método baseado em grades

O método de otimização baseado em grades é um método determinístico, que busca um minimizador da função objetivo examinando-a em uma seqüência de grades sucessivamente refinadas. Cada grade $G^{(m)}$ é definida por um conjunto de n vetores-base linearmente independentes $V^{(m)}$ onde $V^{(m)} = \{v_i^{(m)} \in \mathbb{R}^n : i = 1,\ldots,n\}$. Os pontos da grade $G^{(m)}$ são $G^{(m)} = \{x \in \mathbb{R}^n : x = x_o^{(m)} + h^{(m)}\sum_{i=1}^{n}\eta_i v_i^{(m)}\}$ onde η é um inteiro e $h^{(m)}$ é um número positivo que representa a espessura da malha. Este parâmetro é ajustado a medida que m cresce para assegurar o refinamento das malhas. A base $V^{(m)}$ é usada para formar uma base positiva $V_+^{(m)}$ tal que todo vetor em $\mathbb{R}^n$ é combinação linear positiva dos vetores em V_+ e nenhum elemento de V_+ é combinação linear dos outros. Se $V^{(m)}$ é a base canônica, pode-se construir uma base positiva fazendo $V_+^{(m)} = \{V_1^{(m)},\ldots,V_n^{(m)},-\sum_{i=1}^{n}V_i^{(m)}\}$; outra possibilidade é fazer $V_+^{(m)} = \{V_1^{(m)},\ldots,V_n^{(m)},-V_1^{(m)},\ldots,-V_n^{(m)}\}$.

Por definição, tem-se que $x \in G^{(m)}$ é um *minimizador local de grades* com respeito à base positiva $V_+^{(m)}$ se e somente se $f(x + h^{(m)}v_i) \geq f(x), \forall v_i \in V_+^{(m)}$. Se x não é um *minimizador local de grades* então existe uma direção de descida $s^{(k)} \in V_+^{(m)}$, isto é, uma direção que satisfaz $f(x^{(k)} + h^{(m)}s^{(k)}) < f(x^{(k)}$. Como esta direção promove uma melhoria no valor da função objetivo, é interessante continuar explorando esta direção através de uma busca em profundidade: a função objetivo f é avaliada na seqüência de pontos $x^{(k)} + \alpha_i h^{(m)}s^{(k)}$, onde $\{\alpha_i\}$ é uma seqüência crescente de inteiros, enquanto $f(x^{(k)} + \alpha_{i+1}h^{(m)}s^{(k)}) \leq f(x^{(k)} + \alpha_i h^{(m)}s^{(k)})$. A busca termina na primeira violação desta última condição.

Algoritmo de Minimização em Grades

0 $m \leftarrow 1$, $k \leftarrow 1$ e $x^{(1)} \in G^{(m)}$.
1 Enquanto as condições de parada forem falsas,
- **1.A** Faça $i \leftarrow 1$ e $r \leftarrow 0$.
- **1.B** Enquanto $x^{(k)}$ não for um minimizador local da grade,
  - Calcule uma direção de descida $s^{(k)} \in V_+^{(m)}$; se não houver direção de descida, vá para o passo [1.C].
  - Escolha uma seqüência crescente de inteiros $\alpha_0 = 1 < \alpha_1 < \alpha_2 \ldots$ e determine o menor índice l que satisfaz $f(x^{(k)} + \alpha_{l+1}h^{(m)}s^{(k)}) \geq f(x^{(k)} + \alpha_l h^{(m)}s^{(k)})$
  - Faça $x^{(k+1)} \leftarrow x^{(k)} + \alpha_l h^{(m)}s^{(k)}$.
  - Faça $k \leftarrow k + 1$.
- **1.C** Faça $m \leftarrow m + 1$.

As condições de parada envolvem algum teste para detectar convergência dos iterados, bem como limitações no tempo de execução ou número de iterações. É possível provar que, sob certas hipóteses técnicas, o algoritmo acima gera uma seqüência de pontos que converge para um ponto estacionário de f. Para mais detalhes, veja o artigo [Coope and Price, 2001].

