

Compartilhamento de Dados no Sistema de RV CyberMed

Ícaro L. L. da Cunha^{1*}, Bruno S. Monteiro¹, Ronei M. Moraes², Liliane S. Machado¹

¹Departamento de Informática e ²Departamento de Estatística
Universidade Federal da Paraíba (UFPB)
Cidade Universitária s/n – 58059-900 – João Pessoa – PB – Brazil

*Aluno de Iniciação Científica financiado pelo CNPq.

ivellius@yahoo.com.br, bruno84@gmail.com, ronei@de.ufpb.br,
liliane@di.ufpb.br

Abstract. *The integration of data in a Virtual Reality System can depend on a Data Structure (DS) capable of making available the sharing of information between the routines of the system. This DS must be implemented in order to prevent complex searches and repetitive calculations for these routines. The efficient access to information contained in the DS is important to guarantee the real-time feature of the system. This work presents the conception of a DS based on corner-table and its application in the CyberMed system.*

Resumo. *A integração de dados em um Sistema de Realidade Virtual pode depender de uma estrutura de dados (ED) capaz de disponibilizar o compartilhamento de informações entre as rotinas do sistema. Esta ED deve ser implementada de modo a evitar buscas complexas e cálculos repetitivos por estas rotinas. O acesso eficiente das informações contidas na ED é importante para a garantia de tempo-real do sistema. Este trabalho apresenta a concepção de uma ED baseada em corner-table e sua aplicação no sistema CyberMed.*

1. Introdução

Uma Estrutura de Dados (ED) é um conjunto de dados e métodos eficientes de acesso a estes. A eficiência de uma ED está relacionada à não ambigüidade e à manutenção da integridade de suas informações, bem como à rapidez do acesso às informações contidas na ED [Tenenbaum et al. 1995]. Esta eficiência torna-se possível quando a construção da ED passa por um bom planejamento, ou seja, a ED deve levar em conta características particulares do sistema. Dessa forma, o desempenho de uma aplicação também dependerá da implementação da ED.

Sistemas de Realidade Virtual (RV) são sistemas em que o desempenho é um elemento crítico devido à sua característica de tempo-real. Este fato torna-se mais evidente se o sistema tratar de rotinas que abordem diferentes modos de interação ou explorem diferentes sentidos humanos através de dispositivos específicos. Nestes casos, cada uma das diferentes rotinas necessitará de informações relacionadas aos objetos pertencentes ao ambiente virtual para sua execução [Machado e Moraes 2005]. Por esta razão, é necessário utilizar uma ED que seja capaz de armazenar os dados dos objetos, bem como informações derivadas destes. Para evitar cálculos repetidos de informações similares e necessárias a diferentes rotinas durante a execução, deve-se acrescentar na ED estes resultados calculados aos dados originais dos objetos [Bryson 1996]. Desse

modo, o compartilhamento de dados via uma única ED permitirá a integração entre as rotinas do sistema.

Este trabalho apresenta o levantamento de requisitos, planejamento, concepção, implementação e resultados do desenvolvimento de uma ED para integração do acesso a dados por diversas rotinas de um sistema de RV. Como aplicação desta ED, tomou-se como base o sistema CyberMed. O CyberMed é um sistema de RV baseado em *software* livre que tem como característica apoiar o ensino e treinamento médico. Entre suas rotinas, o CyberMed aborda: visualização tridimensional de objetos gráficos; uso de modelos realistas gerados a partir da reconstrução tridimensional de imagens médicas; interação espacial com sensação de toque (usando dispositivos hápticos [Burdea e Coiffet 2005]); deformação interativa dos objetos tocados; compartilhamento visual para o caso de uso em sala de aula; e supervisão/avaliação das ações do usuário [Machado et al. 2004]. Todas estas rotinas necessitam de informações derivadas dos dados dos objetos utilizados pelo sistema. Uma vez que estas informações seriam calculadas e geradas durante a execução da simulação, o seu uso por diferentes rotinas resultaria em cálculos repetitivos de informações similares, o que seria bastante custoso computacionalmente. Para poder disponibilizar os dados originais dos objetos e as informações derivadas necessários ao CyberMed, pensou-se em uma ED capaz de compartilhar todas as informações entre as rotinas do sistema.

A motivação principal deste trabalho é, portanto, conceber uma ED capaz de armazenar dados diversos relacionados a objetos gráficos e possibilitar um acesso compartilhado e rápido destes por diferentes rotinas de um sistema de RV, garantindo assim a característica de tempo-real do sistema. Toda concepção da ED foi baseada em *corner-table* [Vieira et al. 2004] por esta apresentar uma abordagem que facilita o uso de malhas triangulares e seus relacionamentos para representar a superfície de um objeto gráfico 3D. Como aplicação, esta ED foi utilizada no sistema CyberMed.

2. ED em Sistemas de RV

Uma ED pode ser criada de variadas maneiras. O modo como é feita a sua implementação depende da necessidade do sistema em que ela está sendo implementada.

