

# Aplicativo de Reformatação Curvilínea: Complexidade de Espaço

Wu Shin - Ting  
DCA - FEEC - Unicamp

17 de Setembro de 2009

Este documento apresenta a complexidade de espaço do atual aplicativo de reformatação curvilínea, desenvolvido com base nas imagens de equipamentos de ressonância magnética de 1,5 Tesla do Hospital das Clínicas da Unicamp, e uma análise preliminar do que se pode fazer para adequá-lo às imagens adquiridas por um aparelho de resolução bem maior, como o Philips Achieva de 3 Tesla recentemente instalado no Hospital das Clínicas da Unicamp.

## 1 Aplicativo de Reformatação Curvilínea

Sob o ponto de vista de complexidade de espaço, o aplicativo consiste de seguintes módulos (Figura 1);

**Importação:** módulo que scaneia um arquivo de uma série de imagens em formato DICOM e converte para o formato interno de processamento. Uma cópia binária, de extensão *.raw*, é guardada em *harddisk* (HD). O tamanho deste arquivo é  $VolSize = width \times height \times nSlices \times nBytes$ , onde *width* e *height* correspondem a número de colunas e número de linhas de cada imagem, *nSlices*, a número de planos do volume integral, e *nBytes*, a quantidade de *bytes* (8 *bits*) utilizados para representar os valores de cada amostra. É ainda gerado um arquivo do tipo cabeçalho, em ASCII, de extensão *.dat*.

**Equalização:** é opcional, com o objetivo de aumentar o contraste em imagens. O espaço requerido para processamento é *VolSize*.

**Cômputo e Processamento de Gradientes:** módulo que determina, para cada amostra, a variação do seu valor em relação aos valores das amostras vizinhas a ele. Para cada valor escalar de *nBytes* é gerado um vetor de  $3 \times nBytes$ . Nesta etapa são ainda aplicados filtros para suavizar ruídos. Para estes processamentos, é necessário um espaço um pouco superior

a  $7 \times nBytes$ . Com intuito de aproveitar os resultados para processamentos posteriores, é armazenado o resultado em um arquivo de extensão .grd em HD, também no formato binário. O tamanho deste arquivo é  $3 \times VolSize \times nBytes$ .

**Pré-integração:** a contribuição dos valores das amostras para a cor de cada *pixel* ao longo de uma linha de visão é “pré-somada” e armazenada em uma tabela para evitar cálculos sob demanda, comprometendo a interatividade. O espaço demandado é  $VolSize$ .

**Compactação:** o valor cada amostra e o seu respectivo gradiente são compactados em um espaço correspondente à representação de uma cor, ou seja,  $4 \times VolSize \times byte$ .

**Renderização e Interação:** este módulo é responsável pela geração de imagens. Para assegurar a interatividade, todos os dados relacionados com a manipulação direta são carregados na memória de textura da placa de vídeo. Além de algumas texturas de suporte, merecem destaque três texturas que requerem espaço dependente do tamanho do volume de amostras: textura de valores e gradientes ( $VolSize \times 3$ ), textura de valores pré-integrados ( $VolSize$ ) e textura de visibilidade ( $VolSize$ ). Vale observar que, para reduzir o uso de espaço, todos os valores das amostras são quantizados em 1 *byte*.

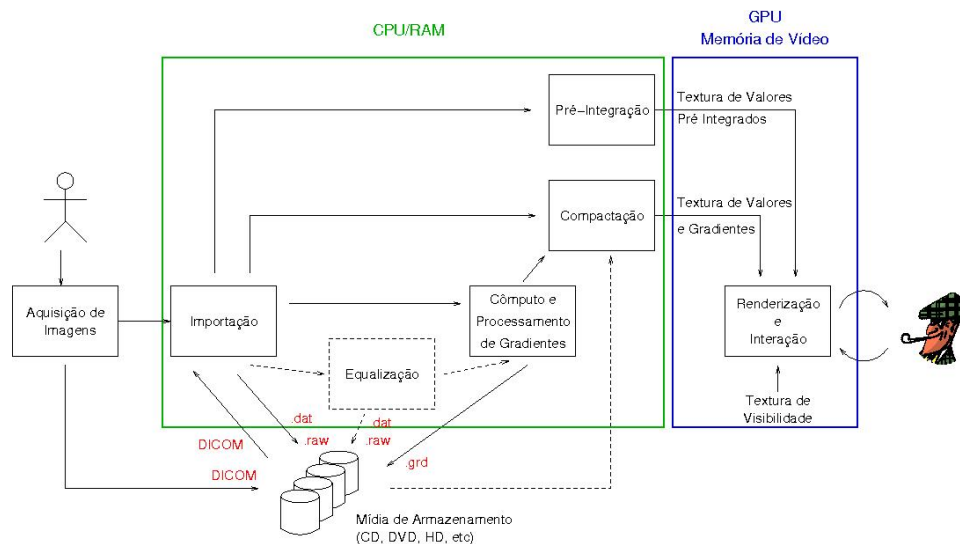


Figura 1: Fluxo de dados.

