

Utilização de um *Ray-Caster* na Visualização de Volumes: Uma Aplicação e um Estudo

Paulo Breda Vieira 

Dep. Electrónica e Telecomunicações
IEETA-Instituto de Eng. Elect. e Telemática de Aveiro
Univ. de Aveiro
breda@inesca.pt

Beatriz Sousa Santos

Dep. Electrónica e Telecomunicações
IEETA-Instituto de Eng. Elect. e Telemática de Aveiro
Univ. de Aveiro
bss@inesca.pt

Sumário

Neste artigo apresenta-se uma aplicação desenvolvida com o objectivo de estudar a utilização de um ray-caster na visualização de volumes definidos por voxels; esta aplicação permite a variação de alguns parâmetros que determinam a qualidade das imagens obtidas. Apresenta-se também um estudo feito utilizando esta aplicação e volumes sintéticos e naturais, variando os referidos parâmetros.

Palavras-chave

Ray-caster, visualização em volume, qualidade

1. INTRODUÇÃO

A Visualização de Dados é uma área em grande expansão, sendo a visualização de volumes definidos por *voxels* uma aplicação actualmente bastante importante, nomeadamente na visualização de dados provenientes de modalidades clínicas de diagnóstico como a Tomografia Axial Computorizada (TAC) e a Ressonância Magnética (RM) [Brodie92, Brown95]. Com o aumento de capacidade das máquinas, é possível construir ferramentas de software cada vez mais completas e sofisticadas, no entanto, depois de obtido o produto final do processo de visualização deparamo-nos com um problema: até que ponto é que a visualização obtida está correcta? Qual é o grau de fidelidade de determinada visualização? Como quantificar a qualidade de uma visualização?

Para ajudar na tarefa de obtenção de uma resposta a estas perguntas, foi criada uma aplicação que nos permite utilizar um *ray caster* [Levoy88] para a visualização de volumes de dados reais ou sintéticos. A realização desta aplicação permitiu obter um melhor conhecimento deste método (muito utilizado [Elvins92]) e consequentemente dos problemas e erros a ele associados. Permitiu também um maior controlo sobre o processo, possibilitando a realização de experiências em que será possível variar diversos parâmetros da visualização. Esta aplicação deverá facilitar a definição de "medidas" com interesse na avaliação da qualidade da visualização.

Inicialmente, pensou-se em criar uma aplicação mais simples, no entanto ao utilizar alguns dados reais provenientes de TAC e RM, surgiram dificuldades em

definir mapas de cor e opacidade adequados, o que nos levou a criar duas ferramentas adicionais, uma para edição de mapas de cor (*Map Editor*) e outra para análise dos volumes de dados (*Image Probe*), obtendo-se assim uma aplicação final com mais algumas potencialidades.

A aplicação, incluindo as ferramentas referidas e os algoritmos utilizados, serão brevemente descritos nas secções seguintes; são também apresentados, como exemplo do tipo de estudos que podem ser realizados, alguns resultados obtidos para diversos conjuntos de dados e correspondentes à variação de alguns parâmetros.

2. DESCRIÇÃO GERAL DA APLICAÇÃO

Como foi referido, a aplicação principal, que designamos por *Ray Caster*, integra duas ferramentas: o *Image Probe* (sonda de imagem) e o *Map Editor* (editor de mapas de cor) desenvolvidas com o fim de permitir a definição de mapas de cor e opacidade adequados a dados complexos, como os de TAC e RM. O *Image Probe* permite analisar, secção a secção, os dados e variações de valor para as várias zonas dos volumes; depois de feita esta análise, pode então utilizar-se a segunda ferramenta desenvolvida, o *Map Editor*, que permite definir os mapas de correspondência entre os valores dos dados e as cores e opacidades a utilizar no processo da visualização.

A aplicação foi desenvolvida para ambientes Windows 9x tendo sido utilizado o Microsoft Visual C++ 5 e poderá necessitar, para que se obtenham (bons) resultados (dependendo dos dados a visualizar e das opções seleccionadas), de uma máquina com alguma capacidade. Os testes foram efectuados num Pentium MMX 200, com 64Mbytes de RAM e, embora esta máquina se revelasse

suficiente, algum poder computacional e memória RAM adicionais seriam bem-vindos. Outro pormenor a ter em conta é a placa gráfica utilizada. Esta deverá trabalhar em modos de 24 bits de cor para resoluções de no mínimo 800x600, mas idealmente de 1024x768 ou mais.

2.1 Ray-Caster

A aplicação principal permite definir:

- quais as características dos dados que pretendemos visualizar, como a resolução (em *voxels*) e as dimensões no espaço (em unidades de espaço u.e.);
- qual o tipo de interpolação a utilizar no processo de *ray-casting*; actualmente estão implementados três tipos de interpolação (ordem zero ou *nearest neighbour* e interpolação trilinear na cor e nos valores);
- qual o valor do passo de amostragem (K), a utilizar no processo de acumulação ao longo de cada raio que é projectado; este parâmetro determina o número de amostras usadas para determinar a cor atribuída a cada pixel da imagem durante o processo de *rendering* e é dado em dimensões do espaço (unidades de espaço, u.e.) e não em *voxels*;
- a utilização, ou não, da optimização de *early ray termination*, que possibilita terminar a acumulação ao longo de um raio quando é atingido um determinado valor para a opacidade; quando esta opção está seleccionada, podemos definir o nível de paragem;
- parâmetros da(s) imagem(s) a criar, como a resolução (em *pixels*) da imagem que será gerada e o ângulo em que a câmara será colocada em relação ao centro do volume de dados (na versão corrente, a câmara é sempre colocada à distância do raio da menor esfera que contém os dados).
- parâmetros de animação para a criação de seqüências de imagens através da variação da posição da câmara; na realidade não será gerada uma animação, mas sim uma seqüência de *frames* que, posteriormente e através de outras aplicações, poderá ser convertida para um formato de vídeo.

