

# NAG - Núcleo para Ambiente Gráfico: Sistema de Janelas e Toolkit para a Visualização e Manipulação de Imagens

Agma J. M. Traina (\*) Jan F. W. Slaets (\*\*) Lírio O. B. Almeida (\*\*\*)

(\*) Instituto de Ciências Matemáticas de São Carlos - Univ. São Paulo  
Dep. Ciências de Computação e Estatística  
Cx. Postal 668 - CEP 13560-970 - São Carlos - São Paulo - Brazil  
e. mail: agma@uspfc.ifqsc.usp.ansp.br

(\*\*) Instituto de Física e Química de São Carlos - Univ. São Paulo  
Dep. Física e Ciência dos Materiais  
Grupo de Instrumentação e Informática  
Cx. Postal 369- CEP 13560-970 - São Carlos - São Paulo - Brazil  
e. mail: jan@uspfc.ifqsc.usp.ansp.br  
e. mail: lirio@uspfc.ifqsc.usp.ansp.br

## Abstract

This work describes the development of a dedicated system, used for visualization and manipulation of Magnetic Resonance (MR) images. A dedicated TMS340XX graphics processor has been used to implement the software of the graphic environment as well as the toolkit for the user interface. The software designed for this system consists of a compact kernel, with all needed primitives to support the creation and manipulation of windows and menus directly. This kernel exploits fully the resources of the graphics processor, and provides an interface for development of applications in "C" Language.

The developed environment is fundamental for the implementation of a user friendly interface, built to manipulate and visualize tomographic images for medical diagnose.

## 1. Introdução

Este trabalho apresenta um sistema de software desenvolvido para apoiar a construção de aplicações gráficas interativas "amigáveis", concebido especificamente para sistemas de visualização de imagens. Este sistema faz parte do projeto de desenvolvimento de um tomógrafo por Ressonância Magnética (RM) com tecnologia nacional, realizado integralmente no Instituto de Física e Química de São Carlos. Este trabalho descreve especificamente o módulo de visualização e manipulação das imagens obtidas pelo tomógrafo já previamente reconstruídas [Traina (1991a)]. Uma arquitetura gráfica dedicada a visualização de imagens também foi desenvolvida, contendo recursos específicos e desejáveis a visualização e manipulação de imagens obtidas pelo tomógrafo [Paiva (1990)].

## 2. Metodologia Utilizada para o Desenvolvimento do Núcleo para Ambiente Gráfico - NAG

Primeiramente desenvolveu-se uma arquitetura dedicada

a visualização de imagens tomográficas, baseada no processador TMS34010 da Texas Instruments [Paiva (1990)]. Dessa forma, todas as camadas de software necessárias ao desenvolvimento de aplicativos de visualização e manipulação de imagens foram também desenvolvidas [Traina (1991b)]. O objetivo deste projeto foi a elaboração de um sistema dedicado, capaz de permitir a visualização e manipulação das imagens obtidas pelo tomógrafo por Ressonância Magnética, em desenvolvimento no Departamento de Física e Ciência dos Materiais do IFQSC [Panepucci et al (1987)] [Tannús (1987)].

Considerando-se as particularidades das aplicações para visualização do sistema de tomografia por RM, bem como as características do TMS34010, optou-se pelo desenvolvimento de um ambiente gráfico próprio. Essa decisão foi tomada pois assim o "software" resultante aproveitaria melhor as características do "hardware"; e seria também compatível com as necessidades do sistema de tomografia, resultando numa relação custo/benefício

maior do que se fosse utilizada uma placa gráfica comercial ou similar.

O ambiente gráfico desenvolvido fornece então um "toolkit" para a implementação de aplicativos interativos "amigáveis", como é o caso da visualização e manipulação de imagens tomográficas.

### 3. Módulos que Compõem o Ambiente Gráfico.

O ambiente gráfico é composto pelos seguintes elementos: *Gerenciador de Janelas*; *Gerenciador de "menus"*; *Subsistema de Comunicação com o Hospedeiro* e *Subsistema de Visualização de Imagens*, os quais serão descritos a seguir.

#### Gerenciador de Janelas.

A principal estrutura do ambiente gráfico desenvolvido é a de *Janela*. Uma *Janela* é uma região retangular da tela, a qual é definida pelas coordenadas de seus quatro cantos. A indicação dessas coordenadas é padronizada em todo o ambiente como sendo a indicação de seu canto superior esquerdo e o canto inferior direito.

