

## Visualização da propagação de ondas sonoras no espaço utilizando o VTK – Visualization Toolkit

Ricardo Casaleiro, Bruno Quintela, Frederico Valente, Vítor Rodrigues,  
Guilherme Campos, Paulo Dias

**Resumo** - Com o auxílio da biblioteca para visualização 3D VTK (Visualization Toolkit), foram desenvolvidas aplicações para visualizar dados planares (2D) e tridimensionais (3D) gerados por um programa de modelação acústica por malha de guias de onda digitais (DWM – Digital Waveguide Mesh). Eficientes e com interfaces de uso intuitivo, as aplicações desenvolvidas facilitam a análise e interpretação dos resultados nos domínios do tempo e da frequência. O aprofundamento deste trabalho poderá conduzir a ferramentas com grande utilidade na validação dos modelos acústicos em causa.

### I. INTRODUÇÃO

O estudo dos campos sonoros que se estabelecem em espaços fechados é um dos principais focos da Acústica Arquitectural. A possibilidade de visualizar as características desses campos sonoros reveste-se de grande interesse, sobretudo nas áreas de projecto e correcção acústica de salas. O objectivo dos trabalhos aqui descritos, propostos no âmbito da disciplina de *Modelação e Visualização 3D* (MV3D – 5º ano do curso de ECT), era testar a biblioteca gráfica VTK (Visualization Toolkit) 1,2 para esse efeito. O desafio consistiu em conceber aplicações para visualização e análise de dados produzidos por um modelo de simulação acústica de salas desenvolvido anteriormente 3. A expectativa é que tais aplicações possam contribuir para o processo de refinamento e validação desse mesmo modelo. Os trabalhos exploraram não só a propagação de uma onda sonora ao longo do tempo, como também a variação espacial da resposta em frequência, tendo em vista, nomeadamente, detectar as ondas estacionárias associadas às frequências de ressonância ou *modos* do espaço.

### II. VTK (VISUALIZATION TOOLKIT)

O VTK é um pacote de *software* aberto (*open-source*) para computação gráfica, visualização e processamento de imagem. É orientado a objectos e suportado em várias plataformas. Implementado em C++, também pode ser utilizado com as linguagens TCL, Python e Java, prestando-se assim ao desenvolvimento de aplicações complexas, criação rápida de protótipos de aplicações e *scripting*. Não fornece componentes de interface de

utilizador, mas pode ser integrado com bibliotecas tais como Tk, X/Motif, JAVA, Visual Studio ou Qt.

O VTK admite várias formas de representar a informação, incluindo conjuntos desorganizados de pontos, informação poligonal, imagens, volumes e grelhas, estruturadas ou não. Disponibiliza também filtros e algoritmos que permitem operar sobre todas elas. Em particular, as grelhas estruturadas de pontos podem ser representadas de forma eficiente e manipuladas através de uma série de ferramentas e *widgets* (“objectos de janela”) 1,2. Esta constituiu a principal indicação de que o VTK seria adequado à análise dos dados acústicos em causa, pois, como veremos no ponto seguinte, eles baseiam-se na discretização do espaço segundo uma grelha estruturada de pontos.

Para criar as aplicações apresentadas neste trabalho, o VTK foi utilizado em conjunto com programação em C++.

### III. MODELAÇÃO ACÚSTICA

Os dados tratados nos trabalhos aqui apresentados foram produzidos por *malhas 3D de guias de ondas digitais* (Digital Waveguide Meshes – DWM) 3. A modelação física por guias de onda digitais constitui um método numérico do tipo FDTD (diferenças finitas no domínio do tempo) para resolver a equação diferencial que rege a propagação de ondas sonoras (equação de onda). Foi desenvolvido por investigadores do CCRMA (Universidade de Stanford) tendo em vista aplicações musicais. As suas principais vantagens são a simplicidade matemática do algoritmo e o significado claro e intuitivo dos parâmetros do modelo 4. É válido para qualquer número de dimensões. As versões 1D e 2D têm conhecido assinalável sucesso em aplicações de síntese de som e simulação de instrumentos acústicos 5. As malhas 3D possibilitam a simulação acústica de salas de forma potencialmente muito rigorosa. De facto, por se tratar de modelação física, todos os fenómenos de onda, incluindo difracção e interferência, são automaticamente tidos em conta (ao contrário do que se passa nos modelos simplificados normalmente empregues para esse efeito, de que se destaca o *ray-tracing*). A principal dificuldade prática deste método reside na sua elevada exigência computacional 6.