Toda esta funcionalidade da aplicação *Ray Caster* é apresentada através de uma interface de utilizador que se organiza em torno de uma janela principal e que apresenta um pequeno menu (com as opções *File* e *Help*) e uma zona para introdução da informação necessária à execução de comandos (como mostrado na Fig. 1).

A zona de introdução de informação encontra-se dividida em três páginas (*Data*, *Options*, *Render*) que podem ser acedidas individualmente através da selecção do separador respectivo. A página *Data* permite definir as características dos dados a visualizar e aceder ao *Image Probe* e ao *Map Editor*. A página *Options* permite variar os parâmetros do algoritmo (passo de amostragem, interpolador e utilização de um critério de paragem). Finalmente a página *Render* (mostrada na Fig. 2) permite definir os parâmetros da(s) imagem(s) a criar.

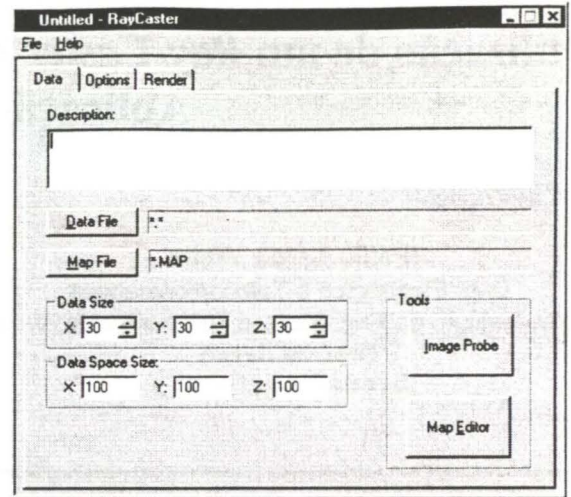


Figura 1 – Janela Principal da aplicação *Ray-Caster*

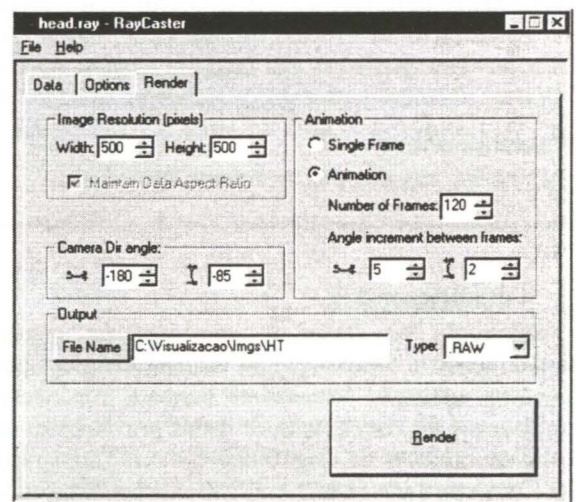


Figura 2 – Página *Render*

2.2 Image Probe

Esta ferramenta (cujo aspecto geral da interface é apresentada na Fig.3) possibilita uma visualização

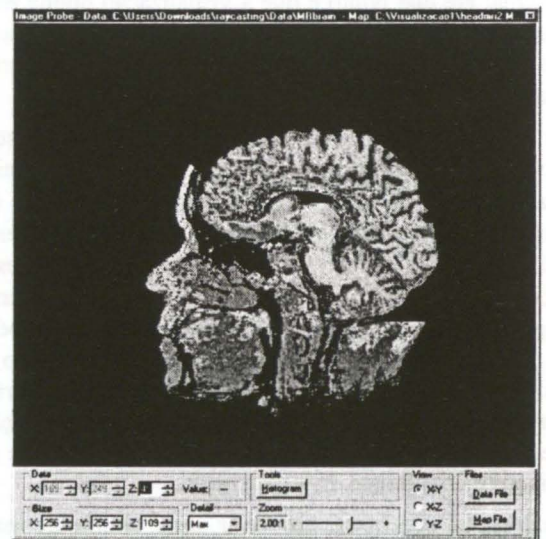


Figura 3 – Aspecto geral da interface da *Image-Probe*

auxiliar dos dados em secções paralelas aos planos coordenados (perpendiculares aos eixos XX, YY ou ZZ) com a finalidade de fornecer ao utilizador uma ideia aproximada da variação dos valores dos dados ao longo de todo o volume.

O utilizador pode seleccionar a parte do volume, a resolução e o factor de escala que pretende usar em cada visualização.

2.3 Map Editor

Map Editor (cujo aspecto geral da interface com o utilizador é dado na Fig.4) permite criar e editar, por manipulação directa, os mapas de cor a utilizar na geração das imagens. Estes mapas são definidos a partir das suas quatro componentes (vermelho, verde, azul e opacidade) sendo as acções sobre as componentes executadas apenas sobre aquela que se encontra seleccionada. A componente de Vermelho (Red) é representada a vermelho, a componente Verde (Green) a verde, a componente Azul (Blue) a azul e a Opacidade (Opacity) a branco. Depois de feita a definição do mapa este pode ser guardado em ficheiro para utilização futura, podendo ser traçados gráficos a partir de valores armazenados em ficheiros.