No sistema de simulação de cirurgia crânio-facial apresentado por Münchenberg et al. (2000), a ED escolhida é em forma de matriz tridimensional. Os dados gerados pelo sistema são armazenados na matriz 3D para criar um modelo háptico e sobre este modelo pode-se identificar posições do objeto e do dispositivo. Para efetuar os cálculos sobre o modelo háptico é criada uma lista de objetos sobre a qual são realizados os cálculos. Os cálculos gerados pela interação no sistema são feitos através do uso de máscaras matriciais. Entre suas funcionalidades, o sistema apresenta rotinas de visualização, de detecção de colisão e de interação háptica que utilizam os dados da ED. De modo análogo, o sistema de simulação de inserção de agulha e implantação de partículas radioativas para tratamento de células cancerígenas [Alterovitz et al. 2003] utiliza duas EDs matriciais: uma para representar a superfície de simulação e a outra para armazenar os dados da imagem. Neste caso, a simulação é feita dentro de uma área bidimensional. A ED escolhida para este sistema é em forma de uma malha triangular 2D, onde os vértices dos triângulos podem se locomover no plano de acordo com a força aplicada sobre eles. Isso caracteriza uma deformação da superfície, a malha deformada

representará então a deformação da imagem contida na outra matriz. O sistema apresenta as rotinas de visualização, detecção de colisão e deformação.

Outro sistema de simulação interativa foi apresentado por Onishi et al. (2003) e visa observar o crescimento de árvores. Este sistema utiliza uma ED para armazenar *L-Strings* para realizar uma modelagem de uma árvore utilizando o sistema *L-System* [Prusinkiewicz and Lindenmayer 1990]. O *L-System* é um sistema que cria uma *string* de símbolos, *L-Strings*, adaptando regras de produção para o símbolo inicial (axioma) visando gerar as formas dos modelos de árvores. Neste sistema, a estrutura das *L-Strings* é construída hierarquicamente para aumentar a eficiência da geração dos modelos da árvore.

Devido aos requisitos críticos das rotinas do sistema CyberMed, a serem apresentadas nas sessões seguintes, este sistema serviu como base para a implementação da ED baseada em *corner-table*. Uma *corner-table* é uma ED capaz de armazenar objetos gráficos formados por malhas triangulares que possibilita usufruir das vantagens que o uso de uma malha triangular apresenta para as rotinas do sistema.

2.1. Corner-Tables

Uma *corner-table* (Tabela de Cantos) é uma ED concisa para representar malhas triangulares. Nela é usado o conceito de cantos para representar a associação de um triângulo a um de seus vértices ou, equivalentemente, associar um triângulo ao bordo oposto a um canto [Rossignac et al. 2001][Vieira et al. 2004].

Nessa ED, os cantos, os vértices e os triângulos são indexados por números não negativos. Cada triângulo é definido por três cantos consecutivos que definem sua orientação, a indexação dos cantos é diferente para cada triângulo. Por exemplo, na Figura 1, os cantos 0, 1 e 2 correspondem ao primeiro triângulo (triângulo 0), os cantos 3, 4 e 5 correspondem ao segundo triângulo (triângulo 1) e assim sucessivamente. Como consequência disso, um canto c é associado ao seu triângulo t por $t = c \text{ mod } 3$. A *corner-table* representa a geometria de uma superfície por associar cada canto c ao seu índice de vértice geométrico.

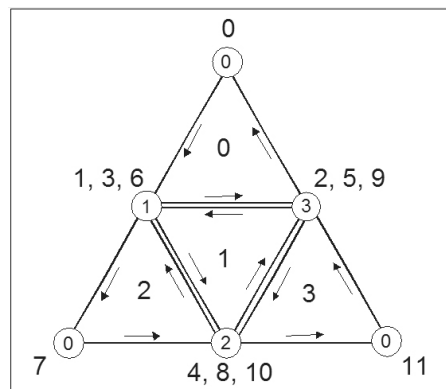


Figura 1. Vértices, cantos e faces de um tetraedro achatado.

Assumindo a orientação anti-horária de construção dos triângulos, para cada canto c , o $prox(c)$ e o $ante(c)$ são cantos de seu triângulo obtidos através do uso de das seguintes equações: $prox(c) = (c+1) \text{ mod } 3$ e $ante(c) = (c+2) \text{ mod } 3$. Estas equações tornam-se importantes na obtenção de dados dos vértices vizinhos dentro de um mesmo

triângulo. Pelo fato do armazenamento dos pontos seguirem o sentido de construção do triângulo, pode-se saber qual vértice vizinho está sendo acessado.

A adjacência de bordos entre triângulos vizinhos é representado por associar a cada canto c seu canto oposto $O[c]$ que tem a mesma borda oposta. A Figura 1 demonstra os vértices e cantos opostos para o caso de um tetraedro e a Tabela 1, uma lista dos dados deste tetraedro. É importante destacar que como o tetraedro é um objeto gráfico sólido criado por malhas triangulares, todo canto tem associado a ele três faces triangulares e conseqüentemente três cantos opostos.

Tabela 1. Lista de dados do tetraedro.