O espaço em análise é discretizado através de uma grelha uniforme de pontos (nós). Várias topologias são possíveis

7. Neste trabalho, foi empregue a mais simples, ilustrada na Figura 1.

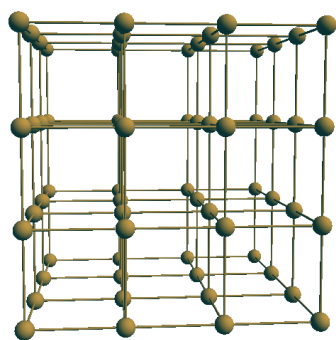


Figura 1 – Malha 3D de guias de onda digitais: topologia rectilínea

Uma *DWM* permite simular propagação 3D porque cada nó comunica com os seus vizinhos próximos. Esta comunicação é realizada através de linhas de atraso unitárias, que implementam discretização temporal: as ondas transmitidas por cada nó são recebidas nos vizinhos passado um período de amostragem; a pressão acústica e as ondas a enviar podem então ser (re)calculadas em todos os nós do modelo, iniciando-se uma nova iteração do algoritmo.

Em função da sua localização no modelo, os nós são classificados em duas categorias básicas:

- Nós *de ar*;
- Nós *fronteira*.

Os primeiros modelam o meio de propagação. Podem ser-lhes atribuídas funções especiais:

- Excitação – nos nós com esta função é injectado o sinal (anecóico) emitido pela fonte sonora que se pretende simular.
- Recepção – nestes é registada da resposta (sinal de saída) do modelo.

Os nós *fronteira* representam as superfícies que delimitam o espaço, sendo configurados de acordo com as propriedades acústicas dos respectivos materiais.

Assim, é possível modelar o comportamento acústico de salas com quaisquer características físicas, para quaisquer posições de fonte e receptor 6.

#### IV. VISUALIZAÇÃO DE DADOS 2D

Ao primeiro grupo de alunos, foi proposta a visualização de dados 2D correspondentes à propagação de uma onda sonora ao longo de um plano horizontal numa sala paralelepípedica contendo dois obstáculos e uma coluna. O sinal utilizado na excitação do modelo foi um impulso de Dirac (ver Figura 2).

##### A. Dados fornecidos

O cabeçalho do ficheiro fornecido pelo modelo acústico (ver 0) indicava a dimensão do plano de corte (número de nós segundo os eixos horizontal e longitudinal), o número de instantes de amostragem considerado e a frequência de amostragem (11025 Hz). O corpo do ficheiro continha os

dados propriamente ditos: por cada instante de amostragem, uma matriz 2D contendo os valores instantâneos da amplitude da onda sonora em todos os nós situados no plano de corte.

Os dados foram carregados do ficheiro para o programa por intermédio de uma classe desenvolvida para o efeito (*SoundDataLoader*) e armazenados sob a forma de grelha estruturada (classe *VTKImageData*).

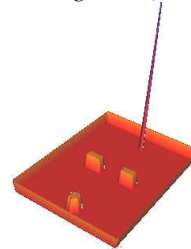


Figura 2: Sala em estudo no momento da excitação inicial

##### B. Programa de visualização

Pretendia-se analisar a evolução temporal não só do valor instantâneo da pressão acústica em todos os nós do plano mas também da sua excursão máxima, no sentido de detectar eventuais nós de ondas estacionárias (pontos onde a pressão acústica se mantém sempre nula). Assim, optou-se por visualizar a informação em duas janelas distintas com interacção separada.