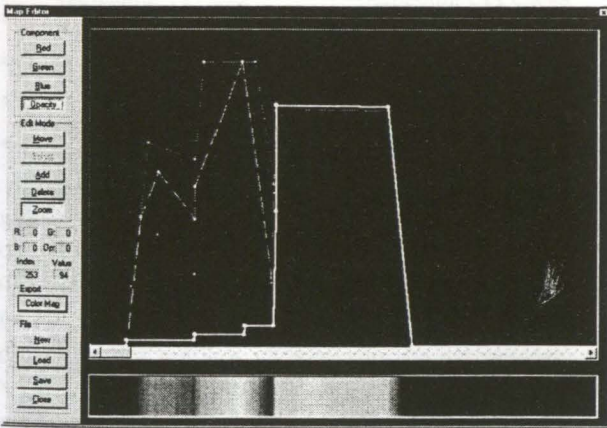


Figura 4 – Aspecto geral da interface do *Map-Editor*

3. ALGORITMO E OPTIMIZAÇÕES

O algoritmo utilizado é, como foi anteriormente referido, o descrito por Levoy [Levoy88] e implica a projecção de um raio através de cada *pixel* da imagem. Este raio é amostrado com um passo K , fazendo-se para cada amostra a acumulação da cor e da opacidade, resultando no fim deste processo a cor a atribuir ao *pixel*. A expressão usada para acumulação da cor e da opacidade é a seguinte,

$$C_{out} = C_{in} + C_s \alpha_f (1 - \alpha_{in})$$

$$\alpha_{out} = \alpha_{in} + \alpha_f (1 - \alpha_{in})$$

Onde os índices *out* representam os valores resultantes da acumulação para cada amostra, os *in* os valores anteriores e os *f* e os *s* os valores no ponto de amostragem.

Para calcular os valores de cor e de opacidade correspondentes ao ponto em que é feita a amostragem

podem usar-se métodos de interpolação, mais ou menos sofisticados. A aplicação desenvolvida permite a utilização de um de três métodos diferentes (ordem zero ou *nearest neighbour*, interpolação trilinear nos valores e interpolação trilinear na cor); permite também levar a acumulação ao longo de cada raio até ao fim ou pará-la quando se atinge determinado valor de opacidade, como explicado nas secções seguintes.

3.1 Tipos de Interpolação

A interpolação mais simples consiste na utilização, para valores de cor e opacidade correspondentes ao ponto de amostragem, dos valores correspondentes ao *voxel* de cujo centro o ponto de amostragem está mais próximo; este tipo de interpolação chama-se de ordem zero ou *nearest neighbour*. Este método embora seja simples de implementar e rápido não produz, em geral, bons resultados, pelo que mais comumente se utiliza uma interpolação trilinear. A interpolação trilinear é equivalente a uma simples interpolação linear a três dimensões; assim, consideram-se os oito *voxels* vizinhos do ponto de amostragem e faz-se uma soma pesada dos seus valores, considerando para peso o valor $(1-d)$, em que d é a distância entre o centro do *voxel* e o ponto de amostragem. Se a interpolação for feita nos valores, procura-se a cor correspondente no mapa de cor e utiliza-se esta como a cor no ponto, se for na cor, calculam-se as cores dos oito *voxels* e em seguida interpolam-se estas para obter a cor final correspondente ao ponto de amostragem.

3.2 Early Ray Termination

O método de *early ray termination* é bastante utilizado para optimizações de velocidade devido à sua simplicidade e à possibilidade de diminuição do tempo de processamento.

Se observarmos as equações de acumulação de cor e opacidade descritas em 3., podemos ver que os valores de cor e opacidade acumulados se obtêm adicionando uma parcela que é multiplicada pelo valor de 1 menos a opacidade acumulada até ao momento. Assim, quando a opacidade acumulada atinge um valor próximo de 1, as amostras seguintes no raio praticamente não têm qualquer influência na cor final do *pixel*, uma vez que os seus valores de cor e opacidade são multiplicados por um valor ≈ 0 . O método de *early ray termination* consiste em parar a acumulação ao longo de um determinado raio quando a opacidade atingir um nível seleccionável (*alpha threshold* na página *Options*).

3.3 Compensação de Opacidade

O processo de acumulação de cor e opacidade é executado, para cada raio, um número de vezes que depende do valor de K e das dimensões dos dados no espaço. Se fosse utilizado o algoritmo apenas como é indicado em 3. quando se aumentasse o número de amostras (diminuisse o K) iríamos obter uma opacidade maior, mais cedo no raio.

Para que ao variar o K não se obtivesse diferentes resultados, utilizou-se a seguinte compensação para a opacidade:

$$\alpha_f = 1 - (1 - \alpha_s)^k$$

ou seja: se utilizarmos um determinado K, obtém-se um valor para a opacidade acumulada; se utilizarmos um K n vezes mais pequeno, precisaremos de n passos para atingir a mesma opacidade acumulada. Conseguiu-se assim tornar a opacidade final do volume menos dependente de K.

4. RESULTADOS

Apresentam-se nesta secção alguns estudos realizados com a aplicação desenvolvida, utilizando conjuntos de dados sintéticos e reais e variando o parâmetro K e o tipo de interpolação. Pretende-se com eles exemplificar o tipo de estudos que podem ser feitos para obter uma melhor compreensão de erros associados à visualização de volumes através de um *ray-caster* e consequentemente a sua melhor utilização. Tendo em vista a definição de potenciais medidas da qualidade, são também propostos “números de erro” que caracterizam as imagens produzidas.