Canto	Vértice	Triângulo	Oposto
0	0	0	2
1	1	0	2
2	3	0	2
3	1	1	0
4	2	1	0
5	3	1	0
6	1	2	3
7	0	2	3
8	2	2	3
9	3	3	1
10	2	3	1
11	0	3	1

3. Compartilhamento de Dados no Sistema CyberMed

O CyberMed é um sistema de RV que suporta aplicações para ensino e treinamento médico. Exploração interativa da anatomia humana e simulação realista de procedimentos médicos são algumas de suas aplicações [Machado et al. 2004]. As principais características do CyberMed são: visualização 3D de objetos gráficos; uso de modelos realistas gerados por reconstrução 3D de imagens médicas; interação espacial com toque, usando dispositivos hápticos; deformação interativa de objetos; compartilhamento visual no caso do uso do sistema em sala de aula; e supervisão e avaliação a partir das ações do usuário.

Os objetos gráficos do CyberMed são formados por malhas triangulares. O uso de faces triangulares para representar as superfícies é mais conveniente que o uso de polígonos arbitrários. Isto é verdade por causa da simplicidade dos triângulos que, por oferecerem vantagens planares, são sempre curvas poligonais planas e tem coordenadas lineares associadas a eles [Gomes e Velho 1998]. Um objeto gráfico pode se apresentar em forma de uma camada ou de múltiplas camadas. No caso de um objeto com múltiplas camadas, cada camada tem associada a ela dados independentes dos dados das outras camadas.

Cada objeto usado pelo CyberMed tem uma série de propriedades físicas e topológicas associadas a ele, essas propriedades são utilizadas pelas rotinas do CyberMed para deformações, toque ou mesmo iluminação. Para tornar essas informações únicas, organizadas e acessíveis para as rotinas do sistema, elas devem estar concentradas em uma única ED que garanta o acesso rápido e eficiente a elas.

Como exemplo da necessidade de acesso a dados em um sistema de RV, podemos citar o caso em que um usuário interage com o ambiente e muda a posição do objeto apontador (através do uso de um mouse ou dispositivo háptico). Para realizar essa interação, deve existir uma rotina que verifica se houve colisão deste objeto com um outro objeto do ambiente 3D. Para este teste de colisão, é necessário identificar a posição do objeto apontador em relação aos objetos da cena. Caso uma colisão aconteça rotinas de deformação, toque, visualização e avaliação são ativadas ao mesmo tempo para realizarem os cálculos necessários e poderem dar um retorno imediato à interação do usuário. Para isto acontecer, muitas informações sobre os objetos da cena são necessárias. Essas informações podem ser: referências a vértices vizinhos de um dado vértice e coeficientes de elasticidade para cálculos de deformação e retorno de força; vetores normais de vértices e triângulos (incluindo os triângulos vizinhos) para cálculos de iluminação. Todas essas informações poderiam estar integradas em um ED. A Figura 2 apresenta como ocorre essa associação de uma ED com rotinas de visualização, detecção de colisão, deformação, interação háptica, interação 2D e avaliação no sistema CyberMed.

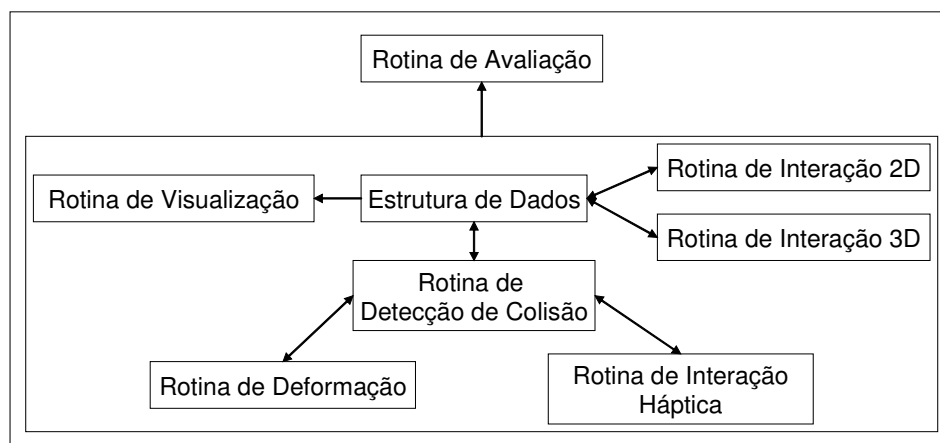


Figura 2. Dependência das rotinas do CyberMed em relação à ED.

