

Ray Casting: Uma implementação com vista à realização de estudos de qualidade*

Paulo Breda Vieira, Beatriz Sousa Santos

Abstract- An application to visualize voxel based volumes using a ray-caster is presented. This application is meant to allow studying the quality of obtained visualizations and thus the values of a set of parameters relevant to the final quality may be selected.

Sumário- Neste artigo apresenta-se uma aplicação desenvolvida para possibilitar a visualização de volumes definidos por voxels, utilizando o método de Ray Casting. Esta aplicação foi desenvolvida com o objectivo de estudar a qualidade das visualizações, e assim, permite variar diversos parâmetros associados ao método implementado, para posterior análise dos resultados.

I. INTRODUÇÃO

A Visualização de Dados é uma área em grande expansão, sendo a visualização de volumes definidos por voxels uma aplicação bastante importante, nomeadamente na visualização de dados provenientes de modalidades imageológicas como a Tomografia Axial Computorizada (TAC) e a Ressonância Magnética (RM) [1,2]. 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 executar o *Ray Casting* [3] de volumes de dados reais ou obtidos sinteticamente. A realização desta aplicação permitiu obter um melhor conhecimento deste método (muito utilizado em visualização de volumes [4]) 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 poderá incluir diversas “medidas” durante o processo, o que deverá facilitar a definição de parâmetros com interesse na avaliação da qualidade da visualização.

De início, quando se começou a criar a aplicação (*Ray Caster*), pensou-se apenas em criar a rotina que efectuava o processo de *Ray Casting* e eventualmente um interface

gráfico simples que lhe servisse de suporte. Utilizaram-se alguns conjuntos de dados sintéticos bastante simples, e também dados reais mais complexos obtidos por métodos de RM e TAC. Foi ao tentar visualizar estes últimos conjuntos de dados que surgiram as maiores dificuldades em encontrar/criar mapas de cor e opacidade adequados. Assim, criaram-se 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 já com algumas potencialidades para a análise do método.

A aplicação, incluindo as ferramentas referidas, será descrita neste artigo, sendo apresentada a sua interface para com o utilizador final. Serão depois descritos alguns resultados e por fim serão apresentados conclusões e trabalho futuro.

II. A APLICAÇÃO DE RAY CASTING

Foi desenvolvida de raiz uma aplicação para efectuar o *Ray Casting* dos volumes de dados. Esta aplicação corre em ambiente Windows 9x e a ferramenta de programação utilizada foi o Microsoft Visual C++ 5.

Esta aplicação permite carregar um ficheiro com os dados do volume a visualizar e também um ficheiro com os mapas de cor e de opacidade/transparência. Depois de seleccionados os ficheiros, existem diversas opções que serão descritas mais adiante e que permitem variar alguns dos parâmetros da visualização e também seleccionar qual o formato da informação que deverá ser apresentada no fim do processo.

Paralelamente à aplicação principal, o *Ray Caster* propriamente dito, existem duas outras: o *Image Probe* (sonda de imagem) e o *Map Editor* (editor de mapas). Como já foi referido, ao iniciar este estudo utilizaram-se alguns volumes de dados resultantes de TACs e RMs. Para conseguir produzir uma visualização com alguma qualidade, tornou-se essencial definir mapas de cor e opacidades adequados aos dados, assim, com o *Image Probe*, criou-se uma ferramenta que nos permite analisar os dados e ver as variações para as várias zonas dos volumes; depois de analisados os dados, utiliza-se então a segunda ferramenta desenvolvida, o *Map Editor*, que nos permitirá definir os mapas que farão a correspondência entre os valores das leituras dos dados e as cores e opacidades a utilizar no processo da visualização. Estas

* Trabalho efectuado no âmbito da cadeira Visualização por Computador, do Mestrado em Engenharia Electrónica e de Telecomunicações.

duas ferramentas serão descritas em maior pormenor na secção sobre o Interface do Utilizador.