O primeiro estudo realizado correspondeu à visualização de vários conjuntos de dados, sintéticos e reais, com diferentes valores de K. Com este estudo pretendeu-se averiguar se é possível determinar um valor de K a partir do qual a “qualidade” das imagens obtidas não aumente significativamente. Se for possível determinar este valor, pode-se utilizá-lo como critério de optimização do algoritmo, já que a utilização de valores de K inferiores (e consequentemente mais amostras ao longo de cada raio), não trazendo melhoria significativa à qualidade das imagens, aumenta o tempo de processamento. Apresenta-se uma dificuldade à partida: como definir “qualidade” da imagem obtida? Este problema é complexo; existem vários níveis a que pode ser feita a avaliação da qualidade em visualização e vários tipos de métodos que podem ser usados para o fazer. Neste trabalho limitámo-nos a propor a utilização de métodos objectivos, quantitativos e não filtrados pelo Sistema Visual Humano. Este tipo de métodos é designado por índices de qualidade num trabalho em que se procura estabelecer um *framework* para a avaliação da qualidade em visualização de dados médicos [Sousa-Santos97].

O segundo estudo correspondeu à visualização dos mesmos conjuntos de dados com tipos de interpolação diferentes. Neste caso apenas se procedeu à avaliação da qualidade das imagens, por simples inspecção visual.

4.1 Variação de K

O primeiro volume utilizado corresponde a um conjunto de dados sintético, composto por 4 cubos e uma esfera. Dois cubos são transparentes (com graus de opacidade diferentes), os outros dois cubos e a esfera são quase opacos. O volume de dados tem $100 \times 100 \times 100$ voxels e dimensão de $20 \times 20 \times 20$ unidades de espaço (u.e.³), sendo a resolução das imagens de 400×400 pixels. Estas imagens foram geradas com passo de amostragem (K)

variável, tendo-se usado a interpolação trilinear na cor. Algumas das imagens obtidas são apresentadas na Fig. 5.

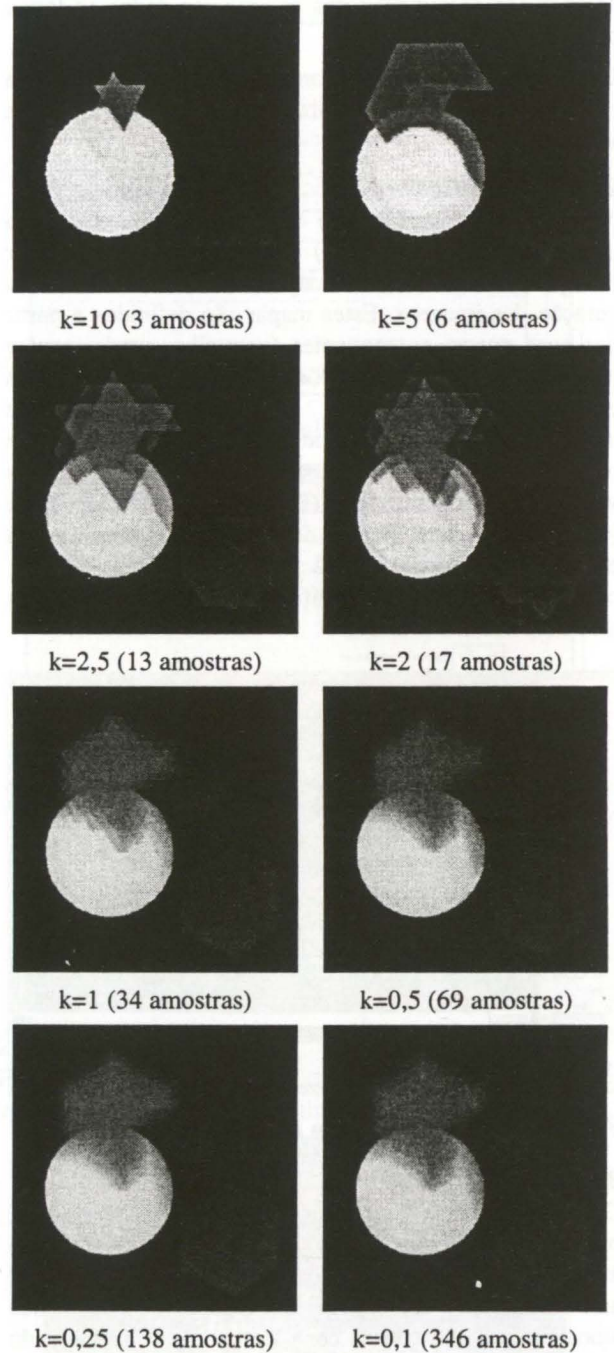


Figura 5 – Imagens obtidas, para um volume sintético, com diferentes valores de K (amostras ao longo do raio)

Analisando estas imagens pode ver-se que a opacidade dos vários volumes é praticamente independente do K, pelo que a compensação utilizada parece satisfatória.

Se examinarmos a sequência de imagens nota-se uma grande degradação da qualidade para valores de K muito elevados. No caso extremo (K=10), para o cálculo da cor em cada *pixel* apenas são usadas cerca de 3 amostras segundo a diagonal do cubo de $20 \times 20 \times 20$ u.e.³ (o que corresponde ao pior caso), enquanto esse número sobe para cerca de 300 quando K=0,1. O tempo que cada uma

destas imagens demora a gerar, como é óbvio, também cresce quando se diminui o K. Aparentemente, não parece que se obtenha um grande acréscimo na qualidade da imagem quando o número de amostras ao longo do raio começa a ser maior do que o número de *voxels* ao longo de uma das dimensões. Esta constatação não nos surpreende já que esta situação corresponde aproximadamente a uma amostra por *voxel*, no caso ideal de o raio projectado ser paralelo a um dos conjuntos de arestas dos *voxels*.

