

# Visualização Interactiva Tridimensional em Java3D

Bruno Caiado\*  
Lic. Eng<sup>a</sup>. Informática, IST  
Av. Rovisco Pais, 1049-001 Lisboa  
bruno.caiado@megamedia.pt

Luís Correia\*  
Lic. Eng<sup>a</sup>. Informática, IST  
Av. Rovisco Pais, 1049-001 Lisboa  
luis.correia@megamedia.pt

João Brisson Lopes  
Dep. Eng<sup>a</sup>. Informática, IST  
Av. Rovisco Pais, 1049-001 Lisboa  
brisson@ist.utl.pt

\* presentemente: Megamedia,  
R. António Pedro 111, 1º, 1150-045 Lisboa

---

## Sumário

*Esta comunicação descreve a concepção e realização em Java/Java3D da aplicação NISVAS para visualização interactiva de conjuntos de dados de grande volume resultantes da simulação numérica de fenómenos complexos. A aplicação, que é portátil, visualiza estes dados através de cores, símbolos, vectores e animações. Descrevem-se os objectivos, a arquitectura e a funcionalidade do visualizador, com realce para a realização da interface com o utilizador que é simples, de fácil aprendizagem e suporta utilizadores com diferentes níveis de experiência. As reacções dos utilizadores permitiram concluir que os objectivos propostos foram alcançados, nomeadamente no referente ao desempenho e à simplicidade e flexibilidade de uso. Apresentam-se direcções para a continuação deste trabalho.*

## Palavras-chave

*visualização interactiva, visualização científica, Java, Java3D, interacção, interface ao utilizador*

---

## 1. INTRODUÇÃO

A simulação de fenómenos complexos tridimensionais origina conjuntos de resultados numéricos cuja análise e compreensão é muito difícil devido ao grande volume e à natureza diversa dos muitos resultados obtidos. Em fenómenos como, por exemplo, a deformação de estruturas em impactos violentos ou o escoamento num leito variável das águas de um rio durante uma cheia, obtêm-se campos de valores de grandes dimensões de deformações, estado de tensão, temperatura, velocidades e concentrações, que, muitas vezes, variam no tempo.

A solução para este problema consiste em visualizar conjuntamente os resultados, combinando o que é visível (uma peça deformada) com o que o não é, como o estado de tensão em toda a peça, “visualizando o invisível”.

O número de grandezas escalares e vectoriais a visualizar pode facilmente atingir e ultrapassar a dezena de grandezas. O utilizador necessita dispor de uma interface simples e poderosa que lhe permita mudar rapidamente a forma como cada grandeza é visualizada ou substituí-la por outra. Além disto, a interface deve permitir ao utilizador o reconhecimento imediato das grandezas que pode manipular e que operações de visualização pode com elas efectuar. Isto coloca requisitos especiais ao desenho da interface, nomeadamente na apresentação do estado da aplicação, na sugestão de operações e na prevenção de erros.

Tradicionalmente, as aplicações de visualização são aplicações especializadas, dependentes de tecnologias proprietárias, do sistema operativo e do hardware gráfico, pois exigem grande capacidade de processamento gráfico. Em consequência, são aplicações concebidas para plataformas específicas e de custo elevado.

A dependência da plataforma também tem como consequência tornar a interface com o utilizador dependente do sistema de janelas da plataforma seleccionada que, por sua vez, impõe um “look and feel” específico à interface. É o caso de aplicações de visualização como o Wavefront Data Visualizer, PV-WAVE, ou VIS5D, entre outros. Este é também o caso do visualizador ISVAS (Interactive System for Visual Analysis, [Karlsson93], [Karlsson94], [Haase95]) que depende de hardware gráfico Silicon Graphics e da API GL, para a apresentação das visualizações sob o sistema de janelas 4Dwm do sistema operativo Irix, e do sistema de janelas Motif para o diálogo com o utilizador. Este tipo de solução era perfeitamente justificável na altura em que aquele visualizador foi concebido.

Entretanto, a capacidade do hardware aumentou e verificou-se igualmente uma evolução significativa dos paradigmas de construção de software e da interacção pessoa-máquina que se traduziram na adopção da Programação Orientada para Objectos e na separação clara entre o software gráfico e o hardware de saída gráfica. A generalização destes paradigmas e do uso de linguagens como C++, OpenGL, Java ou Java3D permitem desenvolver

aplicações de visualização de construção mais simples e mais fáceis de manter cujas interfaces com o utilizador mantêm o mesmo “look and feel” independentemente da plataforma que, no caso de Java, não é necessário recomilar quando se muda de plataforma.

Esta comunicação apresenta uma aplicação para visualização interactiva de dados de grande volume desenvolvida no âmbito de um Trabalho Final de Curso da Licenciatura em Engenharia Informática do Instituto Superior Técnico [Caiado01].

O desenvolvimento desta nova aplicação para visualização, NISVAS (New Interactive System for Visual Analysis), constituiu um desafio importante. Isto deveu-se não só ao objectivo de desenhar e implementar uma interface simples de aprender e utilizar, como também porque a linguagem que veio a ser adoptada para realizar a sua implementação, Java3D, representa ainda hoje uma tecnologia emergente cuja evolução continua.

## 2. OBJECTIVOS

A aplicação de visualização desenvolvida, NISVAS, partiu da análise da funcionalidade apresentada por aplicações para visualização existentes, principalmente a aplicação ISVAS, e de inquéritos realizados junto de utilizadores que permitiram identificar muitos dos problemas sentidos por estes. Desta análise resultou um conjunto de objectivos centrados em:

- Portabilidade
- Interface com o utilizador
- Modularidade
- Desempenho
- Documentação

### 2.1 Portabilidade

A maioria das aplicações para visualização interactiva não é portátil e, em muitos casos, emprega normas da indústria com elevado peso histórico que limitam a sua disseminação e utilização. Por este motivo, definimos como objectivo que a aplicação a desenvolver fosse implementada de forma potenciar a sua portabilidade. Assim, logo à partida, o leque de opções foi reduzido ao emprego de ferramentas como C++, OpenGL, Java e Java3D.