A aplicação, dependendo dos dados a visualizar e das opções seleccionadas, poderá necessitar, para que se obtenham (bons) resultados, de uma máquina com alguma capacidade. Os testes foram quase todos 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, sendo este pormenor apenas limitativo para a visualização, uma vez que os resultados produzidos em ficheiro são sempre a 24 bits, independentemente da placa gráfica utilizada.

A. Interface de Utilizador

Nesta secção será apresentada a interface de utilizador da aplicação desenvolvida, nas três partes em que esta se encontra dividida que são o *Ray Caster* propriamente dito e as ferramentas adicionais criadas, ou seja, o *Map Editor* e o *Image Probe*.

A.1 Ray Caster

A janela principal do *Ray Caster* apresenta um pequeno menu, e uma zona para introdução dos dados e execução de comandos. O menu da aplicação possui os comandos normais para gravar e carregar ficheiros, bem como uma zona para acesso rápido aos documentos utilizados mais recentemente. Os dados que são guardados ao gravar um documento são os que aparecem nas páginas de configuração da aplicação que serão descritas adiante. Em relação ao Mapa de Cor, a única informação que é guardada é a do nome e localização do ficheiro a utilizar, sendo o conteúdo do ficheiro alterado e armazenado dentro da aplicação *Map Editor*, como será descrito na secção respectiva.

A zona de introdução encontra-se dividida em três páginas que podem ser acedidas individualmente através da selecção do separador respectivo.

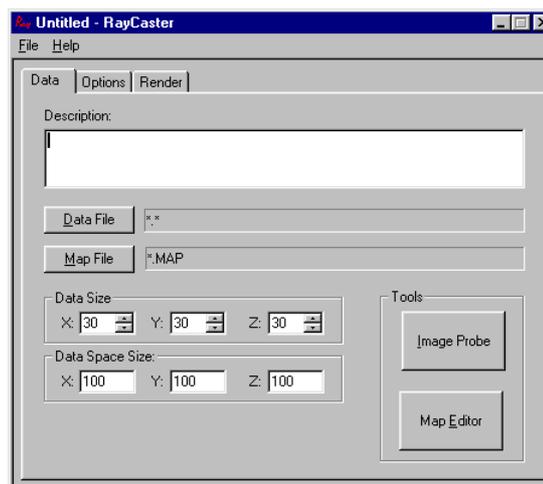


Fig. 1 - A Página de dados

A página de dados (Fig. 1) possui diversos campos que nos permitem definir as características dos dados que pretendemos visualizar. Inclui ainda o acesso ao *Image Probe* e ao *Map Editor*. Podemos assim introduzir nesta página um pequeno texto com a descrição dos dados e eventualmente alguma particularidade relativa a esta visualização e definir qual o ficheiro de dados e qual o ficheiro com a informação do mapa de cor a utilizar. Podemos também indicar quais as dimensões dos dados, isto é a resolução do volume de dados, e quais as dimensões dos dados no espaço. A definição das dimensões no espaço permite-nos trabalhar com volumes de dados não isotrópicos permitindo-nos compensar o seu *aspect ratio*.

A página de opções permite-nos definir qual o tipo de interpolação a utilizar no processo de *Ray Casting* - actualmente estão implementados três tipos: interpolação Ordem Zero (*Nearest Neighbour*), interpolação Trilinear na Cor e interpolação Trilinear nos Valores.

Podemos ainda definir qual o valor do passo a utilizar no processo de acumulação de cor em cada raio que é projectado. Este valor é dado em dimensões do espaço e não dos dados (amostras).

A terceira opção que nos é dada é a de utilizar ou não a optimização de *Early Ray Termination*, que possibilita terminar a acumulação de cor quando é atingido um determinado valor para a acumulação da opacidade num pixel da imagem. Quando esta opção está seleccionada, passamos a poder definir qual o *threshold* de paragem.