Cada uma dessas rotinas precisa obter informações sobre o objeto gráfico dentro da ED. A rotina de visualização precisa de dados como a seqüência de construção dos triângulos, as coordenadas dos seus vértices, os valores das variáveis das transformadas geométricas executadas sobre o objeto gráfico, o vetor normal de cada vértice e o sentido de construção de cada triângulo que forma um objeto. As rotinas de interação 2D e 3D precisam do acesso aos valores de transformadas geométricas para poder atualizá-las. A rotina de detecção de colisão necessita saber das coordenadas dos vértices e dos valores de transformadas geométricas. A rotina de interação háptica necessita saber em qual local na superfície houve a colisão, quais as constantes elásticas do objeto tocado e a massa dos pontos. A rotina de deformação precisa saber de todas as informações que a rotina de interação háptica precisa além do acesso aos vértices

vizinhos de um dado vértice. A rotina de avaliação precisa analisar vários dados calculados na simulação, dados vindos da ED, rotina de deformação, interação háptica e detecção de colisão. Da ED, a rotina de avaliação precisa dos valores das variáveis das transformadas geométricas. Então, não somente pontos e polígonos do objeto são necessários para armazenamento na ED, mas o sistema precisa também que sejam guardados também valores relacionados a: triângulos vizinhos de cada triângulo, vértices vizinhos de cada vértice, cor do objeto gráfico, vetores normais dos polígonos e triângulos, entre outros, para uma execução efetiva dos cálculos das rotinas. Estes valores só precisam ser obtidos ou calculados uma vez durante e armazenados dentro da ED. Nota-se então, que é de fundamental importância que a ED possibilite o acesso direto aos dados originais dos objetos 3D e também aos dados calculados relacionados a esses objetos. Qualquer atraso feito durante o acesso aos dados necessários para estas rotinas do CyberMed (como buscas muito longas e referências constantes entre estruturas de dados diferentes) pode ser fatal para a característica de tempo-real do sistema.

4. Geração da ED

O uso de um objeto 3D pelo CyberMed é feito a partir da leitura de um arquivo fonte de um modelo que contem as descrições do objeto. Os dados do objeto são armazenados em uma ED baseada em *corner-table*. Os dados originais são usados para calcular outros dados necessários para o processamento do sistema como: vetores normais dos vértices e triângulos, constantes de elasticidade das ligações dos vértices (arestas dos triângulos), a massa dos pontos, e outras constantes.

A ED concebida é baseada em uma *corner-table*, mas apresenta algumas modificações: dados como coordenadas e referência a triângulos vizinhos são acessados mais diretamente dentro do seu próprio triângulo (diferentemente de uma lista de pontos seqüenciados) sem necessidade de utilizar índices numéricos; o armazenamento de mais dados é aplicado; e a ED está referenciando triângulos vizinhos de um dado triângulo e vértices vizinhos de um dado vértice, deste modo os dados dos respectivos vizinhos são acessados mais diretamente sem ter que criar uma rotina de busca para obter esse acesso (geralmente, essas buscas são muito custosas, haja visto que a capacidade de armazenamento da ED pode chegar aos milhares de triângulos e vértices). Estas mudanças são necessárias para possibilitar acessos mais rápidos a informações integradas, melhorando o desempenho do CyberMed como um todo.

Outra mudança bastante significativa da ED, se comparada à *corner-table*, é que são geradas três listas diferentes: uma lista de triângulos, onde cada um terá o acesso por ponteiro aos dados de três vértices dentro de uma lista de vértices, e cada vértice terá associado a ele uma lista de referências aos seus vértices vizinhos. Essa mudança se faz necessária, pois no caso da execução de uma deformação da superfície do objeto gráfico, por exemplo, a partir do ponto onde houve colisão, vários pontos ao redor deste poderão se locomover. Então, como o ponto em que houve esta colisão é um vértice, será necessário procurar dentro da lista de vértices pelos seus vértices vizinhos para poder fazer a deformação. Para evitar buscas muito longas e cálculos desnecessários, propos-se a implementação de uma lista de referências aos vértices vizinhos para cada vértice do objeto gráfico.

O arquivo fonte suportado pela ED deve estar no formato VRML 2.0. Este formato foi escolhido pois seus dados são do tipo texto, os objetos podem apresentar

suporte para malhas triangulares, o armazenamento dos dados geométricos de um modelo é na forma de lista de pontos, lista de polígonos e cores. Os outros dados como vetores normais e vizinhos são calculados quando a ED está sendo construída. Duas partes do arquivo VRML que devem ser destacadas: a lista de pontos e a lista de polígonos (no caso dos objetos gráficos do CyberMed, uma lista de triângulos). A lista de pontos apresenta-se como uma seqüência de coordenadas x, y e z dos pontos. A lista de polígonos apresenta uma referência numérica para cada vértice do polígono e um valor que representa o sentido de construção deste. A referência numérica apresenta-se de acordo com a seqüência na lista de pontos, ou seja, se o valor de uma determinada referência for 0 será o primeiro ponto da lista pontos, se for 1 será o segundo e assim por diante. Para especificar o sentido de construção do triângulo o valor de representação de sentido pode ser -1 para indicar sentido de construção anti-horário ou para indicar sentido de construção horário.

No início da execução do CyberMed, arquivos de cada modelo são lidos e seus dados são armazenados. Posteriormente, os dados são utilizados para calcular os triângulos vizinhos (um vizinho relacionado a cada vértice do triângulo) e os vetores normais dos triângulos. A partir destes, os vértices vizinhos e os vetores normais dos vértices são calculados para cada vértice da lista de vértices.