### 2.2 Interface com o Utilizador

Um dos maiores problemas encontrados em aplicações de visualização interactiva refere-se às interfaces com o utilizador que, em geral, são pouco intuitivas, de difícil aprendizagem e utilização e, não raras vezes, pouco consistentes. Um exemplo disto é o caso da aplicação ISVAS que, mesmo para a execução de comandos simples, exige um conhecimento apreciável. Este visualizador apresenta ainda uma profusão de diálogos com o utilizador com muito pouca consistência e de diferente apresentação no caso de diálogos semelhantes. Entre outras coisas, os utilizadores inquiridos salientaram o facto de a interface desta aplicação exigir bastante memorização e de nela se

perderem muitas vezes, mesmo quando só pretendiam conhecer o estado da aplicação.

Uma situação que os utilizadores também apontaram como factor negativo foi a ordem inflexível de carregamento de ficheiros constituindo um conjunto de dados e de terem que explicitamente nomear todos os ficheiros a carregar.

Face aos comentários recolhidos junto dos utilizadores, estabelecemos então que a interface com o utilizador da nova aplicação teria que proporcionar diálogos simples e consistentes, fosse fácil de aprender e utilizar e, além disso, permitisse uma visão coerente do estado da aplicação. A interface deveria igualmente proporcionar atalhos (teclas de atalho e aceleradoras, por exemplo) e redundância na forma de realizar as tarefas, de modo a permitir o seu emprego quer por novos utilizadores, quer por utilizadores experientes.

A interface deveria também permitir a simplificação e a automatização de tarefas complexas, como o carregamento de conjuntos de dados por meio de um único comando.

### 2.3 Modularidade

Um outro objectivo fixado foi a modularidade da arquitectura. Com este objectivo pretendia-se uma identificação muito clara dos diversos módulos constituintes da aplicação e da sua funcionalidade, facilitar a reutilização de código, a sua manutenção e aumento de funcionalidade. Em relação à interface com o utilizador, a opção pela linguagem Java orientada para objectos pretende atingir os mesmos objectivos, além da segurança e robustez que proporciona.

### 2.4 Desempenho

O desempenho das aplicações para visualização é um factor crítico essencial a ter em conta na sua concepção e realização, dadas as exigências do processamento gráfico em termos de recursos. Isto significa que todas as opções a tomar, principalmente a solução para a arquitectura, deveriam ter em conta o seu impacto no desempenho e a necessidade de encontrar o equilíbrio entre facilidade de desenvolvimento, usabilidade da interface com o utilizador e o desempenho global.

### 2.5 Documentação

A maioria das aplicações para visualização interactiva é acompanhada por manuais extensos e complexos, de difícil leitura, e a ajuda on-line disponível é normalmente insuficiente. Deste modo, definiu-se como objectivo a atingir que a aplicação a desenvolver proporcionasse ajuda contextualizada, para além da documentação completa acessível a partir da aplicação, com vista a facilitar e diminuir o período de aprendizagem.

## 3. ARQUITECTURA

A arquitectura da aplicação NISVAS divide-se em dois subsistemas: a interface com o utilizador e o núcleo gráfico. A comunicação entre os dois subsistemas é realizada por módulos de comunicação, tal como a figura 1 apresenta.

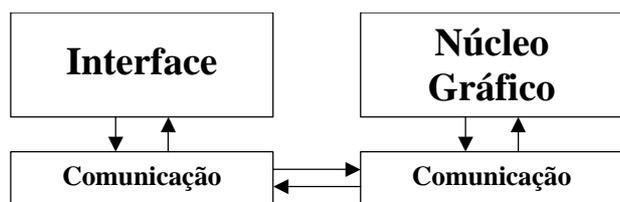


Figura 1 – Arquitectura geral da aplicação NISVAS.

O subsistema de interface é responsável pela interacção com o utilizador, pelo desencadeamento dos respectivos comandos, pela prevenção de erros e por transmitir ao utilizador todos os resultados não gráficos dos comandos executados.

O subsistema núcleo gráfico é responsável pela execução dos comandos gráficos e pela construção e disponibilização de todas as opções de visualização dos conjuntos de dados a visualizar. É também responsável pela criação e destruição das janelas de visualização gráfica e pela gestão de eventos ocorridos nessas janelas.

Esta solução para a arquitectura segue o paradigma da arquitectura cliente/servidor, em que a interface com o utilizador desempenha o papel de cliente e o núcleo gráfico o papel de servidor. A interface processa os eventos ocorridos nas suas componentes, enquanto o núcleo gráfico processa graficamente os pedidos que lhe chegam. A realização da comunicação e do protocolo de comunicação entre os dois subsistemas dependia da arquitectura adoptada.

### 3.1 Primeira Solução para a Arquitectura

Numa primeira aproximação, a arquitectura poderia consistir em dois subsistemas comunicando entre si por meio de sockets. Esta solução permitiria a portabilidade de um núcleo gráfico baseado em OpenGL e uma interface baseada em Java.

As vantagens desta solução consistiriam na portabilidade total (sem recompilação) da interface com o utilizador e na portabilidade do código fonte do núcleo gráfico proporcionada pela linguagem C++ e pela norma da indústria OpenGL. Esta solução exigiria uma definição muito clara do protocolo de comunicação dos dois subsistemas, mas permitiria a sua total modularidade.

A grande desvantagem apresentada por esta solução consistia no facto de a comunicação por meio de sockets ser difícil de portar, obrigando à reescrita dos módulos de comunicação. Mas, mais crítico ainda, o desempenho da aplicação seria afectado dado que se previa que a comunicação entre os dois subsistemas viesse a ser bastante intensa, o que foi confirmado em testes realizados com protótipos.

### 3.2 Solução Adoptada para a Arquitectura

A solução alternativa consistiria em implementar toda a aplicação em Java [Arnold96], com o núcleo gráfico empregando a API Java3D [Bouvier99]. Além do desafio constituído pelo emprego de uma tecnologia emergente, esta solução apresentava as vantagens de uma total portabilidade da aplicação (sem necessidade de recompilação),

ser totalmente orientada para objectos, facilitar a integração de novas funcionalidades e da simplificação do protocolo de comunicação entre a interface com o utilizador e o núcleo gráfico. A única desvantagem desta solução poderia residir no facto de a linguagem Java ser uma linguagem executada por uma máquina virtual e, portanto, poder apresentar um desempenho inferior em relação ao desempenho de código equivalente escrito em C ou C++.