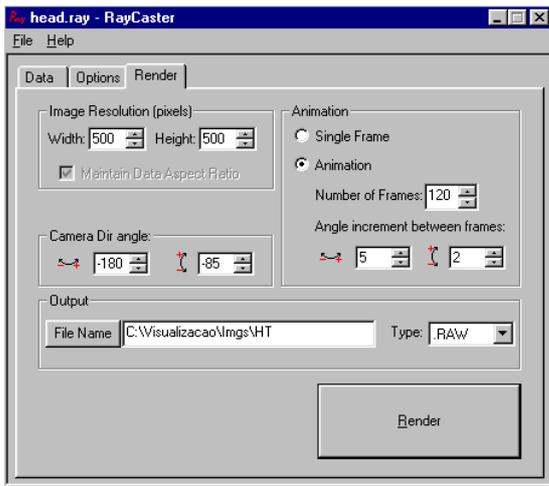


Fig. 2 - Página de Render

A página de *Render* (Fig. 2) permite-nos definir diversos parâmetros da(s) imagem(s) a criar. Podemos definir qual a resolução (em pixels) da imagem que será gerada, sendo, como é óbvio, o tempo de rendering tanto maior quanto maiores forem estas dimensões. Podemos também definir qual o ângulo em que a câmara será colocada em relação ao centro do volume de dados. Nesta versão, a câmara é sempre colocada à distância do raio da menor esfera que contém os dados.

Esta página contém uma zona de parâmetros de animação que nos dá a hipótese de criar sequências de imagens através da variação da posição da câmara. Na realidade não será gerada uma animação, mas sim uma sequência de *frames* que posteriormente, e através de outros programas, poderão ser convertidas para um formato de vídeo.

Na zona de parâmetros de animação podemos optar por gerar uma *frame* única ou uma animação. No caso de seleccionarmos a animação, poderemos então definir o número de *frames* a gerar, e ainda os incrementos angulares horizontais e verticais da câmara entre duas *frames* consecutivas.

Na zona de *Output*, definimos qual a localização e o nome do ficheiro gerado, e ainda o seu formato (para já apenas o formato RAW é suportado). No caso de ser gerada uma animação, o nome aqui definido servirá como base para os ficheiros gerados, sendo-lhe depois adicionado um número sequencial de quatro dígitos para cada *frame*.

A.2 Image Probe

Esta é a ferramenta desenvolvida para nos permitir analisar o volume de dados e verificar as gamas de valores que os diversos constituintes do volume apresentam, permitindo-nos assim criar um mapa de cor mais adequado à visualização pretendida.

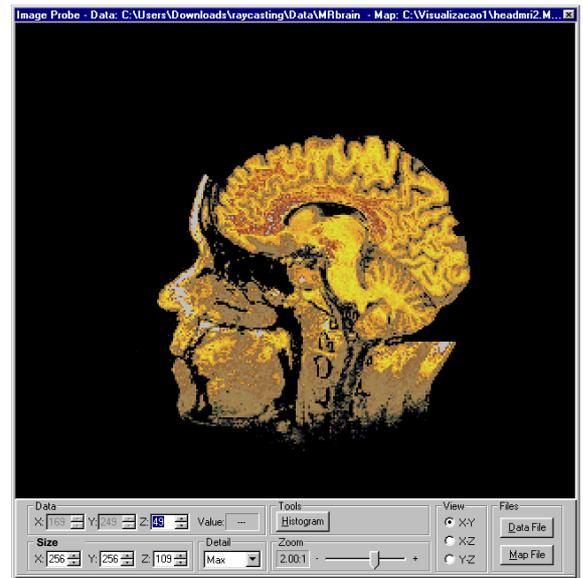


Fig. 3 - Image Probe

A janela principal pode ser vista na Fig. 3 sendo a zona inferior da janela aquela em que se encontram os diversos controlos.

Nesta ferramenta podemos obter uma visualização auxiliar dos dados em secções paralelas a qualquer dos três planos coordenados (perpendiculares aos eixos XX, YY ou ZZ).

Deslocando o cursor sobre a imagem, podemos observar o valor dos dados no ponto em que este se encontra e as suas coordenadas.