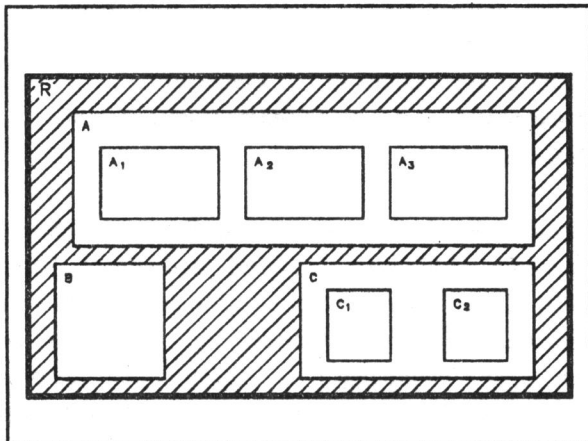


Figura 1 - Exemplo de uma tela com janelas.

As janelas se associam de forma hierárquica, onde a janela principal (raiz da hierarquia de janelas), é a própria tela do monitor do sistema de visualização. A figura 1 mostra um exemplo de como poderia ficar a visualização de um sistema constituído por 9 janelas. Nesse exemplo, a tela é a janela raiz da hierarquia (R), e possui três janelas filhas: A, B e C. Por sua vez, a janela A possui três janelas filhas: A1, A2 e A3; a janela B não possui filhas; e a janela C possui duas filhas: C1 e C2.

A maneira como as janelas podem se associar deve respeitar algumas restrições:

- Uma janela não pode ultrapassar os limites da janela que é seu pai, ou seja, as janelas A1; A2 e A3 são

inscritas na janela pai (A), assim como as janelas A, B e C são inscritas na janela raiz (tela).

- Os atributos de uma janela pai são transmitidos à filha, a menos que sejam explicitamente definidos na hora da criação. Esses atributos podem ser: tamanho e posição da janela, eventos permitidos, quais os planos de memória utilizados, cor de frente, cor de fundo, posição do cursor, arquivo de caracteres fonte.
- O sistema impõe que sempre uma janela deve ser considerada como a *janela corrente*.
- Existem operações que somente são realizadas sobre a *janela corrente*. Um exemplo é a criação de uma outra janela: uma nova janela somente pode ser criada como filha da *janela corrente*.

O sistema fornece dois tipos básicos de janelas (em termos de aparência): Janela com borda e Janela sem borda.

Como ao criar-se uma janela o Gerenciador de janelas transmite os atributos de uma janela pai à janela filha, alterando-se apenas os atributos explicitamente referenciados, as primitivas que efetuam a criação de janelas - *Cria\_janela* e *Cria\_jan\_borda* - recebem como parâmetros pares *atributo-valor*, os quais serão atribuídos à janela criada.

#### Gerenciador de "menus".

Basicamente os "menus" podem ser classificados em dois tipos:

- "menus" estáticos: que são permanentemente visíveis. Estes "menus" são colocados uma vez na tela quando o aplicativo começa a ser executado e só são retirados quando o aplicativo termina.
- "menus" dinâmicos: tornam-se visíveis apenas quando são requisitados. Os tipos mais conhecidos, como já visto anteriormente, são os chamados "menus" tipo "pull-down" e "pop-up".

O ambiente gráfico desenvolvido conta com "menus" do tipo "pull-down" e "pop-up" [Foley et al. (1990)], os quais podem ser colocados de forma hierárquica. A manutenção ou não de um "menu" na tela é opção do aplicativo.

No ambiente gráfico desenvolvido, o "menu" "pull\_down" não precisa mostrar simultaneamente todos os itens, permitindo especificar na criação do "menu" quantos itens serão mostrados ao mesmo tempo, e os demais itens serão mostrados conforme o usuário solicitar, através do rolamento do menu.

Quando um "menu" é criado, o ambiente gráfico coloca-o sobre uma janela (janela com borda, do tipo

borda-singela), e o Gerenciador de Janelas passa também a ter controle sobre o menu. Isso é feito para simplificar a forma de gerenciar "menus", utilizando-se os recursos já descritos para o gerenciamento de janelas.

**Subsistema de Comunicação com o Hospedeiro.**

A arquitetura dedicada desenvolvida baseada no TMS34010 (GSP), utiliza um microcomputador IBM-PC como hospedeiro, pois esta é uma máquina altamente disponível no mercado nacional e de baixo custo. A capacidade de transferir grandes blocos de dados para o GSP torna a interface com o hospedeiro bastante ágil. Por exemplo, o hospedeiro pode transferir blocos de comandos para o GSP, pode parar o GSP temporariamente para carregar um novo programa para o GSP executar, ou mesmo pode ler blocos de dados gráficos gerados pelo GSP.