A biblioteca gráfica *VTK* utiliza uma *pipeline* gráfica para a visualização, que define um conjunto de operações ligadas entre si, desde a leitura dos dados até ao *rendering* final no monitor. Cada uma delas representa uma transformação ou filtragem dos dados.

Na Figura 3 é descrita a *pipeline* utilizada para a visualização do valor instantâneo. O programa começa por carregar os dados do ficheiro sob um formato do *VTK*. Em seguida, normaliza-os e converte-os numa estrutura tridimensional (classe *VTKImageData*) passível de ser transformada pelos filtros que se seguem. Devido à grande disparidade observada entre os valores da onda, adoptou-se uma escala logarítmica (*VTKImageLogarithmicScale* na *pipeline* gráfica). Para criar uma visualização 3D de interpretação intuitiva, aplicou-se então um filtro (*VTKImageDataGeometryFilter*) que extrai a informação relativa ao instante da simulação que se pretende visualizar em cada momento, e um outro filtro (*VTKWarpScalar*) que associa a cada ponto do plano ( $x,y$ ) uma elevação  $z$  proporcional ao valor de pressão no nó correspondente.

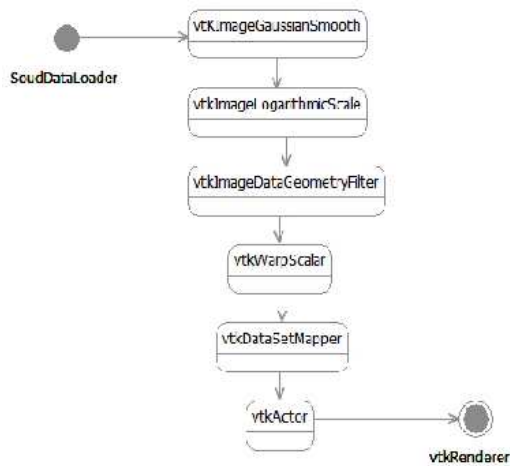


Figura 3: Pipeline de visualização utilizada na janela da pressão acústica

Verificou-se que existiam transições muito bruscas entre os valores (devidas em parte à esparsidade espacial dos dados), que resultavam numa visualização inestética, a partir da qual era difícil extrair informação e obter uma visão global da propagação da onda no tempo. Para resolver esse problema, optou-se por introduzir, no início da *pipeline*, uma filtragem capaz de suavizar a visualização (*VTKGaussianSmooth*). Naturalmente, por modificar ligeiramente os valores originais das amostras, esta filtragem introduz algum ruído, que se procurou minimizar escolhendo um valor baixo para o raio do filtro gaussiano aplicado em cada nó. O final da *pipeline* inclui um *VTKDataSetMapper* associado ao respectivo *VTKActor*, objecto que é colocado na janela de visualização. Em conjunto, eles permitem definir a forma como é feito o *rendering* da superfície final em 3D.

A *pipeline* utilizada para a janela de visualização da excursão máxima é muito semelhante à que acaba de se descrever. A principal diferença consiste na ausência de filtragem gaussiana dos dados.

### C. Resultados

A Figura 4 ilustra a aplicação desenvolvida para visualizar a propagação da onda sonora. Para cada instante, é mostrada a pressão acústica em todos os pontos do plano, através de uma superfície tridimensional ( $x, y, z$ ). A elevação  $z$  representa o valor da pressão sonora no ponto ( $x, y$ ) correspondente. Para facilitar a interpretação do gráfico, foi aplicada uma escala de cores. Na mesma janela são também desenhadas, no plano de elevação  $z = 0$ , as iso-linhas que unem os pontos de pressão nula.

Procedeu-se de forma semelhante para a visualização da excursão máxima (

Figura 5). O seu valor encontra-se representado na ordenada  $z$ ; aplicou-se igualmente uma escala de cores.

Em ambas as janelas, a interacção com os objectos gráficos é assegurada pelas funções do *VTK*. É possível, através de combinações de teclas adequadas, fazer avançar e recuar o tempo na visualização.