A solução adoptada permite definir o núcleo gráfico como um objecto que realiza transparentemente toda a manipulação gráfica dos conjuntos de dados, disponibilizando para tal uma interface bem definida constituída por métodos e propriedades a que a interface com o utilizador acede para executar os comandos desencadeados pelo utilizador. Cada conjunto de dados é, por sua vez, um objecto cujos métodos são invocados pelo núcleo gráfico.

A plataforma de desenvolvimento adoptada foi um computador pessoal Pentium III a 500 MHz com 128 Mb de memória, operando sob Windows 98. As ferramentas de desenvolvimento adoptadas foram o JDK 1.2.2, Java3D 1.1.3, JavaHelp 1.1 e o ambiente de desenvolvimento Forte for Java CE 1.0.

## 4. FUNCIONALIDADE IMPLEMENTADA

A aplicação para visualização interactiva NISVAS opera sobre conjuntos de dados referenciados a geometrias. Os dados podem ser dos tipos escalar e vectorial. A referência geométrica dos dados segue o modelo dos métodos de simulação numérica por Elementos Finitos<sup>1</sup> que definem formas geométricas bi ou tridimensionais com base em malhas de nós colocados no espaço de simulação. A figura 2 apresenta alguns destes elementos, que podem ser contíguos ou não, conforme a figura 3 apresenta.

Os conjuntos de dados podem ser carregados e eliminados da aplicação, sendo possível copiá-los ou alterar o seu nome. Um conjunto de dados pode ser lido pela aplicação de uma só vez ou carregado ficheiro a ficheiro, podendo-se adicionar um novo ficheiro ao conjunto de dados a qualquer momento ou removê-lo do conjunto.

Quando um conjunto de dados é carregado, a aplicação apresenta a sua geometria e selecciona uma iluminação ambiente uniforme. Através da interface, o utilizador pode então alterar a cor dos objectos ou atribuí-la segundo funções de transferência que fazem corresponder a gama de valores de uma grandeza escalar a uma escala de cores.

O utilizador pode também seleccionar grandezas escalares e vectoriais do conjunto de dados, fazendo-as corresponder à cor e tamanho de símbolos ou à cor, intensidade e direcção de vectores referenciados à geometria. A geometria dos objectos pode ainda ser deformada designando uma grandeza vectorial como a deformação dos objectos.

<sup>1</sup> Isto não impede a visualização de resultados de simulações realizadas por outros métodos, como os métodos de Diferenças Finitas.

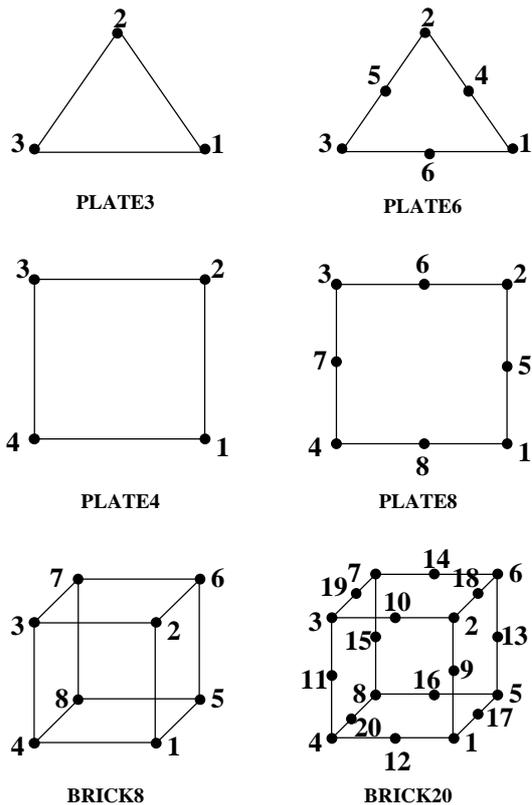


Figura 2 – Alguns elementos de malha empregues pela aplicação NISVAS, mostrando a localização e numeração de nós.

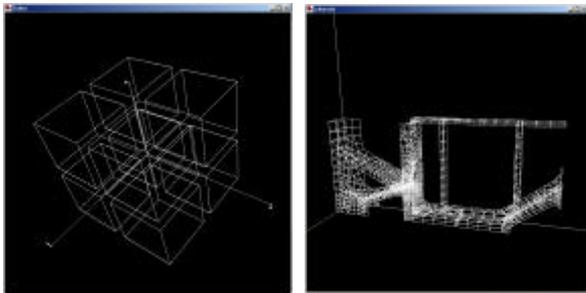


Figura 3 – Geometria de dois conjuntos de dados visualizados pela aplicação NISVAS.

A aplicação NISVAS permite a manipulação da iluminação, a criação de transparências com grau de transparência controlável e planos de corte (para análise do interior dos objectos) que é possível localizar precisamente.

Toda a informação sobre a interacção realizada com um conjunto de dados é guardada coerentemente, sendo possível armazenar externamente uma vista sobre um conjunto de dados. O carregamento posterior de uma vista implica o carregamento automático do respectivo conjunto de dados e a sua apresentação imediata com o mesmo mapeamento de grandezas, iluminação e vistas que possuía quando foi criada.

#### 4.1 Behaviour

A aplicação NISVAS recorre ao conceito de Behaviour do Java3D. Este conceito consiste na definição da uma conexão entre um estímulo (evento) e uma acção. O

visualizador NISVAS emprega não só Behaviours pré definidas em Java3D, como implementa novas Behaviours para realizar as operações de rotação, panning e zoom, quer através da interacção por meio de dispositivos apontadores como o rato, quer através do teclado.

#### 4.2 Mapeamento de Grandezas nos Objectos

O visualizador NISVAS permite fazer corresponder grandezas escalares e vectoriais à cor dos objectos, símbolos e vectores e controlar a cor dos objectos pré definidos das cenas, como luzes e fundos, especificando a cor por meio dos modelos RGB e HSV, conforme a preferência do utilizador. No caso de grandezas escalares, a correspondência é estabelecida entre a gama de valores da grandeza escalar e uma escala de cores que pode ser de quatro tipos: Physics (escala iniciada no azul e terminada no vermelho, idêntica ao espectro de cor), Hueramp (do vermelho ao vermelho, passando por todas as cores saturadas do modelo HSV), Grayscale (do preto ao branco, passando pelos cinzentos) e Hotiron (do vermelho ao branco, passando por laranja e amarelo).