Para verificar esta hipótese, efectuaram-se diversas medidas sobre imagens de 400x400 *pixels*, geradas a partir de um conjunto de dados reais com 256x256x109 *voxels* e 50x50x30 u.e.³.

Considerou-se para imagem padrão uma imagem obtida com um K correspondente a 8 amostras/voxel (i.e., ≈ 3000 amostras ao longo da diagonal) e calcularam-se as diferenças, *pixel a pixel*, entre as diversas imagens obtidas com outros valores de K e a imagem padrão.

Para quantificar as diferenças entre cada imagem e a imagem padrão foram utilizados cinco tipos de “números de erro”, calculados como:

- o número de diferenças
- a soma das diferenças
- a média das diferenças (soma dif./número dif.)
- o máximo das diferenças
- a diferença média por pixel (soma dif./nº total de pixels)

nas componentes de cor (RGB), nas intensidades e nas distâncias no espaço RGB dos *pixels* correspondentes nas duas imagens.

Os resultados obtidos são apresentados na Fig. 6 sendo as ordenadas dos gráficos apresentadas em percentagem do valor máximo teórico.

Observando os gráficos da Fig. 6 verifica-se que, até ao valor de K correspondente a 0,8 amostras/voxel, existe uma queda acentuada dos “números de erro” e que a partir deste, os valores tornam-se muito pequenos. Este valor de K corresponde a 307 amostras ao longo do raio, que é o primeiro valor superior à maior resolução segundo uma das dimensões (256 *voxels*). Os erros a partir deste valor tornam-se muito pequenos, mas, no entanto, constata-se que um aumento do número de amostras para o dobro (K para metade) ainda corresponde a uma diminuição nos valores das diferenças para aproximadamente metade. Assim, aumentar o número de amostras ao longo do raio, parece ainda aumentar a qualidade da imagem, o que poderá eventualmente ser explicado pelo facto de, no caso geral, os raios projectados através de cada *pixel* poderem não ser paralelos a um conjunto de arestas dos *voxels*. Esta questão poderia ser averiguada variando também de forma controlada a direcção destes raios.

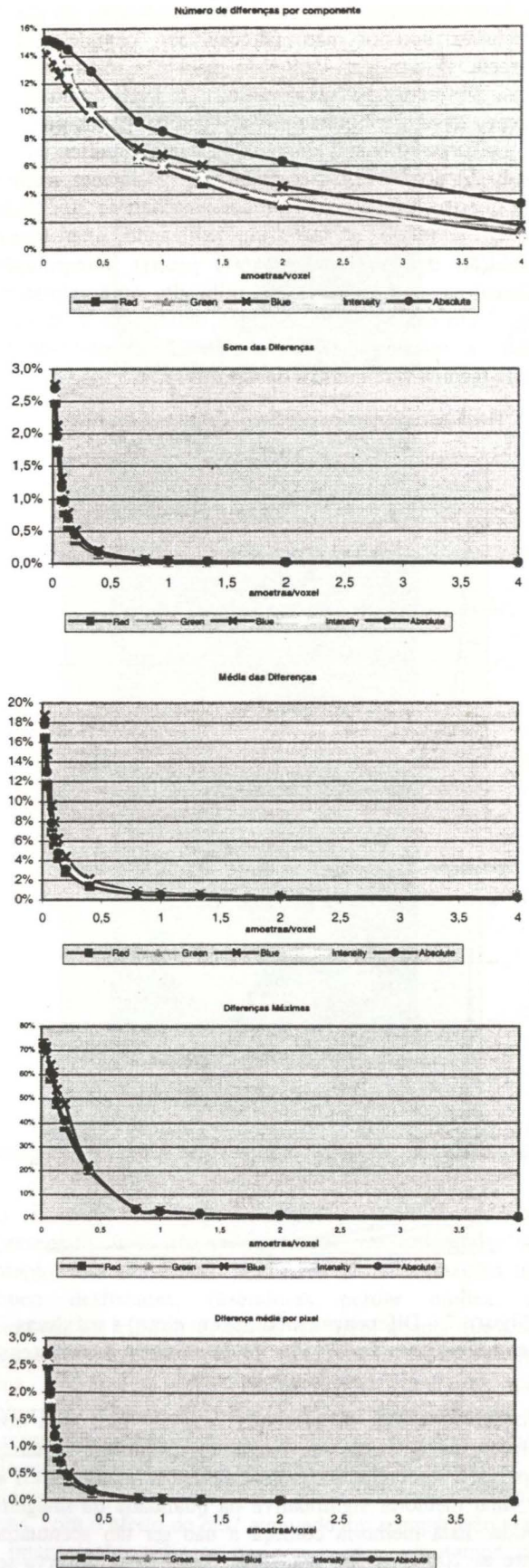
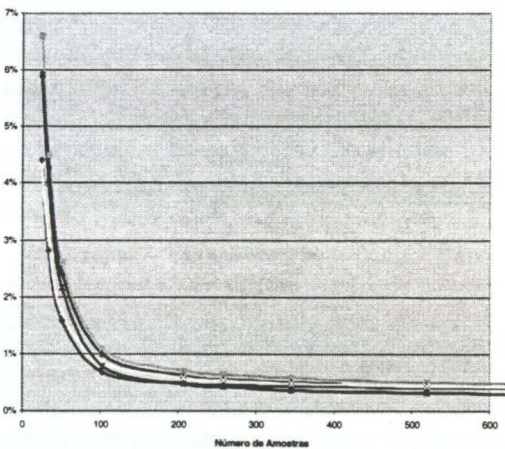