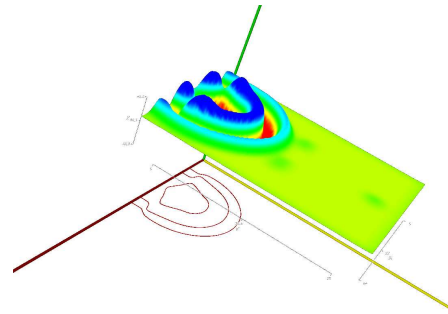


Figura 4: Visualização da propagação da onda sonora

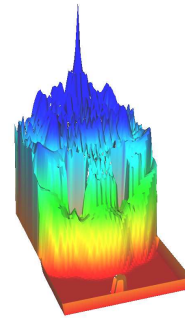


Figura 5: Visualização dos valores máximos de pressão em cada nó

O programa permite ainda a selecção interactiva de nós individuais e a visualização dos seus valores exactos de pressão acústica. Para tal, foi acrescentada uma classe derivada da classe-base *VTKCommand*, a ser utilizada como *callback* em caso de selecção de um nó. Esta facilidade mitiga o problema criado pela filtragem gaussiana, já que são apresentados os valores reais de pressão no nó seleccionado e não os valores pós-filtragem.

## V. VISUALIZAÇÃO DE DADOS 3D

Ao segundo grupo de alunos, foi proposta a visualização de dados 3D, obtidos pelo mesmo método de modelação acústica (ver 0), referentes à propagação de uma onda sonora a partir de um dos cantos de uma sala forma cúbica (1 m de lado), vazia. Pretendia-se explorar a visualização destes dados não só no domínio do tempo como também no domínio da frequência.

### A. Dados fornecidos

Tal como na visualização 2D (ver 0, o ficheiro de dados inicial incluía um cabeçalho indicando as dimensões do modelo (número de nós segundo os 3 eixos), o número de iterações armazenadas e a frequência de amostragem (11025 Hz). Seguiam-se os dados: uma sequência de matrizes 3D ( $x \times y \times z$ ) contendo os valores de pressão acústica observados em cada iteração. Dado que o sinal de excitação utilizado fora (como antes) um impulso de Dirac, esta informação constituía, para este

posicionamento da fonte sonora, um registo das respostas impulsivas (*IR*) em todos os nós do modelo.

A transformada de Fourier de cada *IR* dá a resposta em frequência da sala no ponto correspondente. Através de um *script* em *MATLAB*, aplicou-se o algoritmo *FFT* (*Fast Fourier Transform*) para criar, a partir do ficheiro de dados inicial, um novo ficheiro contendo a amplitude da resposta em frequência (espectro de amplitude) em todos os nós. O ficheiro de respostas impulsivas e o ficheiro de espectros de amplitude são idênticos em estrutura e também em dimensão, dado que o número de componentes da *FFT* de um sinal é igual ao seu número de amostras.

### B. Estratégias de visualização

Os dados no domínio do tempo (respostas impulsivas em cada ponto) e no domínio da frequência (espectros de amplitude em cada ponto) são apresentados de forma idêntica. Trata-se, em ambos os casos, de visualizar uma grandeza escalar variável num domínio 3D, o que coloca óbvias dificuldades práticas. A solução adoptada combina dois métodos:

- geração de *iso-superfícies* (i.e. superfícies unindo pontos em que a grandeza escalar em análise tem o mesmo valor);
- uso de *planos de corte* (um vertical e outro horizontal), livremente posicionáveis pelo utilizador.

A magnitude da grandeza é visualizada através de uma escala de cores. O utilizador pode seleccionar o seu ponto de vista livremente e de forma independente em ambos os métodos de visualização, fazendo rodar o modelo no sentido desejado. Pode também controlar a evolução da variável independente:

- *tempo*, no caso da visualização das respostas impulsivas – é assim possível visualizar a configuração da onda sonora em cada instante, bem como a sua progressão (ou, fazendo o tempo recuar, regressão) através da sala.
- *frequência*, no caso da visualização dos espectros de amplitude – o que permite visualizar a distribuição espacial do campo sonoro em função da frequência.