A cor dos objectos pode igualmente ser definida por uma grandeza vectorial cujas componentes são as componentes RGB da cor a atribuir a cada um dos nós da malha.

A geometria dos objectos pode ser alterada ou deformada, bastando para isto adicionar às coordenadas dos nós da malha os valores de uma grandeza vectorial cujas componentes são interpretadas como os deslocamentos a aplicar aos respectivos nós nas três direcções do espaço. Estes deslocamentos podem ser ampliados ou reduzidos por meio de um factor de escala que é possível controlar interactivamente, podendo ainda ser animados se os deslocamentos forem identificados como pertencentes a momentos diferentes da simulação.

A visualização da geometria dos objectos definidos pelos conjuntos de dados, assim como dos objectos globais de cena (eixos coordenados, caixa envolvente, etiquetas, luzes e fundo de cena), é controlada pelo utilizador, podendo o utilizador escolher torná-los invisíveis para melhor observar outros objectos como símbolos e vectores. Normalmente, a geometria dos objectos é representada por polígonos orientados correspondendo às faces dos elementos da malha. As suas arestas podem também ser visualizadas por opção do utilizador.

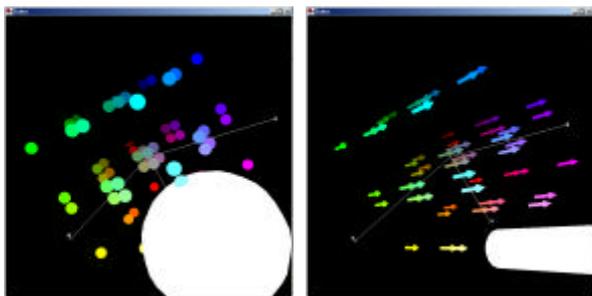
#### 4.3 Símbolos e Vectores

A aplicação NISVAS permite fazer corresponder símbolos e vectores a grandezas associadas aos nós da malha, tal como a figura 4 apresenta. Os símbolos podem ser bidimensionais (círculos, triângulos ou quadrados) ou tridimensionais (esferas, octaedros ou cubos).

Na representação de grandezas vectoriais por meio de vectores, estes podem ser de vários tipos como linhas simples, pirâmides facetadas, pirâmides de faces arredondadas ou cilindros com topos cônicos.

Aos símbolos e vectores é possível atribuir cores tal como para os objectos da geometria. A dimensão dos símbolos

e vectores pode reflectir grandezas escalares cujo factor de escala pode ser controlado pelo utilizador.



**Figura 4 – Representação de uma grandeza escalar por meio de símbolos e de uma grandeza vectorial por meio de vectores (a geometria foi ocultada para facilitar a visualização).**

#### 4.4 Objectos Globais

O visualizador NISVAS define um conjunto de objectos globais para as cenas. Estes objectos (eixos coordenados, caixa envolvente da geometria, etiquetas, cor de fundo, planos de corte e luzes) podem ser tornados visíveis ou invisíveis, ou, no caso de caixas envolventes, serem representados por meio de pontos, linhas e polígonos, consoante a preferência e a necessidade dos utilizadores.

#### 4.5 Luzes

Para além da luz ambiente cuja intensidade e cor podem ser controladas, o visualizador NISVAS permite ainda iluminar as cenas por meio de outras luzes colocadas em locais pré definidos. Estas luzes podem ser direccionais, pontuais ou do tipo foco. A cor e intensidade destas luzes são igualmente controláveis pelo utilizador.

O visualizador permite também controlar as propriedades cromáticas das superfícies dos objectos, como a sua difusibilidade e opacidade.

#### 4.6 Opções de Visualização

Além da funcionalidade que permite definir e localizar vários planos de corte para exame do interior dos objectos, o visualizador NISVAS apresenta ainda funcionalidade de controlar a forma de apresentação das imagens na janela de visualização, permitindo visualizar só as faces anteriores ou só as faces posteriores, ver a cena em todo o ecrã e ligar ou desligar a iluminação.

Por omissão, o visualizador NISVAS apresenta imagens correspondentes à projecção de perspectiva, mas o utilizador pode optar pela projecção paralela ou por vistas seleccionadas (frontal, planta ou alçado) em separado ou simultaneamente.

### 5. INTERFACE COM O UTILIZADOR

Para concretizar os objectivos anteriormente definidos e disponibilizar toda a funcionalidade descrita, a interface do visualizador NISVAS foi cuidadosamente concebida e realizada. A metodologia seguida consistiu na análise das tarefas a realizar pelo utilizador e na avaliação de protótipos, incluindo protótipos de baixa fidelidade, que permitiram escolher e validar soluções.

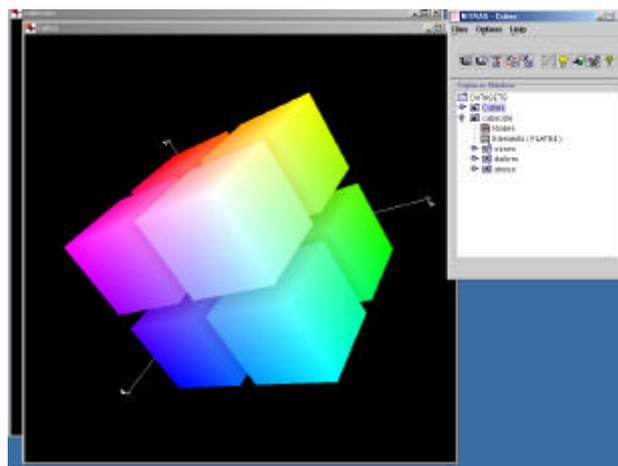
O objectivo pretendido era o de maximizar o emprego da manipulação directa dos objectos e proporcionar aos uti-

lizadores uma interface coerente e de aprendizagem rápida, em que os diálogos com o utilizador não distassem mais do que dois passos a partir da janela principal de diálogo, nem apresentassem simultaneamente mais do que cerca de seis opções [Mayhew92].

Pretendeu-se também proporcionar redundância das formas de acesso aos diálogos para que a interface se adaptasse tanto a utilizadores inexperientes ou pouco frequentes, como a utilizadores muito experientes na aplicação. Além disso, a interface deveria proporcionar ao utilizador uma visão clara do estado da aplicação que não o sobrecarregasse com demasiada informação, isto é, que para além do estado global da aplicação, os detalhes do estado da aplicação fossem apresentados de acordo com o contexto dos diálogos seleccionados.