Figura 6 – Diferenças em função do número de amostras por voxel

Ao fazer diversas experiências, houve situações em que a conclusão anterior não pareceu ser completamente correcta, já que com dados de resolução relativamente baixa, em *renderings* com resolução elevada da imagem, pareceu haver melhorias quando se utilizava um número de amostras ao longo do raio, bastante superior ao da resolução dos dados. No sentido de esclarecer melhor esta questão, criou-se um volume sintético de $30 \times 30 \times 30$ *voxels*, e fez-se o *rendering* utilizando uma maior resolução da imagem (600×600 *pixels*), interpolação trilinear na cor e variando o K utilizado. Apresenta-se na Fig. 7 um resumo dos resultados obtidos, as diferenças média e máxima para cada valor de K utilizado (e consequentemente número de amostras).

Percentagem das diferenças médias em função do n. de passos



Percentagem das diferenças máximas em função do n. de passos

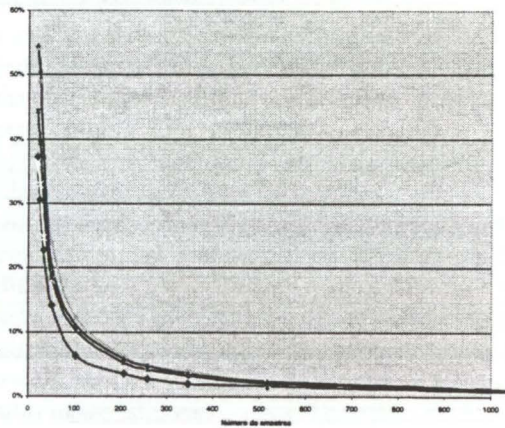
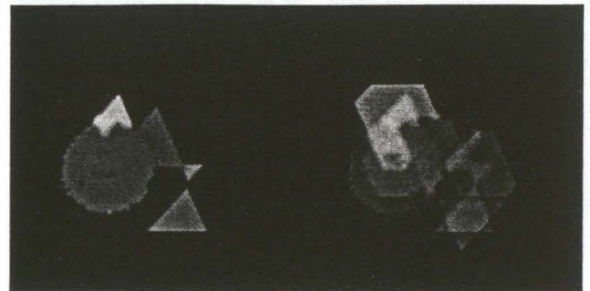


Figura 7 – Diferenças médias (em cima) e máximas (em baixo) para cada valor de de número de amostras ao longo do raio

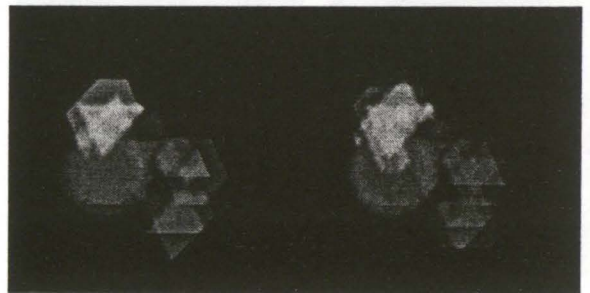
Analisando a Fig. 7, podemos verificar que, para um número de amostras ao longo de cada raio bastante superior à resolução do volume de dados, continuamos a ter uma melhoria significativa na qualidade da imagem obtida. Esta melhoria começa a não ser tão acentuada entre as 100 e as 300 amostras. Neste caso, observa-se que o número de amostras ao longo do raio que corresponde ao melhor compromisso qualidade/velocidade é da ordem de grandeza da resolução da

imagem gerada. Parece assim não ser a resolução dos dados o único factor a influenciar o valor ideal para o número de amostras ao longo do raio. Uma amostra da sequência de imagens gerada pode ser vista na Fig. 8.



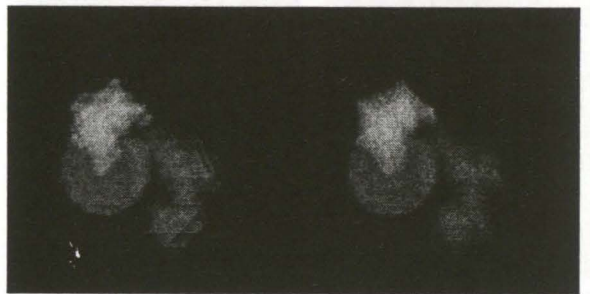
10 amostras

20 amostras



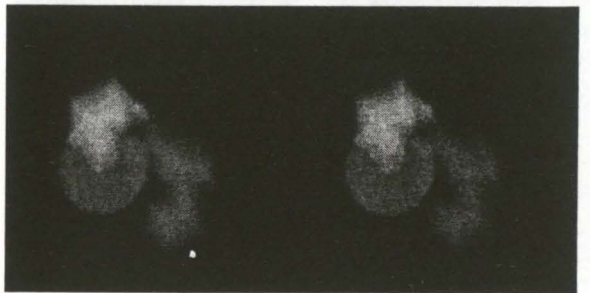
25 amostras

34 amostras



51 amostras

103 amostras



207 amostras

259 amostras

Figura 8 – Sequência de imagens geradas com uma resolução de 600×600 *pixels*

As experiências realizadas levam-nos a confirmar que a determinação de um valor de K (e consequentemente do número de amostras a utilizar ao longo de cada raio) que optimize o compromisso qualidade/velocidade, é um problema complexo e parece depender de vários factores, nomeadamente da resolução dos dados e da resolução da imagem gerada. Embora o primeiro factor pareça ser o dominante, e portanto um bom candidato para critério de escolha de um K quasi-óptimo, colocam-se ainda duas questões essenciais na escolha do K óptimo:

- qual dos “números de erro” propostos (ou algum outro) é mais adequado como medida (objectiva, quantitativa e que não leva em conta o Sistema Visual Humano) de qualidade;
- se este tipo de medida é suficiente como descrição da qualidade das visualizações obtidas.