A ED é composta por três tipos de listas dinâmicas: lista de triângulos, lista de pontos e listas de vértices vizinhos (como mostra a figura 3). A lista de triângulo armazena: três ponteiros para seus vértices dentro da lista de vértices, seu vetor normal, o sentido de construção, três referências para os triângulos vizinhos e um ponteiro para o próximo triângulo da lista. A lista de vértices armazena: suas coordenadas x, y e z, seu vetor normal, um ponteiro para a sua lista de vértices vizinhos e um ponteiro para o próximo vértice da lista. As listas de vértices vizinhos armazenam: a referência para o vértice vizinho e um ponteiro para o próximo vértice vizinho da lista.

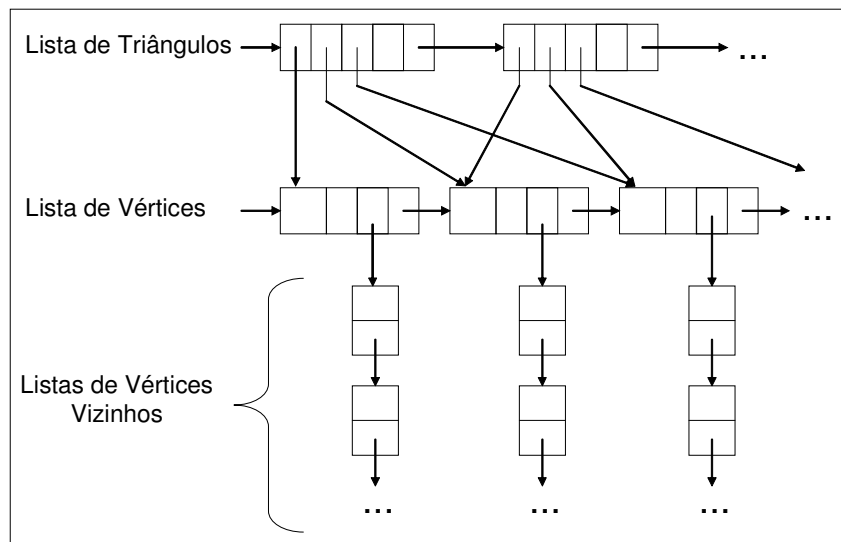


Figura 3. Relacionamentos das listas da classe objeto.

A referência para os triângulos vizinhos é armazenada na mesma ordem que o vértice do triângulo a quem ele é oposto. Deste modo e pelo fato que os ponteiros são

todos armazenados seguindo essa ordem, buscas mais rápidas dentro da ED são asseguradas.

O cálculo para determinar os triângulos vizinhos é feito por uma busca linear dentro da lista de triângulos, para cada triângulo, três pares de vértices diferentes são selecionados, e com eles, é feito uma busca por outro triângulo que contenha um destes pares, o triângulo achado é então referenciado como triângulo vizinho.

Os cálculos dos vetores normais são feitos após o cálculo de determinação dos triângulos vizinhos. Primeiramente, são executados os cálculos das normais dos triângulos e, em seguida, os cálculos das normais dos vértices.

Os cálculos de determinação dos vértices vizinhos e das normais dos vértices são feitos ao mesmo tempo. Para achar o vértice vizinho é feita uma busca utilizando a referência de triângulos vizinhos e as equações de calcular o próximo vértice e o vértice anterior (apresentados no tópico de *corner-table*). Por obrigação, um vértice tem no mínimo dois vizinhos (os outros vértices pertencentes ao mesmo triângulo deste vértice). A busca pelos outros vértices vizinhos pode apresentar três características que são ilustradas na figura 4. Esta busca baseia-se na busca por outros triângulos em que o vértice esteja presente, para isso usa-se a referência dos triângulos vizinhos.

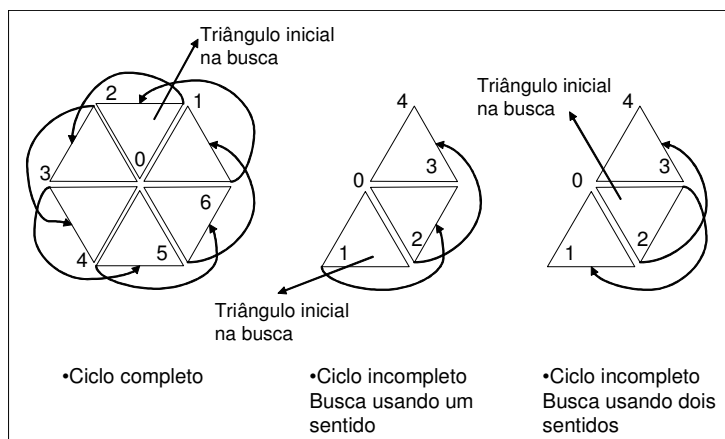


Figura 4. Modos de determinação do vértice vizinho.