Finalmente, a interface deveria facilitar a identificação das opções de comando por parte do utilizador.

Para atingir estes objectivos, a interface do visualizador NISVAS apresenta dois tipos de janelas: a janela principal, a partir da qual o utilizador dialoga com a aplicação e introduz comandos, e as janelas de visualização, uma por cada conjunto de dados. A figura 5 apresenta o aspecto de um ecrã onde se vê a janela principal do visualizador e duas janelas de visualização correspondendo a dois conjuntos de dados que estão a ser visualizados.



**Figura 5 – Ecrã mostrando a janela principal do visualizador NISVAS e a visualização de dois conjuntos de dados, cada um na respectiva janela de visualização.**

#### 5.1 Manipulação Directa na Janela de Visualização

Ao carregar um conjunto de dados, a aplicação NISVAS instancia imediatamente a janela de visualização do conjunto onde, também imediatamente, é apresentada a geometria correspondente ao conjunto de dados carregado<sup>2</sup>.

Os objectos presentes numa janela de visualização podem ser manipulados directamente deslocando o rato e premindo simultaneamente um dos seus botões. As opera-

<sup>2</sup> As restantes grandezas serão posteriormente mapeadas na geometria através dos diálogos disponíveis a partir da janela principal.

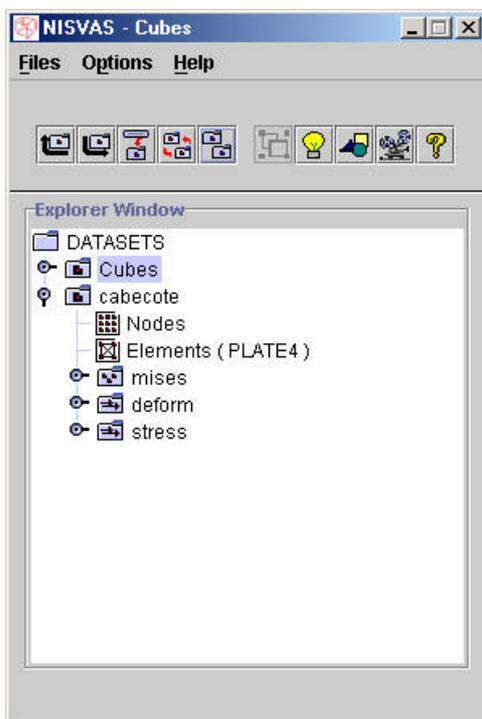
ções assim realizáveis são rotações dos objectos (botão esquerdo premido), zoom (premindo o botão central) e panning (botão direito). Como se trata de manipulação directa, as operações realizadas podem ser imediatamente desfeitas por deslocamento do rato na direcção oposta.

Todas as operações de rotação, zoom e panning realizadas numa janela de visualização podem ser desfeitas com um a única acção. Esta consiste em premir e largar o botão esquerdo do rato sem o mover.

## 5.2 Estado Geral da Aplicação

O estado geral da aplicação é apresentado ao utilizador através das janelas de visualização e da janela principal.

As janelas de visualização permitem que o utilizador avalie de forma imediata o estado da aplicação no que diz respeito à transformação de vista aplicada aos objectos de cada conjunto de dados e, em menor grau, do mapeamento das grandezas do conjunto, do controlo de luzes e dos objectos globais da cena.



**Figura 6 – Janela principal da aplicação NISVAS com barra de menu (topo), barra de ferramentas (no meio) e zona de exploração (na parte inferior) com dois conjuntos de dados.**

A janela principal (veja-se a figura 6) encontra-se dividida em três zonas: zona da barra do menu principal, zona da barra de ferramentas e zona de exploração. Esta última zona apresenta o estado da aplicação referente aos conjuntos de dados correntemente carregados.

O conjunto de dados correntemente activo é assinalado pela aplicação NISVAS na zona de exploração, rodeando o seu nome com uma caixa com cor de fundo invertida. Para seleccionar um conjunto de dados como o conjunto activo basta clicar sobre o seu nome na zona de exploração.

Por omissão, cada conjunto de dados é apresentado de forma expandida na zona de exploração, isto é, os nomes de todos os ficheiros contendo as grandezas do conjunto de dados são apresentados, tal como a figura 6 o faz para o conjunto *cabecote*. Esta apresentação repete-se para as grandezas que sejam definidas em mais do que um instante de tempo (mises, deform e stress, neste exemplo). É possível ocultar esta lista de nomes, como acontece com o conjunto *Cubes* na mesma figura, clicando no símbolo – localizado à esquerda do nome do conjunto. O símbolo passa então a + e a lista pode ser de novo apresentada, voltando a clicar sobre ele.

## 5.3 Adaptação ao Utilizador e Redundância

A redundância na emissão de comandos ou na selecção de diálogos é essencial para a adaptação de qualquer aplicação interactiva às preferências, grau de perícia dos utilizadores e contexto das tarefas que pretendem realizar<sup>3</sup>.

Na aplicação NISVAS, a redundância permite entre 4 a 5 formas distintas para aceder aos comandos da interface. A selecção de comandos pode ser realizada por meio de

- Menu principal
- Barra de ferramentas
- Menus PopUp na zona de exploração
- Teclas de atalho
- Teclas aceleradoras

A operação de eliminação de um conjunto de dados, por exemplo, pode ser realizada através do menu principal (Files->Unload Dataset), da barra de ferramentas (segundo ícone a contar da esquerda), por menu PopUp (clicando o botão direito do rato sobre o nome do conjunto na zona de exploração e seleccionando Unload Dataset), por teclas de atalho de navegação no menu principal (Alt+F, U) ou por teclas aceleradoras (CTRL-U).

O utilizador inexperiente pode assim realizar operações passo a passo através do menu principal, enquanto um utilizador experiente poderá realizar as mesmas operações através da barra de ferramentas ou de teclas aceleradoras.