4. Implementação e Resultados

Resposta Impulsiva

O cálculo da resposta impulsiva da sala é feito no método *generateImpulseResponse*, que se encontra no arquivo *ImpulseResponse.java*. Seus dados de entrada consistem nas características de uma sala de escuta cubóide (cuja descrição é dada pelas dimensões e coeficientes de absorção de cada uma de suas superfícies) e posições fixadas para a fonte sonora e o ouvinte. O número máximo de fontes virtuais gerado é também passado como parâmetro, e é representado por um "raio máximo" de salas em torno da sala real. Em outras palavras, para um raio máximo igual a R, serão geradas todas as fontes virtuais contidas em salas cujos índices (i, j, k), indicadores de suas posições em relação à sala real, satisfaçam a condição $|i| + |j| + |k| \leq R$.

A lista das intensidades das reflexões que chegam ao ouvinte a cada instante é gerada e o vetor correspondente à resposta impulsiva é gerado pelo método *generateFFTInputArray*, que está no arquivo *FrequencyResponse.java*. Como os instantes de chegada de cada reflexão, para salas pequenas, são menores que 1, foram tomadas as x casas decimais seguintes como sendo os índices daquela reflexão em um vetor adequado de tamanho igual a uma potência de 2. Aqui, x é igual a $-\log_{10}\epsilon$, onde ϵ é uma constante real próxima de 0. Os valores para o raio máximo de salas e para o erro máximo permitido estão definidos no arquivo *Defs.java*, nas variáveis *maxSum* e *epsilon*, respectivamente.

A resposta impulsiva assim simulada é uma função com muitas descontinuidades, o que gera no espectro uma quantidade muito grande de ruído, especialmente em altas freqüências. Uma maneira de compensar este efeito é aplicar um filtro passa-baixa [Moore, 1990] à resposta impulsiva produzida. O método *applyFilter*, também em *FrequencyResponse.java*, implementa e aplica à resposta impulsiva um filtro FIR.

Resposta em Freqüência e Cálculo da Distorção

Unindo os dois procedimentos descritos nas seções anteriores (FFT e cálculo da resposta impulsiva), torna-se simples obter a resposta em freqüência da sala. A resposta logarítmica em freqüência é utilizada a seguir para calcular a distorção, de acordo com a definição apresentada. O método *stdDeviation*, no arquivo *FrequencyResponse.java*, é responsável por realizar o cálculo da resposta impulsiva, gerar a partir deste resultado a entrada adequada para o algoritmo da FFT e obter a resposta em freqüência, tudo isso através de métodos auxiliares. Este método foi encapsulado na classe *ObjectiveFunctionPAOSE* para que pudesse ser chamado a partir dos algoritmos de otimização.

Algoritmos de Otimização

Como mencionado anteriormente, foram implementados dois algoritmos de otimização: o *Density Clustering* e o *Grid-based method*. Ambos foram escritos de forma que pudessem ser reutilizados para a resolução de outros problemas, permanecendo independentes desta aplicação específica. Foi criada também uma classe *Cache*, responsável por armazenar os valores da função objetivo em todos os pontos já percorridos pelo algoritmo. Essa estratégia foi escolhida porque cada cálculo do valor da função objetivo é relativamente caro; por isso, é necessário aproveitar todos os cálculos já efetuados.

O algoritmo *Grid*, implementado no método *executeAlgorithm* do arquivo *GridMethod.java*, depende dos seguintes parâmentros: a dimensão d do problema, o ponto de partida do algoritmo *xInit*, uma matriz contendo os limites de cada uma das d coordenadas dos pontos a serem sorteados, um objeto do tipo *ObjectiveFunction*, que deve encapsular a chamada da função objetivo desejada, o número máximo de iterações permitido, o tamanho da malha inicial, e o objeto *Cache* que fará o armazenamento dos pontos visitados.