O tempo necessário para a realização dos cálculos para gerar a ED cresce de acordo com o número de vértices e triângulos do modelo. Mesmo que este tempo não seja levado em consideração durante a simulação (a visualização em si só é feita após a geração de todos os dados), a espera pelos cálculos se torna inconveniente. A saída encontrada para este problema aponta para a criação de um arquivo intermediário após os cálculos, para que estes sejam feitos somente uma vez para um determinado modelo. Este arquivo contém os dados originais do arquivo fonte e os dados calculados a partir dos dados originais. Neste caso, ele será utilizado no lugar do arquivo fonte VRML sempre que for identificada a sua existência.

5. Resultados

A implementação da ED do sistema CyberMed seguiu o paradigma de Programação Orientada a Objetos (POO) e utilizou a linguagem de programação utilizada foi C++ ANSI. Neste caso, foi implementada uma classe que contém os atributos e métodos de

acesso da ED para possibilitar o uso dela pelas rotinas do CyberMed. Os atributos da ED foram implementados como privados para manter a integridade dos dados armazenados por ela.

A ED é carregada antes da execução das rotinas do CyberMed. Isto se faz necessário para evitar que o tempo de carregamento da ED afete o desempenho da execução das dessas rotinas. A geração de dados da ED se mostrou custosa. Como solução para isso, foi implementada uma rotina de importação dos dados de um modelo armazenados na ED para um arquivo intermediário de extensão CYB. Este arquivo se apresenta no formato texto e armazena ordenadamente os dados do objeto e seus cálculos após uma primeira utilização. Neste caso, evita-se a necessidade de se passar pelas rotinas de cálculos na próxima vez em que os dados do modelo forem necessários. A Tabela 2 mostra uma comparação de tempo utilizado para a leitura do arquivo do modelo e para gerar a ED completa (com os cálculos relativos ao objeto) de 7 objetos gráficos diferentes. A máquina utilizada para executar este teste foi um AMD Athlon XP 2600 com 1.0 Gb de RAM usando o sistema operacional Linux, distribuição Fedora Core 3 32 bits.

Tabela 2. Comparação de tempo para a execução da leitura e de cálculos dos dados de alguns modelos do CyberMed usando o arquivo fonte e o arquivo intermediário.

Modelo	Número de Vértices	Número de Faces Triangulares	Tempo da geração da ED usando arquivo VRML (em segundos)	Tempo da geração da ED usando arquivo intermediário (.cyb) (em segundos)
Cubo	8	12	0.04	0.04
Esfera	114	224	0.07	0.06
Corpo	3367	5534	4.83	0.21
Medula	8036	16072	56.75	0.71
Ossatura Pélvica	12070	24164	136.94	95
Sistema Digestivo	27276	53418	694.26	2.04
Musculatura Facial	31940	62348	902.50	2.42

A cada triângulo de um objeto gráfico são associados mais dados a ele do que os presentes no arquivo fonte, o que tornou necessário escolher modelos com quantidades muito grandes de triângulos e vértices para testar a capacidade da ED. Este teste também é importante, pois o CyberMed possibilita a utilização de modelos mais complexos com múltiplas camadas como mostra a Figura 5. Para esta figura, o tempo necessário para a execução da rotina de criação da ED utilizando o arquivo VRML foi de 171.76 segundos e utilizando o arquivo intermediário foi de 62.71 segundos.

6. Conclusões

Este trabalho apresentou os requisitos e dados de concepção de uma ED para um sistema de RV. A maneira em que foi planejada a implementação da ED garantiu algumas características importantes para o seu uso pelo sistema CyberMed: o compartilhamento dos dados desta ED pelas diversas rotinas do CyberMed; a

integridade de dados, já que suas informações são protegidas de alterações imprevistas; e acesso mais eficiente aos dados, já que dados são acessados mais diretamente evitando assim buscas complexas.

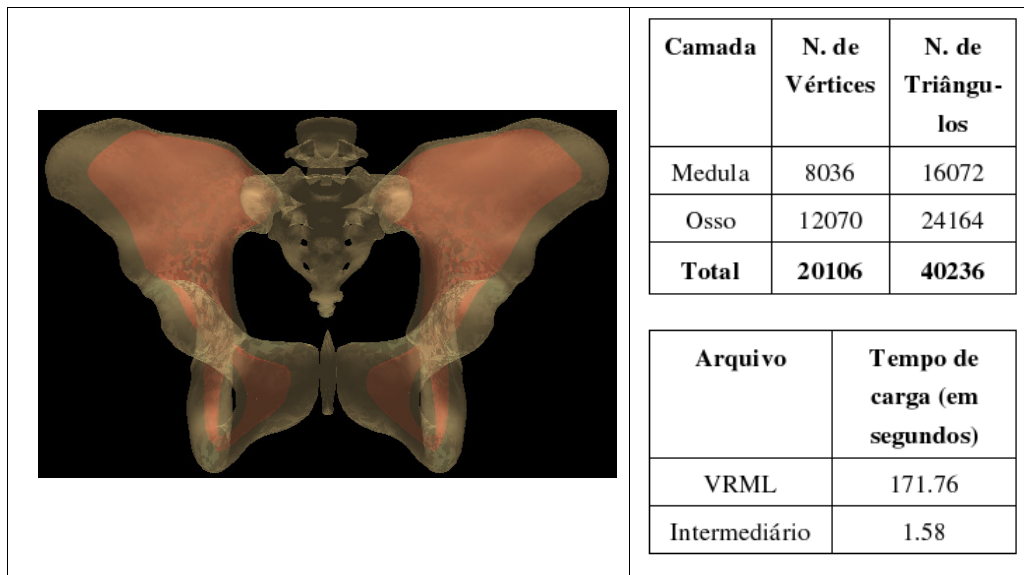


Figura 5. Visualização em dupla camada do osso e de sua medula. A camada do osso apresenta transparência para possibilitar a visualização da camada interna (a medula).