Podemos ainda definir e alterar as dimensões do volume de dados, variar a ampliação com que são apresentados os dados, e alterar o detalhe com que as secções serão apresentadas no écran. Esta última opção pode ser bastante útil para navegar em volumes de resolução elevada, uma vez que permite tornar o display das secções mais rápido à custa de um menor detalhe nestas.

A.3. O Map Editor

O *Map Editor* (Fig. 4) permite-nos criar e editar os mapas de cor a utilizar no processo de *Ray Casting*.

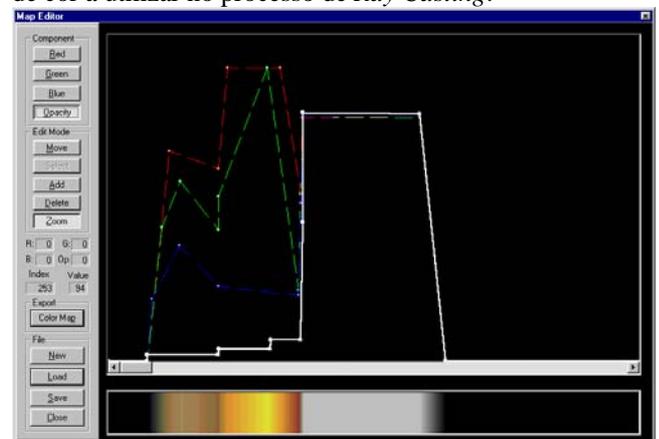


Fig. 4 - Map Editor

Esta ferramenta apresenta três áreas distintas no écran: uma, à esquerda, em que se encontram os diversos

controles e opções; uma zona inferior em que se encontra uma previsão do mapa de cor correspondente à definição actual; e por fim na área maior do écran, podemos ver e alterar a contribuição das componentes RGB da cor, bem como a opacidade.

Na área de controlos podemos seleccionar qual a componente em que estamos a trabalhar. A componente seleccionada aparecerá como uma linha contínua mais grossa, enquanto as restantes aparecerão a tracejado fino, além disso, o botão correspondente ficará premido. As acções sobre as componentes são sempre executadas apenas sobre a que se encontra seleccionada. A componente de Vermelho (Red) é representada com a cor vermelha, a componente Verde (Green) com a cor verde, a componente Azul (Blue) com a cor azul e a Opacidade (Opacity) com a cor branca.

Na área de controlos podemos também alternar entre os diversos modos de edição disponibilizados, e que definirão o resultado das acções do rato sobre a janela de componentes. O botão correspondente ao modo seleccionado ficará premido, e, além disso, o cursor quando se encontrar sobre a janela de componentes mudará de acordo com a acção que será executada. Os modos disponíveis são: o modo *Move* para mover os pontos de controle da componente seleccionada; o modo *Add* para adicionar novos pontos de controle; o modo *Delete* para apagar pontos de controle; e por fim o modo *Zoom* para variar a ampliação com que vemos as janelas de componentes e de previsão da cor.

Existe ainda uma zona que nos dá indicações acerca do ponto em que se encontra o cursor, e por fim uma outra que possui os controlos destinados a trabalhar com ficheiros.

III - ALGORITMOS E OPTIMIZAÇÕES

A. Ray Casting

O método utilizado para efectuar o *Ray Casting* é o descrito por Levoy [3], e que corresponde à projecção de um raio por cada pixel da imagem, sendo este raio percorrido com um passo determinado (K na aplicação) fazendo-se a acumulação da cor e da opacidade a cada passo, resultando no fim a cor do pixel. A fórmula para acumulação da cor e da opacidade é a seguinte,

$$\begin{aligned} C_{out} &= C_{in} + C_f \alpha_f (1 - \alpha_{in}) \\ \alpha_{out} &= \alpha_{in} + \alpha_f (1 - \alpha_{in}) \end{aligned} \quad (1)$$

Onde os índices *out* representam o valor resultante em cada passo, os *in* o valor do passo anterior e os *f* o valor no ponto de amostragem.