O algoritmo *Density Clustering* foi implementado no método *executeAlgorithm* do arquivo *DensityClustering.java*. Ele executa uma série de iterações bus-

cando o ótimo global de um problema. Para a execução deste método, é necessário o conhecimento dos seguintes valores: a dimensão d do problema, o número N de pontos a serem sorteados a cada iteração, a fração de redução γ, o número máximo de iterações desejado para o algoritmo, o número máximo de iterações para o método de otimização local, uma matriz contendo os limites de cada uma das d coordenadas dos pontos a serem sorteados, um objeto do tipo *ObjectiveFunction*, que deve encapsular a chamada da função objetivo desejada, a medida de Lebesgue do conjunto viável (que para este problema de otimização consiste simplesmente no produto das dimensões da sala) e o objeto *Cache* que fará o armazenamento dos pontos visitados. O procedimento de minimização local usado nesta implementação do *Density Clustering* foi o próprio *Grid-based method*.

O Programa

Em `http://gsd.ime.usp.br/acmus/` encontra-se disponível para download o programa a que se refere este artigo. O método principal do programa do Projeto Acústico Ótimo de Salas de Escuta (PAOSE) encontra-se no arquivo *MainPAOSE.java*. A interface (em linha de comando) do programa é

```
java MainPAOSE arquivo.txt [OPÇÃO]
```

O arquivo.txt deve conter os dados da sala (dimensões e coeficientes de absorção) no seguinte formato:

```
X Y Z
alpha0 alpha1 beta0 beta1 gamma0 gamma1
```

onde X, Y e Z são as dimensões da sala e alpha0, ..., gamma1 são os coeficientes de absorção de suas superfícies. O segundo argumento consiste em um código que define o procedimento de otimização a ser utilizado no programa. As opções são `-d` para o método *Density Clustering* com o algoritmo baseado em grades como método de otimização local e `-g` para o método baseado em grades como método de otimização global.

O programa imprime na saída padrão as seguintes informações: tempo de execução até a obtenção da resposta; localizações ótimas encontradas para fonte e ouvinte; quantidade de distorção gerada nas localizações ótimas encontradas; maior quantidade de distorção dentre as geradas nos pontos visitados pelo método de otimização; este valor é utilizado para comparação com o ótimo. Ainda, definindo o valor da variável booleana *debug* do arquivo *Defs.java* como *true*, o programa imprime na saída padrão o caminho percorrido pelo algoritmo de otimização.

Testes Computacionais

Os testes realizados até o momento são preliminares e sugerem algumas modificações, discutidas na seção a seguir. Considerou-se inicialmente o problema de localizar duas fontes e um ouvinte, dispostos simetricamente em uma sala de escuta de dimensões $X = 3m \times Y = 4m \times Z = 5m$. Uma das fontes ocupa a posição (x_F, y_F, z_F) e a outra $(X - x_F, y_F, z_F)$, onde (x_F, y_F, z_F) satisfaz $0 \leq x_F \leq X/2$, $Y/2 \leq y_F \leq Y$ e $0 \leq z_F \leq Z$. O ouvinte deve ser localizado em (x_O, y_O, z_O) onde $x_O = X/2$ e $0 \leq y_O \leq Y/2$. Este é um problema que pode ser resolvido considerando-se apenas

a primeira fonte e o ouvinte, visto que a resposta impulsiva produzida pela segunda fonte é idêntica à produzida pela primeira, o que é conseqüência direta da simetria, e portanto a distorção harmônica produzida pela segunda fonte é também idêntica.