Para esclarecer estas questões afigura-se-nos necessária a realização de outros estudos, nomeadamente de experiências controladas com observadores humanos ou modelos do sistema visual humano que permitam avaliar, em primeiro lugar, se algum destes “números de erro” é conveniente como medida de qualidade das imagens obtidas e caso isto se verifique, qual o valor de K a partir do qual as melhorias “medidas” dessa forma deixam de ser visíveis.

4.2 Variação do Tipo de Interpolação

O parâmetro que se estuda agora é o tipo de interpolação utilizada no algoritmo. Como referido anteriormente, a aplicação permite a utilização de três tipos de interpolação.

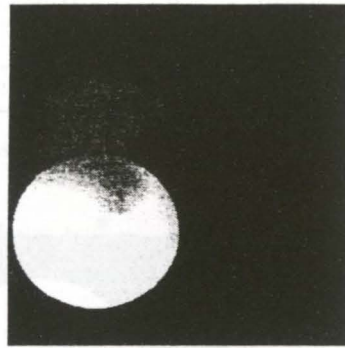
Apresentam-se na Fig. 9, para o conjunto de dados sintéticos já descritos na secção anterior, as imagens obtidas para cada um dos tipos de interpolação. A resolução das imagens geradas foi mais uma vez de 400x400 pixels, sendo as dimensões do espaço 20x20x20 u.e.³.

A imagem obtida por interpolação trilinear na cor apresenta transições mais suaves entre os diversos volumes e entre estes e o fundo negro. A imagem obtida por interpolação nos valores resulta mal porque o tipo de dados gerados e o mapa de cor utilizado não foram os mais adequados a uma interpolação deste tipo. Para utilizar interpolação nos valores, os valores dos vários volumes devem estar bastante distanciados e o mapa de cor deve possuir as transições entre as diversas cores associadas. No caso apresentado, os vários volumes tinham valores sequenciais (0-espaço, 1,2,3,4-cubos, 5-esfera) estando o mapa de cor definido apenas para estes valores. Assim, nas superfícies, quando é feita a interpolação, o valor resultante pode ter uma cor muito diferente das cores dos valores usados na interpolação.

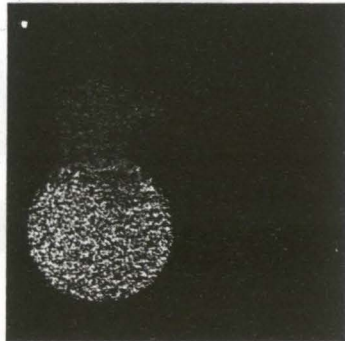
Utilizando um conjunto de dados diferente, obtidos a partir do projecto Visible Human [VisHuman95] e com um mapa de cor mais contínuo, voltaram a fazer-se alguns testes de variação do tipo de interpolação utilizada, de que se apresentam alguns resultados na Fig. 10.

Para obter as imagens apresentadas, utilizou-se uma resolução da imagem de 400x400 pixels e um número de amostras ao longo do raio de pouco mais de 10000 (sendo a resolução dos dados 432x140x240 u.e.³). Os tempos de *rendering*, para este volume e na plataforma descrita na secção 2, são apresentados na tabela I. O tempo de *rendering* para a interpolação trilinear nos valores é bastante inferior ao da interpolação trilinear na cor uma vez que são efectuados menos cálculos em vírgula flutuante. É no entanto um pouco superior ao

tempo de *rendering* com interpolação ordem zero, como seria de esperar.



Interpolação Ordem zero



Interpolação Trilinear nos valores



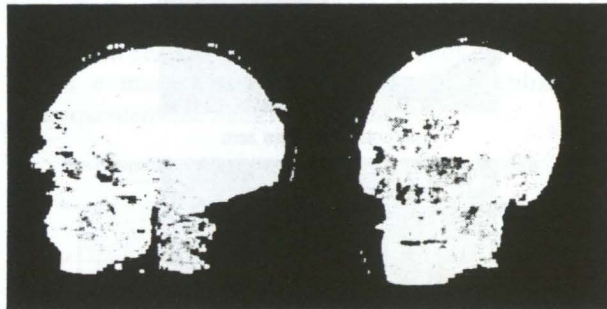
Interpolação Trilinear na cor

Figura 9 –Imagens geradas com diferentes tipos de interpolação: ordem zero (em cima), trilinear nos valores (no centro) e trilinear na cor (em baixo)