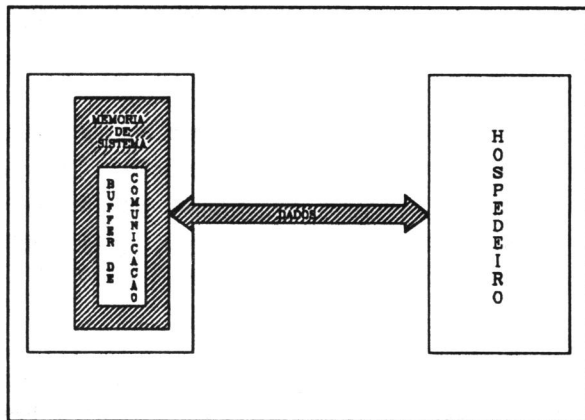


Figura 2 - Esquema de comunicação entre o GSP e o hospedeiro.

A interface de comunicação com o hospedeiro pode ser acessada tanto pelo hospedeiro quanto pelo GSP, ocupando registradores que são mapeados ou em 4 regiões consecutivas de 16 bits no espaço de endereço de memória do GSP, ou em registradores de E/S [Texas (1986)].

Utilizando-se como base uma ferramenta de carregamento de dados no GSP, criou-se uma nova ferramenta chamada *SDBLR*, a qual após carregar o programa objeto na memória local do GSP, permanece residente no hospedeiro, monitorando um "buffer" de comunicação colocado entre o GSP e o hospedeiro. Este "buffer" possui 20 bytes, pois dessa forma acomoda perfeitamente a estrutura de dados utilizada para controlar a comunicação, a qual contém os campos de controle para os dispositivos do hospedeiro. Na figura 2 é ilustrado o esquema de comunicação entre o GSP e o

hospedeiro através do "buffer" de comunicação.

Além disso foi necessário desenvolver um subsistema de comunicação com o hospedeiro, o qual gerenciaria todas as solicitações de operações a disco e E/S executadas nos periféricos do hospedeiro, transmitindo e recebendo os dados envolvidos. Os recursos do hospedeiro gerenciados são: teclado, vídeo e disco.

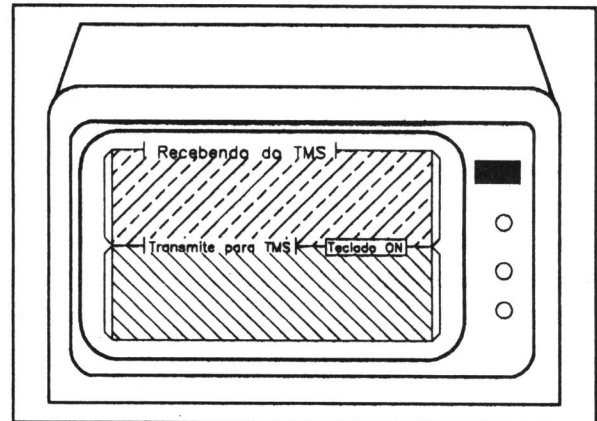


Figura 3 - Organização da tela do monitor do hospedeiro, quando o SDBLR está em execução.

Para controlar a comunicação entre o hospedeiro e o GSP, foi construído um ambiente de gerenciamento no hospedeiro. Assim, a tela do hospedeiro apresenta três janelas: uma janela na parte superior que recebe os dados enviados pelo GSP para serem mostrados na tela do hospedeiro; uma janela na parte inferior que mostra os dados que o hospedeiro está transmitindo para o GSP, e uma janela pequena no centro que indica se o teclado está transmitindo para o TMS (ON), ou se está a serviço do próprio hospedeiro (LOCAL). A figura 3 mostra como a tela do hospedeiro é organizada pelo subsistema de Comunicação.

O subsistema de Comunicação pode ser dividido em duas partes, uma sendo executada no hospedeiro e a outra no GSP, que sincronizada com a primeira executam as operações complementares à comunicação solicitada.

**Subsistema de Transferência de Imagens Tomográficas de RM.**

A operação de transferência das imagens foi colocada em um subsistema à parte, apesar de necessitar de comunicação entre o hospedeiro e o GSP, pois foi necessário fazer-se um tratamento diferenciado sobre os dados que compõem a arquivo de imagem.

A primitiva que efetua a operação de leitura de uma imagem no disco do hospedeiro, e a transfere para o

GSP foi chamada de `i_le_imagem`. Esta primitiva abre o arquivo de imagem no hospedeiro e o transfere para o GSP. O GSP cria uma janela com borda que comporte exatamente a imagem, e coloca a imagem na janela para que o usuário possa manipulá-la.