### C. Algoritmos de visualização

O primeiro passo consiste em ler os dados do ficheiro sob um formato do *VTK*. São definidos, usando *VTKImageData*, agregados de grelhas estruturadas. Cada grelha corresponde a um instante de tempo ou a uma frequência específica, consoante se estejam a visualizar respostas impulsivas ou espectros de amplitude. Após o preenchimento das grelhas estruturadas, são criadas, através da classe *VTKContourFilter*, iso-superfícies unindo os pontos do espaço com igual valor (de pressão instantânea ou amplitude espectral, conforme o caso). As acentuadas diferenças de valores observadas entre pontos vizinhos, relacionadas com a baixa resolução espacial da grelha (cerca de 5.4 cm de espaçamento inter-nodal), resultavam na criação de iso-superfícies em torno de cada

nó. Este problema foi resolvido de modo similar ao utilizado no caso 2D: aplicou-se interpolação entre os valores dos nós vizinhos, recorrendo a *VTKGaussianSplatter*, suavizando assim as transições.

Com o auxílio de *VTKPlaneWidget*, é criado um novo *renderer* para visualizar os cortes horizontal e vertical posicionados no modelo. É criada uma janela para visualizar em detalhe o último plano de corte seleccionado. Desta forma, a posição do corte no modelo 3D é mostrada a par dos dados correspondentes, facilitando o seu estudo detalhado. Face à esparsidade espacial da informação, foi usado *VTKImageReslice* para ampliar a imagem do corte na janela e interpolar dados de nós vizinhos, melhorando assim a qualidade da visualização.

### D. Resultados

As Figura 6 e Figura 7 ilustram a visualização da evolução temporal da pressão sonora, mostrando os resultados obtidos em dois instantes separados por 2.4 ms (iterações 3 e 29, respectivamente).

Do lado esquerdo são apresentados os planos de corte (horizontal e vertical), cujo posicionamento é controlado pelas teclas do cursor. À direita encontram-se as iso-superfícies de pressão acústica (a vista foi rodada para melhor as evidenciar). É visível que a fonte sonora se localiza num dos cantos da sala.

O mapa de cor aplicado nas iso-superfícies e planos de corte é definido por uma tabela (*VTKLookupTable*). Cada valor (e portanto cada iso-superfície) é mapeado num tom diferente, produzindo um efeito de gradação, evidente sobretudo na Figura 7. Um objecto do tipo *VTKScalarBarActor* indica o mapeamento utilizado.

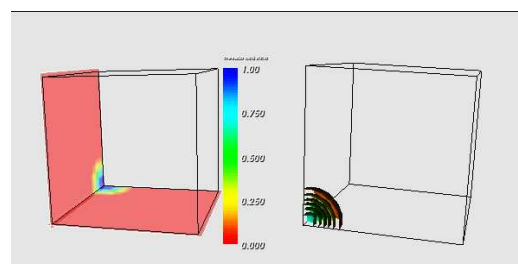


Figura 6: Visualização da pressão acústica na iteração 3 (0,27 ms)

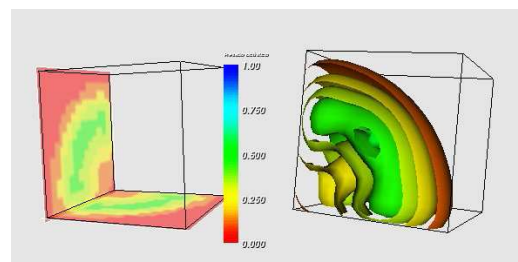


Figura 7: Visualização da pressão acústica na iteração 29 (2,63 ms)

Na visualização de espectros de amplitude, a variável ‘frequência’ é controlada de forma inteiramente análoga à usada para ‘tempo’ na visualização de respostas impulsivas. A principal motivação para explorar os dados desta forma no domínio da frequência reside em tentar determinar, por inspecção visual, as frequências a que ocorrem ondas estacionárias – ditas frequências de ressonância ou modos de vibração. Estes modos assumem grande importância no estudo do comportamento acústico de uma sala. Neste caso, dada a forma geométrica particularmente simples, é possível calculá-los analiticamente. A figura 8 mostra o resultado obtido à frequência de 167.5 Hz, correspondente ao 1º modo axial da sala.