Observe que uma parte de processamento ocorre na unidade de processamento central (CPU) e a outra, na unidade de processamento gráfico (GPU). Adicionalmente, alguns dados processados são armazenados em HD

para uso posterior. Enfim, para que o aplicativo funcione corretamente, é necessário dispormos de espaços em HD, RAM e memória de vídeo.

## 2 Requerimento de espaço para MRI 3T

Com base no que foi exposto na seção 1, podemos estimar o espaço necessário para visualizar **um** exame de MRI. Tomamos como exemplo o exame que nos foi entregue pelo eng. Felipe Bergo através de um DVD. Este exame contém 994 arquivos. Destes 994 arquivos foram identificados quatro séries que contém imagens, conforme sintetiza a Tabela 1.

Tabela 1: Estimativa do espaço requerido por série de imagens.

| Série            | $VolSize \times nBytes$              | HD       | RAM      | vídeo    |
|------------------|--------------------------------------|----------|----------|----------|
| VVBM SENSEE      | $512 \times 512 \times 40 \times 2$  | > 84 MB  | > 367 MB | > 52 MB  |
| VBM              | $512 \times 512 \times 356 \times 2$ | > 746 MB | > 1.3 GB | > 466 MB |
| FLAIR 3D         | $400 \times 400 \times 320 \times 2$ | > 409 MB | > 716 MB | > 256 MB |
| COR T2 MULTI-ECO | $512 \times 512 \times 240 \times 2$ | > 503 MB | > 880 MB | > 314 MB |

Foi rodado o aplicativo em uma máquina provida de placa GeForce NVidia 8600 com 512 MB, 1G de memória RAM e 200G de HD com espaço livre. Como esperado, só conseguimos visualizar a série VVBM SENSEE. Como uma tentativa de contornar o problema, foi testada uma nova forma de organização de dados, conforme detalha a Seção 3.

## 3 Uma Proposta de Reestruturação do Aplicativo

São dois *bottlenecks* no fluxo de dados do aplicativo: disponibilidade de memória RAM e disponibilidade de memória na placa de vídeo. Como solução, propomos os seguintes passos de solução a curtíssimo prazo, até final do mês de fevereiro/2010:

**falta de espaço em RAM:** armazenar em HD os dados intermediários, resultantes do cômputo de gradientes, equalização, etc.

**falta de espaço na memória de vídeo:** quantizar as imagens para a metade da resolução, ou seja, em 8 bits.

Figura 2 ilustra a proposta da re-estruturação. A primeira solução já foi testada e conseguimos gerar todos os arquivos de gradientes do exame em um pouco mais de 1 hora em uma máquina provida de processador AMD Athlon 64, 1GB de memória RAM e placa de vídeo GeForce 6600GT 128MB. Em relação à solução da falta de espaço na memória de vídeo, geraremos umas imagens para os neuro-cientistas/neuro-cirurgiões avaliarem a qualidade do resultado.

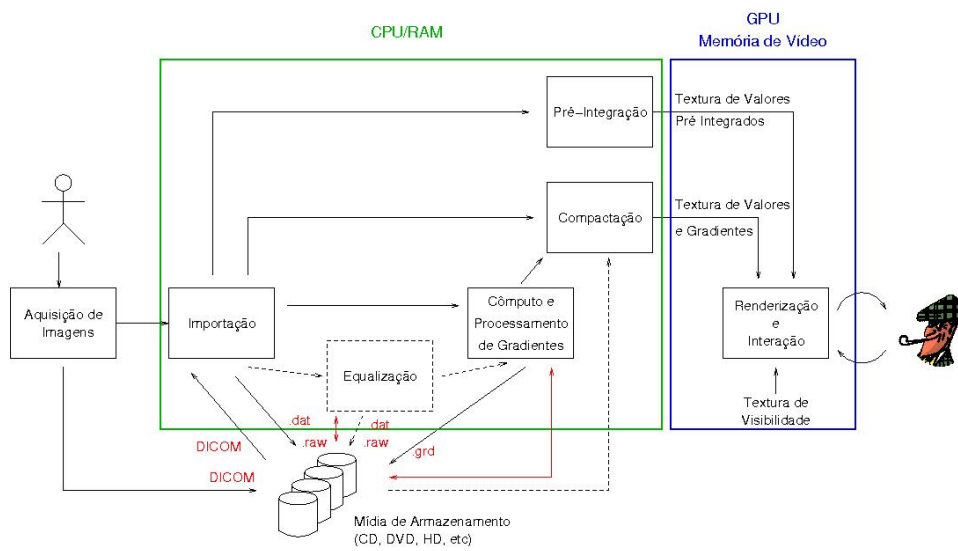


Figura 2: Fluxo de dados reestruturado.