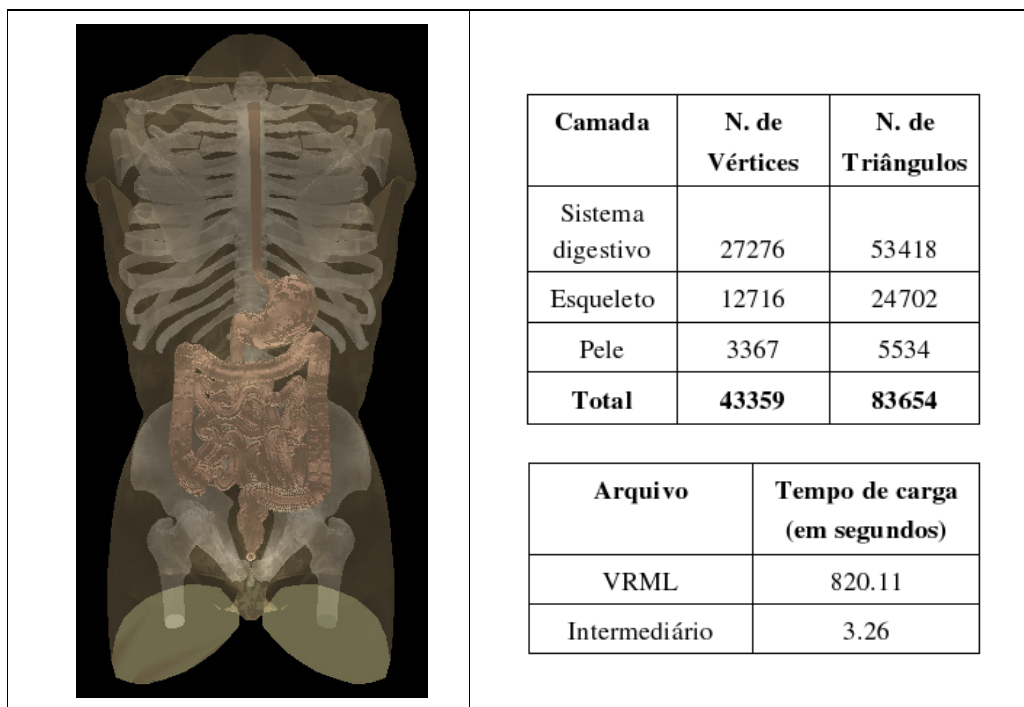


Figura 6. Visualização do tórax humano com 3 camadas: sistema digestivo, esqueleto e pele.

A geração da ED foi testada com vários modelos do CyberMed e outros mais simples. Entre os mais complexos dos modelos destacam-se o modelo da ossatura pélvica e da medula com 120106 vértices e 40236 faces triangulares (Figura 5), o modelo do apresentado na Figura 6 com 43359 vértices e 83654 faces triangulares e o da cabeça humana com 68960 vértices e 133198 faces triangulares (Figura 7). O uso destes modelos se faz importante para testar a capacidade da ED quanto ao uso de múltiplas camadas. As camadas têm seus dados armazenados separadamente uma das outras, o que é possível observar durante a visualização através das características de transparência atribuídas individualmente. Pode-se notar isso nas Figuras 5, 6 e 7 nas diferenças de transparência entre cada camada.