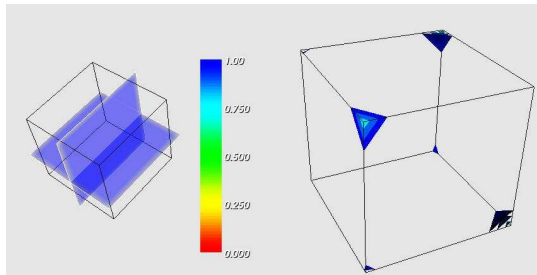


Figura 8: Visualização da amplitude da resposta à frequência de 167.5 Hz

## VI. CONCLUSÕES E TRABALHO FUTURO

Este trabalho reforçou a convicção de que o *VTk* constitui uma excelente ferramenta para visualizar o comportamento de modelos físicos de simulação acústica de salas, podendo assim auxiliar o seu desenvolvimento e validação. De facto, aproveitando as potencialidades de interacção e filtragem de dados oferecidas pelo *VTk*, foi possível desenvolver aplicações que, embora simples, se mostraram eficientes e capazes de produzir visualizações versáteis, facilmente controláveis pelo utilizador e de interpretação intuitiva.

Por razões de ordem prática, os testes restringiram-se a modelos de dimensões relativamente reduzidas. Ainda assim, tornou-se evidente o valor didáctico das visualizações no domínio do tempo.

O trabalho carece de aprofundamento sobretudo no que toca à visualização de dados no domínio da frequência. O objectivo mais imediato será determinar os modos da resposta em frequência de uma sala através da detecção dos padrões visuais (nomeadamente superfícies nodais) que denunciam a presença de ondas estacionárias. Salas de geometria simples (como a utilizada em 0), em que seja possível calcular analiticamente os modos, permitirão validar os resultados. O objectivo seguinte será desenvolver um algoritmo de identificação automática de ondas estacionárias para aplicação a salas de geometria complexa, que não permitam cálculo analítico dos modos.

Seria também interessante testar modelos com maior resolução, simulando salas maiores e de geometria mais complexa, bem como diferentes posicionamentos da fonte

sonora, para avaliar cabalmente o desempenho dos vários programas de visualização desenvolvidos (em 2D e 3D, nos domínios do tempo e da frequência) e ajustar os seus parâmetros (nomeadamente de filtragem gaussiana).

## REFERÊNCIAS

1. Lisa S. Ávila et al. (2004) “The *VTk* User’s Guide”. Kitware.
2. Will Schroeder et al. (1998) “The Visualization Toolkit – 2<sup>nd</sup> edition”. Prentice Hall.
3. Campos, G. (2003) “Three-Dimensional Digital Waveguide Mesh Modelling for Room Acoustic Simulation”. Tese de Doutoramento, Universidade de York (UK).
4. Scott A. Van Duyne e Julius O. Smith III (1993) “Physical Modeling with the 2-D Digital Waveguide Mesh” (*ICMC Proceedings*, Tokyo, pp. 40-47).
5. Julius O. Smith III (1998) “Principles of digital waveguide models of musical instruments”, pp. 417-466 in *Applications of Digital Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg (eds). Norwell, MA: Kluwer.
6. Campos, G. and Howard, D. M. (2000) “On the Computation Time of Three-Dimensional Digital Waveguide Mesh Models (*Proceedings of the 26<sup>th</sup> Euromicro Conference*, Maastricht, Holland, September 5-7, vol. II, pp. 332-339).
7. Campos, G. and Howard, D. M. (2005) “On the Computational Efficiency of Different Waveguide Mesh Topologies for Room Acoustic Simulation”. (*IEEE Transactions on Speech and Audio Processing*, Volume 13, Issue 5, September, pp. 1063 – 1072)
8. Everest, F. (1994) “The Master Handbook of Acoustics”. New York: TAB Books – McGraw-Hill.