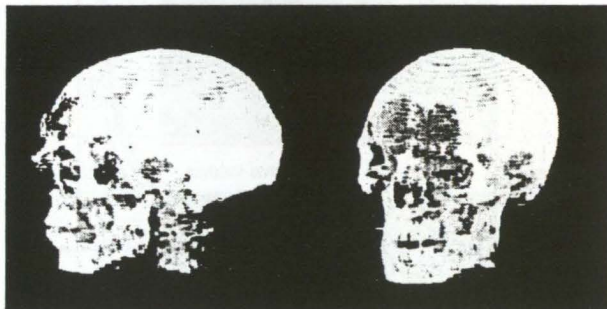
As imagens obtidas por interpolação trilinear na cor, apresentam uma maior suavidade de contornos, no entanto, esta suavização por vezes torna as imagens um pouco desfocadas, fazendo-as perder nitidez. A interpolação trilinear é, apesar de tudo, aquela que gera os melhores resultados para visualização de dados com uma amostragem relativamente baixa, uma vez que consegue atenuar o efeito escada mais visível na interpolação ordem zero. Se se utilizar um mapa de cor adequado, quando os dados a visualizar o permitam, a interpolação nos valores pode gerar resultados bastante bons, com o efeito de *blur* atenuado em comparação com a interpolação trilinear na cor e com um tempo de *rendering* pouco superior ao da interpolação ordem zero, parecendo portanto um bom compromisso.

Tipo de interpolação	Tempo de <i>rendering</i>
Ordem zero	31 min
Trilinear nos valores	42 min
Trilinear na cor	2h 45 min

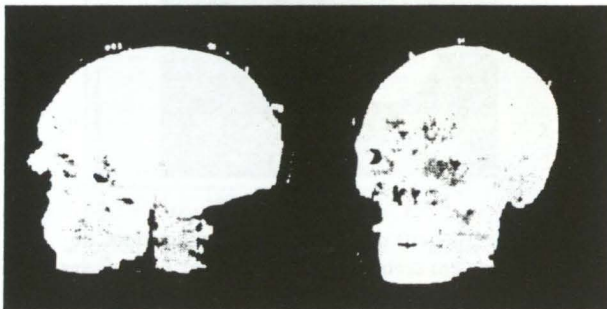
Tabela I – Tempos de *rendering* de um volume real, para os diversos tipos de interpolação



Interpolação Ordem zero



Interpolação Trilinear nos valores



Interpolação Trilinear na cor

Figura 10 – Imagens geradas com diferentes tipos de interpolação: ordem zero (em cima), trilinear nos valores (no centro) e trilinear na cor (em baixo)

5. CONCLUSÕES E TRABALHO FUTURO

Embora a aplicação desenvolvida possua já algumas capacidades, existem ainda diversos melhoramentos possíveis, tais como introdução de opções correspondentes à determinação de histogramas, à utilização de vários tipos de *shading*, à variação da distância a que se encontra a câmara e à utilização de outros métodos de optimização ao algoritmo de *ray cast*.

No desenvolvimento da aplicação tentou-se por vezes optimizar o código, mas essa não foi a principal preocupação. Esta orientação poderá ter influenciado

negativamente a eficiência da implementação executada: obtiveram-se tempos de *rendering* que chegaram às dezenas de minutos ou mesmo horas para imagens de maior resolução e *Ks* menores. No entanto, tendo em conta os fins a que se destina a aplicação, não se considera este factor muito limitativo.

Apesar das limitações referidas consideramos que a aplicação desenvolvida constitui uma boa base para a realização de experiências que permitam obter uma melhor compreensão do algoritmo de *ray-caster* e dos factores que influenciam a qualidade das visualizações obtidas por este método, um dos métodos mais usados em visualização de volumes definidos por *voxels*.

O estudo apresentado na secção de resultados constitui exemplo do tipo de estudos que podem ser realizados utilizando esta aplicação. Os resultados nele obtidos não são completamente conclusivos quanto à variação da qualidade das visualizações obtidas, no entanto fornecem-nos indicações úteis para a realização de futuros estudos, mais sistemáticos, em torno de várias questões que parecem importantes. Destas questões realçamos os parâmetros que são mais significativos (e.g. *K*, nº de amostras por voxel ao longo do raio) e quais os números de erro propostos que devem ser usados como medidas de qualidade e como estes se correlacionam com a qualidade percebida por observadores humanos.

6. REFERÊNCIAS

- [Brodie92] Brodrie, K., L. Carpenter, R. Earnshaw, J. Gallop, R. Hubbard, A. Mumford, C. Osland, P. Quarendon, *Scientific Visualization, Techniques and Applications*, Springer Verlag, 1992
- [Brown95] Brown, J., R. Earnshaw, M. Jern, J. Vince, *Visualization, Using Computer Graphics to Explore Data and Present Information*, John Wiley, 1995
- [Elvins92] Elvins, T., "A Survey of Algorithms for Volume Visualization", *Computer Graphics*, vol.26, num.3, pp. 194-201, 1992
- [Kaufman96] Kaufman, A., "Recent Trends in Volume Visualization", *2nd IEEE-EMBS International Summer School*, June 1996, Berder Island, France.
- [Keller93] Keller, P., M. Keller, *Visual Cues*, IEEE Computer Society Press, 1993
- [Levoy88] Levoy, M., "Display of Surfaces from Volume Data", *IEEE Computer Graphics and Applications*, vol.8, num.3, pp. 29-37, 1988
- [Sousa-Santos97] Sousa Santos, B., J.L. Dillenseger, J.L. Coatrieux, "Some Recurrent Concepts of Quality Evaluation in Medical Visualization", *Electrónica e Telecomunicações*, Universidade de Aveiro, Vol. 2, N.2, Janeiro 1998, pp. 244-247
- [VisHuman95] *Digital Humans: A Multimedia Tour of the Visible Human Project (CD)*, Multimedia Medical Systems, 1995