B. Interpolação trilinear nos valores

A interpolação trilinear é o equivalente a uma simples interpolação linear só que adaptada para três dimensões. Assim, consideram-se os oito voxels que envolvem o

ponto de amostragem e faz-se uma soma pesada dos seus valores, considerando para o peso o valor 1 menos a distância do voxel ao ponto. Procura-se no mapa de cor a cor correspondente e utiliza-se esta como a cor no ponto.

C. Interpolação trilinear na cor

Esta interpolação é muito semelhante à anterior, no entanto, em vez de se fazer a média pesada dos valores dos voxels e posteriormente verificar qual a cor correspondente no mapa de cor, aquilo que se faz é calcular a cor para cada um dos oito voxels e de seguida interpolar estas para obter a cor final do ponto.

D. Early Ray termination

O método de *early ray termination* é um método que é bastante utilizado para optimizações de velocidade em *Ray Casting* devido à sua simplicidade e à possibilidade de obtenção de bons resultados.

Se observarmos com atenção as equações de acumulação de cor e opacidade descritas em III.A, podemos ver que os novos 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, os pontos seguintes no raio projectado praticamente não têm qualquer influência para a cor final do pixel na imagem uma vez que são multiplicados por um valor muito próximo de zero.

O método de *early ray termination* consiste então em parar a acumulação de cor para um determinado raio quando a opacidade atingir um nível parametrizável (*alpha threshold* na página de opções).

E. Compensação da 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 da visualização. Se utilizássemos o algoritmo simplesmente como é indicado em III.A, quando aumentássemos o número de passos de acumulação (diminuíssemos o K) iríamos obter uma opacidade maior, mais cedo no raio. Assim, para que ao variar o K não obtivéssemos diferentes resultados, utilizou-se a seguinte compensação para a opacidade,

$$\alpha_f = 1 - (1 - \alpha_s)^K \quad (2)$$

ou seja, se utilizarmos um K pequeno (<1), a opacidade que é utilizada na acumulação (α_f) é mais pequena do que a opacidade da amostragem (α_s), se o K for grande (>1) a opacidade utilizada é superior à da amostragem. Conseguiu-se assim tornar a opacidade final do volume independente de K, podendo-se variar este para obter maior ou menor precisão no processo de *Ray Casting*.

As dimensões do volume no espaço influenciam a opacidade final do volume: quanto maiores forem as

dimensões (no espaço) do volume, mais opaco este será, já que a opacidade foi compensada/normalizada para 1 unidade de espaço. Esta normalização para 1 unidade de espaço significa que é necessário percorrer 1 unidade de espaço de um determinado volume com uma determinada opacidade, para que a contribuição deste volume para a acumulação no raio seja igual à sua opacidade, independentemente do passo com que o volume é percorrido. Assim, se a dimensão do volume for inferior a 1 unidade de espaço, qualquer que seja o passo com que se percorre o volume, a opacidade final é sempre inferior à opacidade do volume. A única excepção ocorre para as opacidades extremas, isto é 0 ou 1, uma vez que a fórmula de compensação utilizada não altera estes valores.

IV - RESULTADOS

Apresentam-se nesta secção alguns resultados obtidos com a aplicação desenvolvida, utilizando conjuntos de dados sintéticos e reais, a título de exemplo do tipo de estudos sobre a qualidade das imagens que pode ser feito.

As primeiras imagens geradas corresponderam a um conjunto de dados sintéticos, composto por 4 cubos e uma esfera, com cores e opacidades diferentes, com 100x100 x100 voxels e dimensão de 20x20x20 unidades de espaço (u.e.). Foram geradas imagens com uma resolução de 400x400 pixels correspondentes a uma variação do passo de acumulação (K) ao longo do raio projectado entre 0,25 e 10, tendo-se usado a interpolação trilinear na cor.