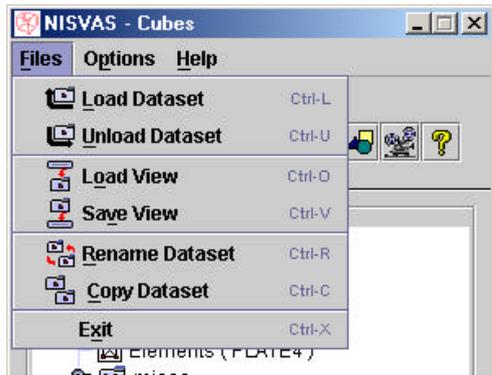
## 5.4 Acesso aos Diálogos

Como foi dito anteriormente, pretendeu-se proporcionar o acesso aos diálogos de comando através de um mínimo de acções de selecção, com o intuito de evitar que os utilizadores tivessem que construir modelos mentais complexos e difíceis de aprender, reduzindo assim também o número de erros na selecção.

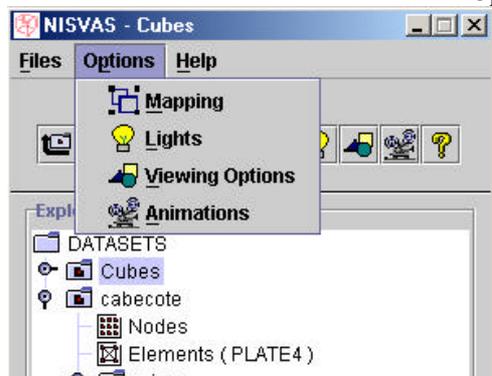
O estudo de análise de tarefas, reflectido na estrutura do menu principal, permitiu a adopção de uma estrutura hierárquica com apenas dois níveis de profundidade e um número reduzido de opções em cada nível. Este esforço de sistematização foi particularmente impulsionado pela

<sup>3</sup> Evitando, por exemplo, os incómodos devidos à necessidade de alternar entre o uso do teclado e do rato.

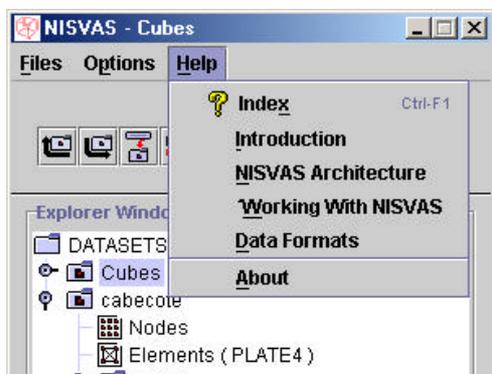
solução adoptada para o diálogo de mapeamento que apresentaremos em detalhe mais à frente.



Files



Options



Help

Figura 7- Opções disponíveis nos grupos do menu principal.

Assim, a hierarquização das tarefas a realizar permitiu dividi-las em três categorias:

- Files – carregamento e eliminação de conjuntos de dados, carregamento e armazenamento de vistas e saída da aplicação.
- Options – mapeamento de grandezas e controlo da iluminação, das opções gerais de visualização e das animações.
- Help – sistema de ajuda e documentação, constituída por índice, introdução, arquitectura, como utilizar a aplicação NISVAS e formato dos dados de entrada.

O ícone mais à esquerda na barra de ferramentas também proporciona o acesso directo à documentação.

A figura 7 apresenta as opções disponíveis a partir de cada grupo do menu principal, em que é claro o pequeno número de opções de cada grupo.

A análise de tarefas do grupo mais complexo, o grupo Options, permitiu separá-las em quatro diálogos distintos: Mapping, Lights, Viewing Options, e Animations, apresentados pelas figuras 8 , 9 , 10 e 11 , respectivamente.



Figura 8 – Diálogo de mapeamento (Mapping).

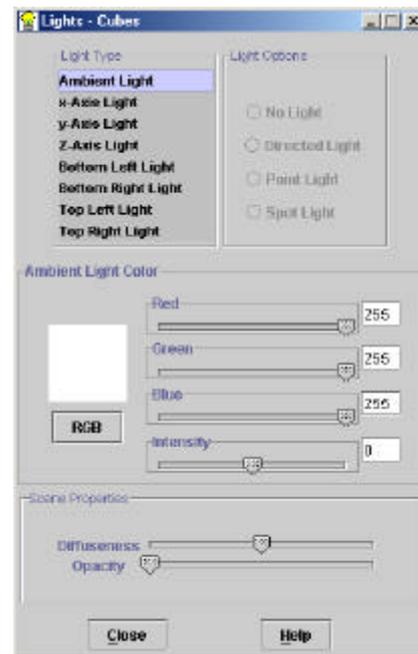


Figura 9 – Diálogo de controlo da iluminação (Lights).



Figura 10 – Diálogo das opções gerais de visualização (Viewing Options).

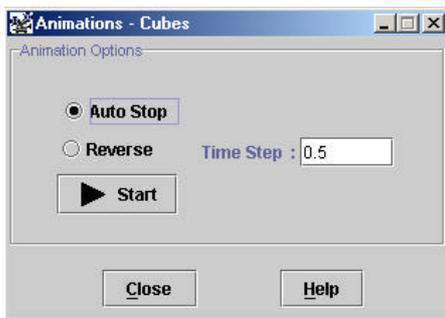


Figura 11 – Diálogo de controlo de animações (Animations).

À excepção do diálogo de mapeamento, a análise de tarefas permitiu simplificar consideravelmente o conteúdo de todos os diálogos, tal como é patente das figuras. Em todos os diálogos existe o botão Close para encerramento dos diálogos, à esquerda e em baixo, e, à direita, o botão Help que proporciona o acesso à ajuda contextualizada referente ao diálogo corrente.

### 5.5 Diálogo de Mapeamento

O diálogo de mapeamento é talvez o diálogo mais importante de toda a interface do visualizador NISVAS pois é a partir deste diálogo que o utilizador selecciona o mapeamento das grandezas dos conjuntos de dados em cores, símbolos, vectores e deslocamentos e especifica as respectivas propriedades. Este diálogo permite controlar também o aspecto e apresentação dos objectos globais pré definidos (caixa envolvente da geometria, rótulos e etiquetas, eixos coordenados e cor de fundo).

A análise das tarefas de mapeamento permitiu identificar que estas tarefas incidem sobre quatro grupos de objectos: geometria, símbolos, vectores e objectos globais. Existem algumas semelhanças entre as operações que é possível realizar sobre estes quatro grupos de objectos, mas existem igualmente bastantes diferenças.