O método *Grid* aplicado aos dados acima produziu a seguinte resposta

```
Número de avaliações da função no Grid: 75
Grid Method - tempo de resposta: 87.509 s
Valor do pior ponto encontrado: 1.4202278753437534
- Localizações ótimas encontradas -
Fonte: (0.2109375, 3.5, 4.84375)
Ouvinte: (1.5, 0.5, 0.0)
Quantidade de distorção: 0.5617007729474769
```

Neste exemplo observa-se uma variação de 252.84% entre a pior localização e a melhor localização encontrada. Vela a pena mencionar também que a localização inicial do método foi $(x_F = 0.0, y_F = 2.0, z_F = 0.0)$ e $(x_O = 1.5, y_O = 0.0, z_O = 0.0)$, com uma distorção de 1.339057806330562 (238.38% acima do ótimo).

O método *Density Clustering* produziu a resposta

```
Density Clustering - tempo de resposta: 1378.040 s
Valor do pior ponto sorteado: 0.9838447204945652
- Localizações ótimas encontradas -
Fonte: (1.499908447265625, 3.5316162109375, 4.10125732421875)
Ouvinte: (1.5, 0.8748779296875, 0.00152587890625)
Quantidade de distorção: 0.5761921592717776
```

Observe que o tempo de resposta é bem mais alto do que o Grid, o que é esperado, visto que o Grid é uma subrotina do Density Clustering, utilizada um grande número de vezes. Neste teste em particular a solução final encontrada não era melhor do que a do Grid, o que pode ser explicado como "sorte" do método Grid em encontrar um mínimo local tão bom em apenas uma tentativa. Outras considerações sobre a comparação dos métodos é dada na seção seguinte.

Os mesmos dados de descrição da sala foram fornecidos aos dois métodos sem a exigência de simetria das localizações; neste caso apenas uma fonte foi considerada, que poderia ser localizada em qualquer ponto da sala, o mesmo acontecendo com o ouvinte. Os dados obtidos foram

```
Número de avaliações da função no Grid: 57
Grid Method - tempo de resposta: 61.775 s
Valor do pior ponto encontrado: 0.7797158869018019
- Localizações ótimas encontradas -
Fonte: (2.25, 0.0, 0.0)
Ouvinte: (2.25, 2.0, 2.5)
Quantidade de distorção: 0.3958269857678017

Density Clustering - tempo de resposta: 850.816 s
Valor do pior ponto sorteado: 1.7976931348623157E308
- Localizações ótimas encontradas -
```

```
Fonte: (1.6875, 1.75, 2.5)
Ouvinte: (1.6875, 1.75, 2.490234375)
Quantidade de distorção: 0.36589052199761113
```

No último exemplo observa-se o comportamento típico dos dois métodos. O Density Clustering encontrou uma solução melhor do que a do Grid, em tempo também maior. Outras observações sobre as implementações são feitas a seguir.

5. Conclusão e Trabalhos Futuros

Os resultados da seção anterior são ainda preliminares, pois várias melhorias ainda podem ser feitas no método levando-se em consideração apenas o modelo de sala atual. No entanto, pode-se afirmar que são inovadoras a proposta de usar estratégias de otimização global e a disponibilização de código-aberto para a localização de fontes e ouvinte em salas de escuta.

Considerando que o método baseado em Grade busca apenas um minimizador local da função, é natural que seu tempo de execução seja muito menor que o método *Density Clustering*, que em particular efetua várias chamadas do *Grid*. É natural supor que um usuário escolherá o método de busca global apenas quando tiver tempo de esperar a resposta. Assim sendo, algumas variantes do método devem ser exploradas. Uma destas variantes corresponde a substituir a base positiva do método Grid $V_+^{(m)} = \{V_1^{(m)}, \ldots, V_n^{(m)}, -\sum_{i=1}^n V_i^{(m)}\}$, escolhida para minimizar o número de chamadas da função objetivo, pela base $V_+^{(m)} = \{V_1^{(m)}, \ldots, V_n^{(m)}, -V_1^{(m)}, \ldots, -V_n^{(m)}\}$, que fornece mais liberdade de movimentação para o método e aumenta a chance de se encontrar uma direção de descida a partir do ponto atual.