Nestas imagens pode constatar-se que a opacidade dos vários volumes era praticamente independente do K, pelo que a compensação utilizada pareceu satisfatória. Notou-se ainda uma grande degradação da qualidade para valores de K muito elevados, já que no caso extremo de K=10, apenas são feitas cerca de 3 (34,6/10) amostragens para o cálculo da cor em cada pixel segundo a diagonal do cubo de 20x20x20 u.e. (o que corresponde ao pior caso), enquanto esse número sobe (linearmente) para cerca de 300 com K=0,1. O tempo que cada uma destas imagens demorou a gerar, como é óbvio, também aumenta quando se diminui o K. Aparentemente, não parece que se tenha obtido um grande acréscimo na qualidade da imagem quando o número de passos utilizados na acumulação começa a ser maior do que o número de voxels (para K=0,1 temos 10 amostras/u.e., 346 amostras ao longo do raio segundo a diagonal).

Para tentar verificar esta hipótese, efectuaram-se diversas medidas com as imagens de 400x400 pixels, geradas a partir de um conjunto de dados reais com 256x256x109 voxels, 50x50x30 u.e. Considerou-se para imagem padrão uma imagem obtida com um K de 0,025 (i.e., ≈ 3072 amostras ao longo da diagonal) e calcularam-se as diferenças pixel a pixel entre as diversas imagens obtidas com outros valores de K e esta.

Os “números de erro” utilizados para avaliar a qualidade das imagens correspondem às diferenças entre as componentes de cor de cada pixel, as diferenças entre as intensidades do pixel e a diferença absoluta entre pixels.

As diferenças nas componentes de Vermelho, Verde e Azul (RGB) são diferenças simples das componentes de cor para cada pixel das imagens. A diferença em intensidade consiste na diferença entre as intensidades dos pixels, ou seja a diferença entre as médias das três componentes da cor para cada pixel. Por fim, a diferença absoluta é a distância, no espaço RGB, entre as coordenadas da cor de cada pixel, ou seja, é a raiz quadrada da soma dos quadrados das diferenças de cada uma das componentes da cor do pixel:

$$\text{Dif. Absoluta} = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (3)$$

É de notar que as imagens utilizadas têm uma grande área negra que representa uma grande área em que o erro entre as diversas imagens geradas é nulo (o valor da cor é sempre zero), assim, nos cálculos em percentagem em função de um valor máximo que teoricamente poderia ser atingido, os valores medidos são inferiores ao que na realidade se devia avaliar caso se utilizasse apenas a zona de dados não nula. Apesar desta eventual imprecisão nas medições, a evolução na variação relativa para os diferentes valores de K continua a ser válida.

Observando as variações dos diversos “números de erro” com K, verifica-se que até ao valor K=0,25 existe uma queda acentuada e que a partir deste, com excepção do primeiro, número de diferenças, os valores tornam-se muito pequenos. Este valor de K corresponde a um número de amostras ao longo do raio de 307, que é o primeiro valor superior à maior resolução de amostras (256). Os erros a partir deste valor tornam-se muito pequenos, no entanto, constata-se que um aumento do número de amostras para o dobro (K para metade) corresponde a uma diminuição nos valores das diferenças medidas para aproximadamente metade. Assim, aparentemente, aumentar o número de amostras aumenta a qualidade da imagem, no entanto não é fácil determinar qual é o valor a partir do qual o erro não tem significado. Esta questão é de solução complexa tornando-se necessário realizar outros testes, possivelmente com observadores ou modelos do sistema visual humano que permitam avaliar, em primeiro lugar, se algum destes parâmetros 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 deixam de ser visíveis. No entanto, se se pretender um compromisso entre velocidade/qualidade, o valor de K óptimo parece estar perto do correspondente a um número de passos semelhante ao da resolução máxima dos dados, uma vez que a partir daí, a variação absoluta nas diferenças toma valores muito pequenos quando comparados com os anteriores. 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 de resolução elevada, pareceu haver melhorias quando se utilizava um número de passos bastante superior ao da resolução dos dados.