A concepção deste diálogo deveria permitir um diálogo muito claro para o utilizador. Assim, adoptou-se por uma apresentação do tipo Tabbed Panel, disponibilizada pelo API da linguagem Java, em que as opções se encontram separadas por secções identificadas através de etiquetas separadoras e em que todas as subsecções da secção corrente são visíveis em simultâneo. Deste modo, evita-se a necessidade de o utilizador ter que memorizar a organização hierárquica deste diálogo. Esta característica é reforçada pelo facto de o número de subsecções de cada secção (4, no máximo) ser muito reduzido.

A tabela 1 apresenta as divisões em secções e subsecções resultante da análise de tarefas. As figuras 12 a 15 apresentam diálogos de mapeamento para cada uma das secções.

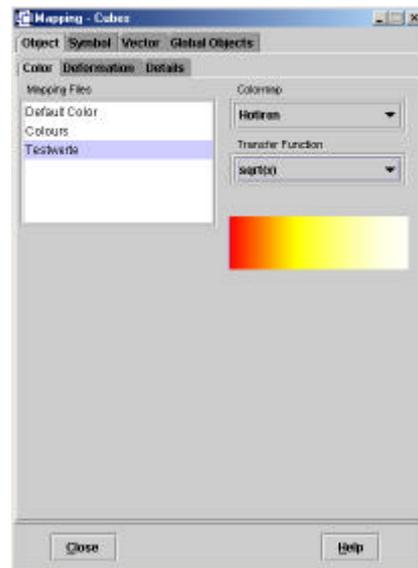


Figura 12 - Diálogo de mapeamento na geometria dos objectos.

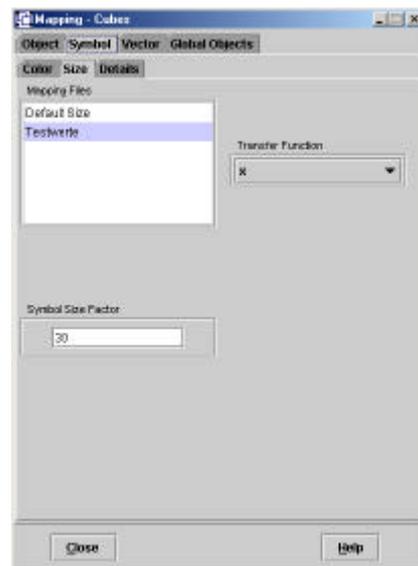


Figura 13 – Diálogo de mapeamento de grandezas em símbolos.

Grupo	Object	Symbol	Vector	Global Objects
Subsecção	Color	Color	Color	Bounding Box
	Deformation	Size	Length	Labels
	Details	Details	Details	Coordinate Axis
				Background Color

Tabela 1 – Secções e subsecções do diálogo de mapeamento.

los.

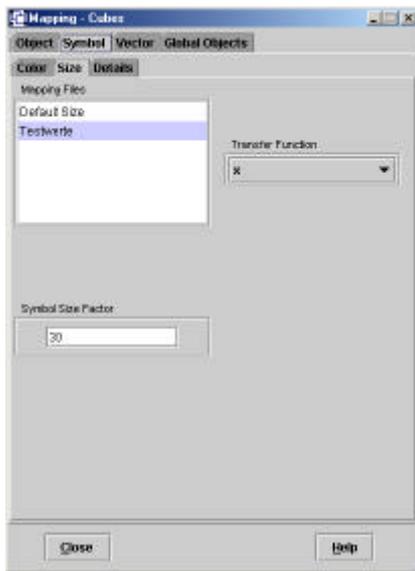


Figura 14 – Diálogo de mapeamento de grandezas em vectores.

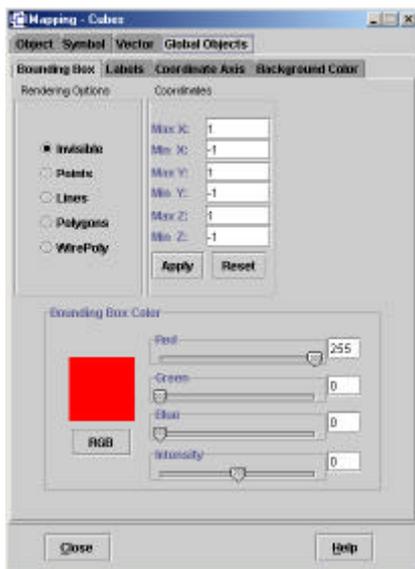


Figura 15 – Diálogo de mapeamento e controlo dos objectos globais.

## 6. EXEMPLO DE APLICAÇÃO

Como exemplo, apresentamos de seguida o caso do estudo da deformação de uma peça de protecção de carruagens ferroviárias contra choques. A geometria desta peça é constituída por 1452 nós a que correspondem 1545 elementos planares do tipo PLATE4 (veja-se a figura 2 ). Além da geometria, o conjunto de dados contém a informação vectorial da deformada da peça e do estado de tensão, conjuntamente com o valor de teste (escalar) para o critério de von Mises, em todos os nós da malha. Estes dados estão disponíveis para 21 momentos do processo de deformação da peça, o que torna este conjunto de dados bastante complexo.

A figura 16 apresenta a deformação da peça, obtida pelo mapeamento dos valores do campo da sua deformada como deformação (ampliada) da geometria, seguida pela animação. A peça é representada por meio de linhas (wireframe) para melhor visualização dos resultados.

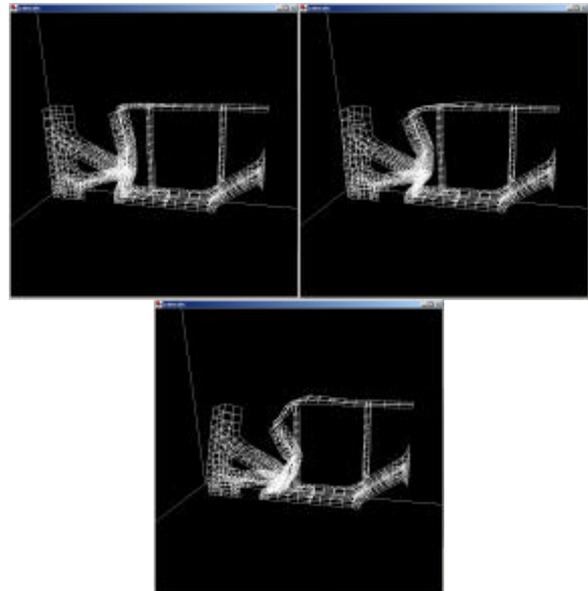


Figura 16 – Deformação de uma peça sob impacto violento em três momentos do processo.