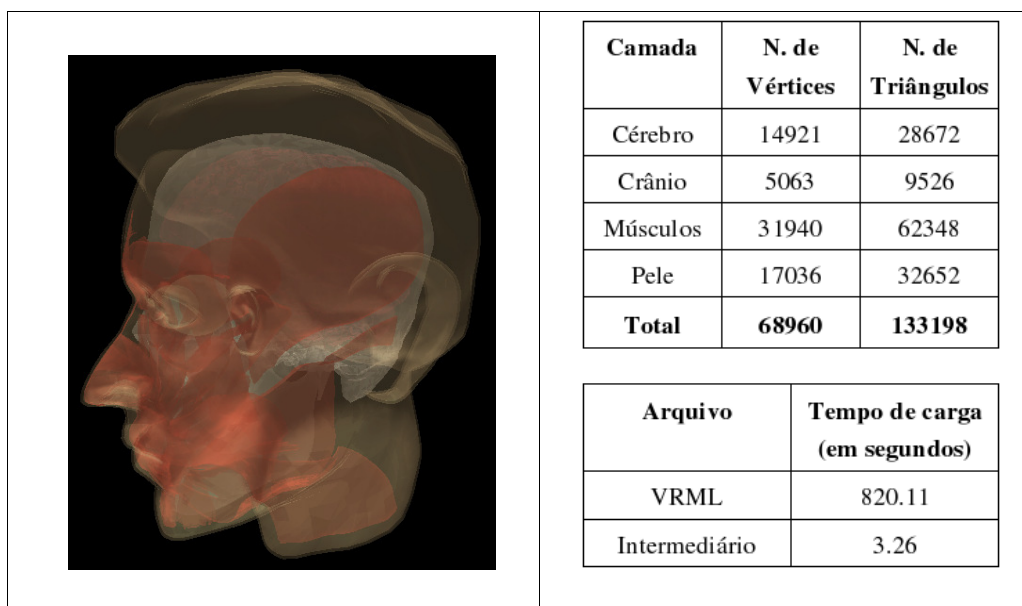


Figura 7. Visualização da cabeça humana com 4 camadas: cérebro, crânio, musculatura facial e pele.

Para melhorar o desempenho da rotina de criação da ED, já que os cálculos realizados por esta rotina são custosos quanto ao tempo, foi necessário criar um arquivo intermediário para o caso de um arquivo fonte de um modelo ser lido pela primeira vez. Neste caso, quando utilizar o modelo novamente, a leitura realizada será do arquivo intermediário, dispensando assim os cálculos de geração de dados. O uso destes arquivos intermediários se mostrou bastante eficiente em comparação com o uso dos arquivos fontes. Nas figuras 6 e 7, por exemplo, tiveram uma melhora significativa, para a figura 6 o tempo da execução da rotina de criação da ED utilizando o arquivo VRML foi de 820.11 segundos e utilizando o arquivo intermediário foi de 324.04 segundos, para a figura 7 o tempo da execução da rotina de criação da ED utilizando o arquivo VRML foi de 1310.55 segundos e utilizando o arquivo intermediário foi de 553.25 segundos.

Em geral, a ED apresenta uma série de vantagens para sistemas de RV. Pois ela suporta acessos a dados por diferentes rotinas críticas sem prejudicar a característica de tempo-real do sistema.

8. Agradecimentos

Este trabalho é parcialmente financiado pelo CNPq (CT-INFO 506480/2004-6) e pela FINEP (Processo 01-04-1054-000).

Referências

- Alterovitz, R.; Pouliot, J.; Taschereau, R.; Hsu, I. e Goldberg, K., (2003) “Simulating Needle Insertion and Radioactive Seed Implantation for Prostate Brachytherapy”, In: *Medicine Meets Virtual Reality 11*, Edited by James Westwood et al., IOS Press, pp. 19-25.
- Bryson, S., (1996) “Virtual Reality in Scientific Visualization”, In: *Communications of the ACM*. Vol. 39, No. 5, pp. 62-71.
- Burdea, G. e Coiffet, P. (2003), “Virtual Reality Technology”, Segunda Edição, Wiley-Interscience.
- Cunha, Í.; Moraes, R. e Machado, L., (2006) "Unifying Data Structures for Virtual Reality Applications", Proc. World Congress on Computer Science, Engineering and Technology Education - WCCSETE'2006, Santos, março.
- Gomes, J. e Velho, L. (1998) “Computação Gráfica Vol. 1”, Série de Computação e Matemática, IMPA.
- Machado, L.S. e Moraes, R.M. (2005) “Cenários 3D Interativos com Software Livre”, *Revista de Informática Teórica e Aplicada*. Vol. 12, No. 2, pp. 91-112.
- Machado, L.S.; Campos, S.; Cunha, Í. e Moraes, R., (2004) “CYBERMED: Realidade Virtual para Ensino Médico”, *IFMBE Proceedings*. Vol. 5, pp. 573-576.
- Münchenberg, J.; Wörn, H.; Brief, J.; Kübler, C.; Hassfeld, S. e Raczowsky, J. (2000) “Intuitive Operation Planning Based on Force Feedback”, In: *Medicine Meets Virtual Reality 2000*, Edited by James Westwood et al., IOS Press, pp. 220-226.
- Onishi, K.; Hasuike, S.; Kitamura, Y. e Kishino F., (2003) “Interaction Techniques: Interactive Modeling of Trees by Using Growth Simulation”, In: *Proc. ACM Symp. on VR SW and Tech.*, ACM Press, Osaka, Japão, pp. 66-72.
- Prusinkiewicz, P. and Lindenmayer, L., (1990) “The algorithmic beauty of plants”, Springer-Verlag, New York.
- Rossignac, J.; Safanova, A. e Szymczak, A., (2001) “3D Compression Made Simple: Edgebreaker on a Corner-Table”. *Proc. Shape Modeling International Conference*, Genova, Itália, maio, pp. 278-283.
- Tenembaum, A.; Langsam, Y. e Augenst, M., (1995) “Estruturas de Dados Usando C”, Makron Books.
- Vieira A.; Velho L.; Lopes H.; Tavares G. e Lewiner T., (2004) “Fast Stellar Mesh Simplification”, *Brazilian Symp. Comp. Graphics and Image Processing*, pp. 27-34.