Outras extensões do trabalho correspondem a melhorias no modelo da sala e no resultado da simulação por acústica geométrica. A consideração com maior prioridade de inclusão é a representação dos coeficientes de absorção como funções da freqüência. Neste respeito podem ser utilizadas tabelas de materiais com valores de coeficiente de absorção por bandas de freqüência (por exemplo terços de oitava), ou ainda aproximações baseadas em modelos matemáticos das curvas de absorção, com preferência para o que representar o menor *overhead* computacional para o método. Coeficientes de absorção variam também em função do ângulo de incidência; tal dependência poderia também ser levada em consideração através de valores tabulados ou de modelos analíticos.

A consideração de outras geometrias além da cubóide apresenta grande interesse prático e também grande dificuldade teórica e computacional. Ao considerar-se salas de geometria poliédrica geral, o modelo das fontes virtuais deixa de possuir fórmulas fechadas para a localização das fontes e passa a depender de técnicas de álgebra linear; também não é desprezível o fato de que em geral o número de fontes virtuais de ordem n cresce exponencialmente em função de n (para geometria cubóide esse crescimento é polinomial). Uma alternativa é utilizar o modelo de traçado de raios, que gera uma grande quantidade de raios saindo da fonte real em todas as direções, e acompanha o caminho geométrico destes raios até atingirem o ouvinte; ou ainda utilizar uma estratégia híbrida de fontes virtuais e traçado de raios.

Finalmente, outros fenômenos acústicos tais como difusão, difração e sombreamento acústico (referente à visibilidade de fontes virtuais) poderiam ser considerados, às custas de um aumento considerável tanto de complexidade no modelo matemático como de custo computacional. Embora seja esperado que estas considerações tornem o modelo mais robusto e condizente com a realidade, no contexto do projeto automático de salas tais inclusões devem ser avaliadas em relação à melhoria real da solução obtida pelo método de otimização e o acréscimo de custo computacional envolvido.

Referências

AcMus (2005). `http://gsd.ime.usp.br/acmus/projeto.html`.

Beranek, L. L. (1993). *Acoustics*. New York: Acoustical Society of America.

Coope, I. D. and Price, C. J. (2001). On the convergence of grid-based methods for unconstrained optimization. In *SIAM J. Optim. 11(4)*, pages 859–869.

D'Antonio, P. and Cox, T. J. (1997). Room optimizer: A computer program to optimize the placement of listener, loudspeakers, acoustical surface treatment and room dimensions in critical listening rooms. In *103rd Convention of the Audio Engineering Society*, pages Preprint 4555, Paper H–6. New York.

F. H. Iazzetta, F. K. and Silva, F. (2001). Acmus: Design and simulation of musical listening environments. In *Proceedings of the 8th Brazilian Symposium on Computer Music*. Fortaleza.

Horst, R. and Pardalos, P. M. (1995). *Handbook of Global Optimization*. Kluwer Academic Publishers.

López, M. R. (2001). *Acondicionamiento Acústico*. Paraninfo.

Moore, F. R. (1990). *Elements of Computer Music*. Prentice-Hall.

Morrison, N. (1994). *Introduction to Fourier Analysis*. Wiley-Interscience.

Queiroz, M. (2003). Some optimization models for listening room design. In *Proceedings of the 9th Brazilian Symposium on Computer Music*. Campinas.

Rindel, J. H. (1997). Computer simulation techniques for the acoustical design of rooms - how to treat reflections in sound field simulation. In *Proceedings of ASVA'97*, pages 201–208. Tokyo.

Rindel, J. H. (2000). The use of computer modeling in room acoustics. In *Journal of Vibroengineering*, pages 41–72.

Rinnoykan, A. H. G. and Timmer, G. T. (1987). Stochastic global optimization methods. part i: clustering methods. In *Mathematical Programming 39*, pages 27–56.

Warusfel, O. (1995). Predictive acoustics software and computer aided optimization in room acoustics. In *15th Intl. Congress on Acoustics*, pages 693–696. Trondheim, Norway.