#### 4. Conclusões

Atualmente as imagens obtidas pelo tomógrafo de RM do IFQSC, são reconstruídas através de uma Transformada Rápida de Fourier (FFT) e podem ser visualizadas na arquitetura gráfica dedicada. As imagens são armazenadas em disco no hospedeiro, podendo ser acessadas e visualizadas a qualquer instante. Aplicativos para visualização e manipulação das imagens foram implementados a fim de se validar o ambiente gráfico desenvolvido. Apesar de ser um sistema básico (atualmente é constituído por em torno de 30 primitivas fundamentais), o ambiente gráfico supre as necessidades de sistemas dessa natureza.

A relação custo/benefício de um sistema deste tipo que será integrado ao tomógrafo por RM é um fator preponderante. Assim, para o sistema de visualização de imagens tomográficas buscou-se desenvolver um ambiente gráfico próprio que fosse adequado tanto ao processo de visualização quanto a arquitetura gráfica dedicada utilizada. Foi necessário dimensionar o software de acordo com as possibilidades de bom funcionamento do hardware. Ou seja, por que não utilizou-se sistemas de janelas já consagrados? Porque estes sistemas são grandes e complexos, necessitando de máquinas com poder computacional elevado para poderem executar convenientemente. O nosso objetivo é o de executar primeiramente o sistema de visualização e processamento de imagens tomográficas, não gastando o poder computacional da máquina somente para gerenciar a interface Homem/Máquina do sistema, o que ocorreria se fosse utilizado um ambiente padronizado nesta configuração atual da arquitetura.

O NAG suporta um conjunto poderoso de recursos úteis a visualização de imagens tomográficas, permitindo a construção de aplicativos de maneira bastante facilitada (no que diz respeito a interface com o usuário). O ambiente provê recursos para a utilização de "menus", janelas de diversas formas e estados, com tratamento de "clipping", "scroll", cor, planos de bits, eventos, etc. Além disso, dentro do arquitetura dedicada oferece acesso aos periféricos do hospedeiro, tais como: arquivos em disco, teclado e vídeo. Entretanto, todo esse suporte não requer mais do que 128 Kbytes de memória de

trabalho para ser executado. Isso jamais seria conseguido utilizando-se um sistema gráfico comercial que desse o mesmo suporte.

Este projeto envolveu o desenvolvimento de um ambiente gráfico completo, dispondo de um Sistema de Gerenciamento de Janelas, com seu próprio sistema de janelas e todos os "drivers" específicos para a arquitetura em questão.

Deve-se ressaltar que nas estruturas internas desenvolvidas para o ambiente gráfico foram previstos diversos elementos com vistas a futuras ampliações, acompanhando novas versões ampliadas do hardware da arquitetura dedicada, que já estão sendo testadas no IFQSC. Uma nova versão da arquitetura dedicada com um membro mais recente da família de processadores TMS340, o TMS34020 está sendo implementada. Algumas expansões solicitadas pelo sistema de tomografia, já foram realizadas no NAG de maneira bastante simples, o que ajudou a validar a sua modularidade.

As rotinas que constituem o ambiente gráfico foram desenvolvidas em linguagem de programação "C" na sua grande maioria, contudo as rotinas que acessam diretamente os recursos do GSP foram escritas em linguagem de montagem do processador TMS34010.

#### 5. Bibliografia

- A.J.M. Traina (a), **Estudo e Implementação de Software Dedicado para um Sistema de Visualização de Imagens**, *Tese de Doutorado* apresentada ao Instituto de Física e Química de São Carlos - USP, 1991.
- A.J.M. Traina (b), J.F.W. Slaets, L.O.B. Almeida, **Um Ambiente Gráfico para a Visualização de Imagens Tomográficas Obtidas por Ressonância Magnética**, in *Anais do VI SIBGRAPI*, pp.215-218.
- M.S.V. Paiva, **Projeto de Uma Arquitetura de Hardware para Visualização de Imagens Digitais**, *Tese de Doutorado* apresentada ao Instituto de Física e Química de São Carlos - USP, 1990.
- H. Panepuccci, et alli, **Novas Imagens do Corpo - Tomografia por Ressonância Magnética**, *Ciência Hoje*, 4, nº 20, 46-56.
- A. Tannús, **Desenvolvimento de Tecnologia de Tomografia por Ressonância Magnética Nuclear**, *Tese de Doutorado* apresentada ao IFQSC-USP (1987).
- Texas Instruments, **TMS34010 User's Guide**, 1986.
- J.D. Foley, J.D., A. van Dam, S.K. Feiner, J.F. Hughes, **Computer Graphics - Principles and Practice**, Second Edition, Addison-Wesley Publishing Company, 1990.