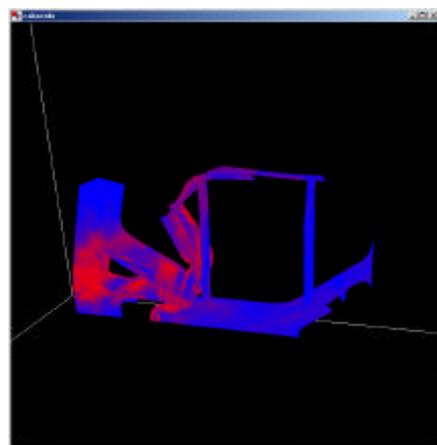


Figura 17 – Apresentação da deformação de uma peça simultaneamente com o módulo do estado de tensão representado por cor.

A figura 17 apresenta a peça num dos momentos da deformação, permitindo apreciar simultaneamente a deformação e o estado de tensão local, traduzido em cor atribuída segundo uma escala do tipo Physics.

## 7. CONCLUSÕES

O desenvolvimento de raiz da aplicação NISVAS para a visualização de dados de grande volume resultantes da simulação de fenómenos complexos bi e tridimensionais constituiu um projecto complexo que requereu cuidados extremos na concepção da arquitectura, na selecção das ferramentas de desenvolvimento e, principalmente, na concepção e realização da interface ao utilizador.

Com efeito, o trabalho realizado enfrentou vários desafios importantes como o desenvolvimento segundo conceitos e paradigmas actuais, nomeadamente a Concepção Orientada para Objectos, e o emprego de tecnologias emergentes, como é o caso da tecnologia Java3D. É conveniente realçar que, durante a realização deste projecto, esta tecnologia foi alvo de constantes actualizações, razão pela qual houve que estar atento à publicação de novas versões e, por vezes, reescrever parte do código fonte para poder usufruir da novas funcionalidades entretanto disponibilizadas ou eliminar código escrito para ultrapassar erros (bugs) presentes em versões anteriores das ferramentas de Java e Java3D.

Mas foi ao nível da concepção e realização da interface com o utilizador que se colocaram os maiores desafios. Com efeito, o sucesso ou insucesso de uma aplicação interactiva para visualização, como a aplicação NISVAS, depende muito mais da interface com o utilizador e da facilidade com que o utilizador a emprega, do que da funcionalidade da aplicação. A metodologia de trabalho empregue, baseada na análise de tarefas e na construção e avaliação de protótipos, foi determinante para os excelentes resultados obtidos.

Destes resultados, é de salientar o nível de simplificação da hierarquia de comandos, conseguido sem diminuir a funcionalidades pretendida da aplicação, e que permite ao utilizador ter uma visão permanente das opções ao seu dispor simultaneamente com a possibilidade de seleccionar comandos pela forma que melhor se adapte ao seu perfil de utilizador (adaptação e redundância). Estas características da interface permitem que a utilização da aplicação NISVAS seja simultaneamente simples, poderosa e de fácil aprendizagem, não requerendo a construção de modelos mentais complexos da aplicação por parte do utilizador.

As reacções positivas registadas junto dos utilizadores confirmaram que os objectivos delineados no início deste trabalho tinham sido efectivamente alcançados. Os utilizadores realçaram as potencialidades disponibilizadas pela aplicação NISVAS, a facilidade de uso da sua interface, além do bom desempenho da aplicação. Algumas das reacções recolhidas salientaram que, à partida, nunca teriam esperado que uma aplicação tão complexa e desenvolvida em Java3D pudesse apresentar as características e nível de desempenho verificados.

## 7.1 Trabalho Futuro

As reacções dos utilizadores e avaliadores nos testes e demonstrações realizadas constituem um encorajamento para continuar o desenvolvimento da aplicação de visualização NISVAS. Com efeito, a arquitectura e as soluções encontradas constituem uma excelente base de trabalho para futuros desenvolvimentos.

Um destes desenvolvimentos incidirá no suporte a outros formatos para os dados de entrada que não o formato actual. Os utilizadores apontaram a conveniência em permitir a entrada de dados em formatos normalizados, como formatos neutros ou outros, como o formato HDF. Foi também identificada a necessidade de poder comunicar directamente dados à aplicação NIVAS a partir das simulações (visualização on-line).

O trabalho futuro incidirá igualmente no aumento de funcionalidade de animação de conjuntos de dados, com o mapeamento temporal de mais do que uma grandeza e na animação de sistemas de partículas para visualização de escoamentos de fluidos, estacionários ou transitórios, como, por exemplo, escoamentos atmosféricos, escoamentos em rios, estuários e em alto mar, com transporte de partículas e poluentes ou túneis de vento virtuais.

Um outro desenvolvimento programado consistirá na manipulação directa das grandezas constituintes dos conjuntos de dados para, através da metáfora da calculadora de bolso, combinar, por exemplo, grandezas escalares em grandezas vectoriais, escalar grandezas ou aplicar-lhe transformações.

## 8. REFERÊNCIAS

- [Arnold96] K. Arnold, J. Gosling, The Java Programming Language, Addison-Wesley, 1996.
- [Bouvier99] D. J. Bouvier, Getting Started with Java3D API, Sun Microsystems Inc., 1999.
- [Caiado01] B. Caiado e L. Correia, Visualização de Dados de Grande Volume, Trabalho Final de Curso, Licenciatura em Engenharia Informática, Instituto Superior Técnico, 2001.
- [Haase95] H. Haase e outros, ISVAS 3.2 Interactive System for Visual Analysis User's Guide, Fraunhofer Institute for Computer Graphics, 1995.
- [Karlsson93] K. Karlsson, ISVAS 3.1 Interactive System for Visual Analysis User's Guide, Fraunhofer Institute for Computer Graphics, 1993.
- [Karlsson94] K. Karlsson, Ein Interaktives System zur Visuellen Analyse von Simulationsergebnissen, PhD diss., THD, Darmstadt, 1994.
- [Mayhew92] D. J. Mayhew, Principles and Guidelines in Software User Interface Design, Prentice Hall, 1992.
- [Preece98] J. Preece, Human Computer Interaction, Addison-Wesley, 1994.
- [Shneiderman98] B. Shneiderman, Designing the User Interface, Addison-Wesley Longman, 1998.