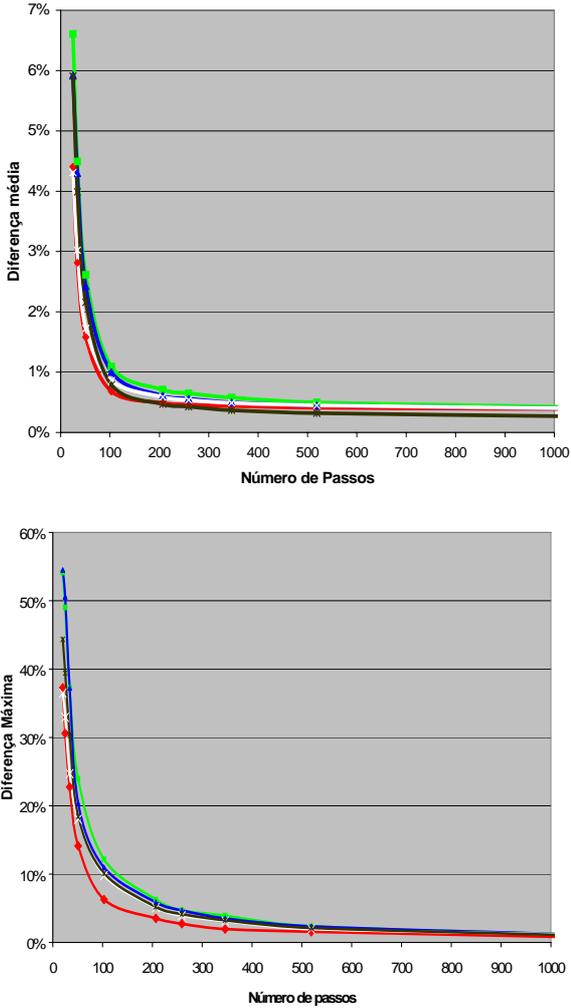


Fig. 5 - Percentagem das diferenças (médias em cima e máximas em baixo) em função do nº de amostras ao longo do raio

Assim, para averiguar melhor esta questão, criou-se um volume sintético de 30x30x30, e fez-se o rendering utilizando para dimensões da imagem 600x600 e interpolação trilinear na cor, variando o K utilizado.

Apresenta-se na Fig. 5, um resumo dos resultados obtidos, apenas com as diferenças média e máxima. 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 os 100 e os 300 passos. Assim, parece não ser a resolução dos dados o único factor a influenciar o valor de compromisso velocidade/qualidade para o número de amostras de composição no raio projectado; observa-se que o número de amostras ao longo do raio em que o melhor compromisso se obtém é da ordem de grandeza da resolução da imagem gerada. Uma amostra da sequência de imagens geradas pode ser vista na Fig. 6.

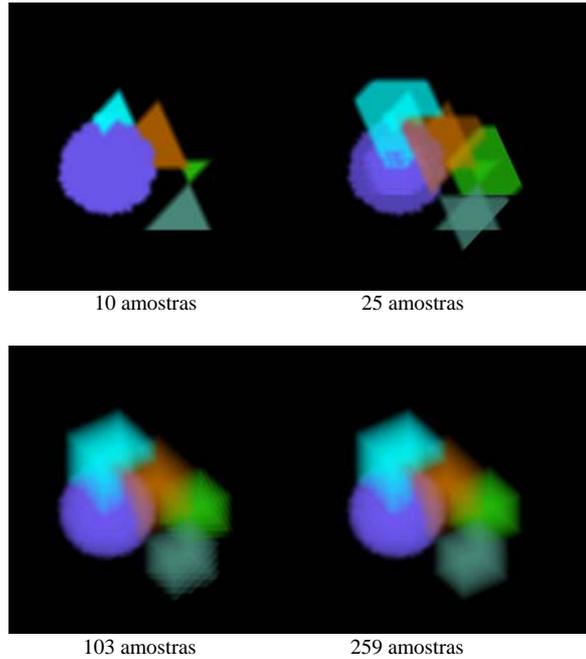


Fig. 6 - Sequência de imagens, com 600x600 pixels, geradas com vários valores diferentes de K (e consequentemente de amostras ao longo de cada raio) e usando um volume artificial de 30x30x30 voxels

V - CONCLUSÕES E TRABALHO FUTURO

No desenvolvimento da aplicação tentou-se otimizar o código, não tendo sido, no entanto, esta a principal preocupação, uma vez que se pretendia essencialmente ter o algoritmo a funcionar de maneira a obter resultados. Esta orientação poderá ter influenciado de forma negativa a eficiência da implementação executada; este factor não é muito limitativo, tendo em conta os fins a que se destina a aplicação e a possibilidade de execução de diversas *frames* sem intervenção do utilizador.

A aplicação desenvolvida pode constituir uma boa base para algum desenvolvimento adicional e para testes sobre a qualidade de visualização em imagens obtidas através de métodos de *Ray Casting*. Possui já algumas capacidades, existem ainda diversos melhoramentos e opções que podem ainda ser introduzidos. Assim, em relação ao método de *Ray Casting*, não foi desenvolvido qualquer código para o *shading* dos resultados, podendo-se criar esta opção utilizando por exemplo *Depth-only shading*, *Gray Level shading* ou *Phong Shading* [5]. Actualmente pede simular-se o primeiro utilizando uma opacidade não nula para o espaço vazio no volume.

Também não é possível variar a distância aos dados a que se encontra a câmara. A introdução desta opção permitiria fazer *zoom* e a visualização de pormenores.

Existe na ferramenta *Image Probe* um comando de criação de histogramas ainda não implementado; esta opção poderá tornar mais fácil a identificação dos valores limite de zonas localizadas do volume, pelo que o seu desenvolvimento seria também um acréscimo importante.

Neste momento apenas é possível criar animações a partir de *frames* obtidas com diferentes posições da

câmara; seria interessante poder variar outros dos parâmetros que são disponibilizados e também outros a implementar como o *Zoom* ou o centro da visualização. A variação do mapa de cor pode também gerar animações com efeitos interessantes, especialmente com a variação das opacidades, permitindo a oclusão/visualização de zonas específicas dos dados.

Uma questão a não desprezar numa aplicação como esta é a da velocidade. As rotinas desenvolvidas podem ainda ser bastante optimizadas, uma vez que são feitos muitos cálculos em vírgula flutuante e há ainda outras hipóteses de pré-calcular valores que se mantêm constantes durante determinados cálculos.

Para melhorar a velocidade do método, pode-se também introduzir a utilização de *bounding boxes* que limitam o volume de dados que necessita de ser atravessado para gerar a imagem. Esta hipótese permite simultaneamente a introdução de zonas de corte nos dados através da exclusão de partes do volume.

Não fica certamente completa a lista de tarefas que poderão ser realizadas para melhorar a aplicação, mas apresenta-se já um conjunto de ideias razoável que poderão servir de base a uma futura implementação.

REFERÊNCIAS

- [1] Brodlie, K., L. Carpenter, R. Earnshaw, J. Gallop, R. Hubbard, A. Mumford, C. Osland, P. Quarendon, *Scientific Visualization, Techniques and Applications*, Springer Verlag, 1992
- [2] Brown, J., R. Earnshaw, M. Jern, J. Vince, *Visualization, Using Computer Graphics to Explore Data and Present Information*, John Wiley, 1995
- [3] Levoy, M., "Display of Surfaces from Volume Data", *IEEE Comp. Graph. and Appl.*, vol.8, num.3, pp. 29-37, 1988.
- [4] Elvins, T., "A Survey of Algorithms for Volume Visualization", *Computer Graphics*, vol.26, num.3, pp.194-201, 1992
- [5] Kaufman, A., "Recent Trends in Volume Visualization", 2nd IEEE-EMBS International Summer School, Berder Island, France, 1996.