

BIBLIOGRAFIA

1. Microsoft Windows Source Compiler, User's Guide.
2. Microsoft Windows Software Development Kit, Environment and Tools.
3. Microsoft Windows Software Development Kit, Guide to Programing.
4. MURRAY III, W. H. e PAPPAS, C.H. Programação para Windows versão 3, McGraw Hill.
5. FOLEY, J. A., VAN DAM, J. F. and HUGHES, J. Computer Graphics - Principles and Practice, Addison - Wesley Company, Reading MA (1990).

PAINEL 15

Desenvolvimento de Interface Gráfica para Visualização de Moléculas Tridimensionais

DESTAQUE

Patrícia Alves Dias

Bolsista de Inic. Científica, Informática, UFRJ

Paulo Sérgio da Silva Pinto

Orientador, Químico, D.Sc.

Márcia Viana de Sá Earp

Co-orientadora, Analista de Sistemas

1. INTRODUÇÃO

A modelagem molecular por computador tem a finalidade de simular estruturas e suas respectivas interações. A Computação Gráfica (1) é a ferramenta ideal para a visualização das mudanças ocorridas em um sistema em função do tempo, além de possibilitar a visualização das estruturas, a representação das moléculas e a manipulação interativa de modelos geométricos. Assim, tomando como base *softwares* acadêmicos de construção e cálculo de propriedades químicas, foi desenvolvida uma interface gráfica para a visualização das estruturas químicas geradas em uma *workstation Silicon Graphics*.

As moléculas são modeladas geometricamente, a partir de dados tabulares obtidos através de seus átomos e das ligações entre eles. Foram utilizadas técnicas de computação gráfica para o desenvolvimento dos algoritmos de simulação tridimensional das moléculas e de manipulação dos modelos, possibilitando efeitos como *zoom* e rotação.

2. METODOLOGIA

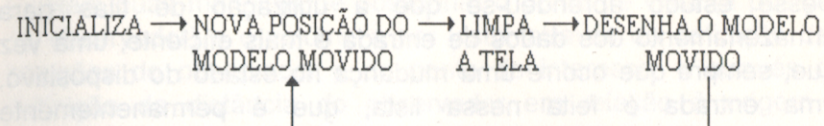
Inicialmente, o trabalho se concentrou no estudo das técnicas de programação no sistema operacional *unix-like* IRIS da *Silicon Graphics* e em como usar sub-rotinas e funções da *Graphics Library* para escrever aplicações gráficas de desenho de sólidos coloridos bi ou tridimensionais em movimento e em tempo real, sob controle interativo dos dispositivos de entrada de dados.

Foram estudados comandos UNIX e utilitários para criar e modificar o código fonte de aplicações, além de comandos de manipulação de diretórios e arquivos, compilação e execução de aplicações.

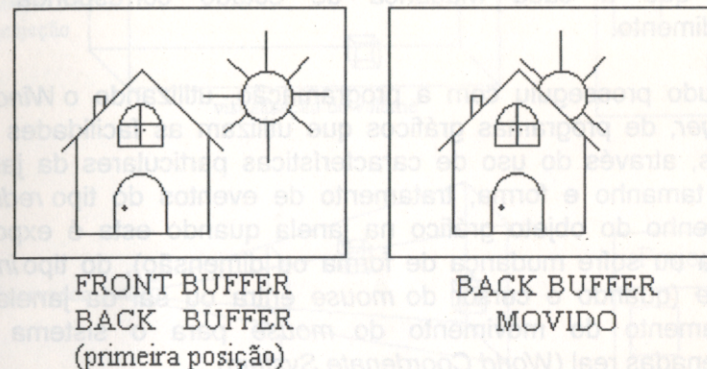
Após estes conceitos básicos de utilização do sistema operacional, foram feitos estudos de como visualizar e criar objetos bi e tridimensionais no plano XY, para desenvolvimento de programas de visualização de cenas compostas de objetos coloridos, através do uso de sub-rotinas, funções específicas de desenho de objetos gráficos (círculos, arcos, linhas, polígonos) e texto, de definição de cores e padrões de textura, e transformação do sistema de coordenadas real (*World Coordinate System*) para o sistema de coordenadas do computador (*Window Coordinate System*).

Foi simulada a movimentação de modelos geométricos, através do uso de algoritmos pré-definidos e estudadas formas de prevenir o efeito de oscilação das imagens com o uso da técnica de *Double Buffering*.

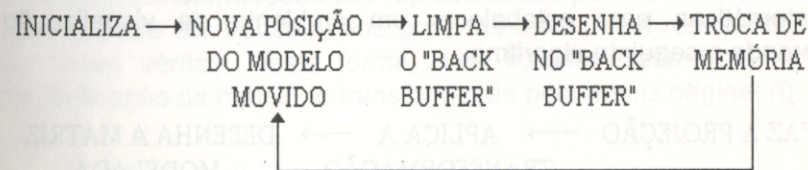
Inicialmente o processo de movimentação era feito através do seguinte esquema :



Com a utilização da técnica de *Double Buffering*, a cena que vai ser movimentada é dividida em duas, a parte de fundo (*back buffer*) e a parte da frente (*front buffer*).



Desta forma o esquema passa a ser o seguinte :



Após isto, foi feito um estudo da metodologia de leitura e utilização dos dados obtidos dos dispositivos de entrada (*mouse*, teclado), com a utilização do conceito de fila de entrada (*input queuing*) para armazenamento desses dados, diferenciando os métodos de processamento da entrada: empilhamento (*polling*) e enfileiramento (*queuing*).

Desse estudo aprendeu-se que a utilização de filas para armazenamento dos dados de entrada é mais eficiente, uma vez que, sempre que ocorre uma mudança no estado do dispositivo, uma entrada é feita nessa lista, que é permanentemente examinada, ao contrário de outro método (*polling*), que necessita de procedimentos do programador para armazenar o estado do dispositivo. Assim, é utilizada uma fila de eventos do tipo FIFO (*First In First Out*) para armazenar o estado do dispositivo de entrada e um algoritmo de recuperação dessa informação, de modo que a cada mudança de estado corresponda um procedimento.

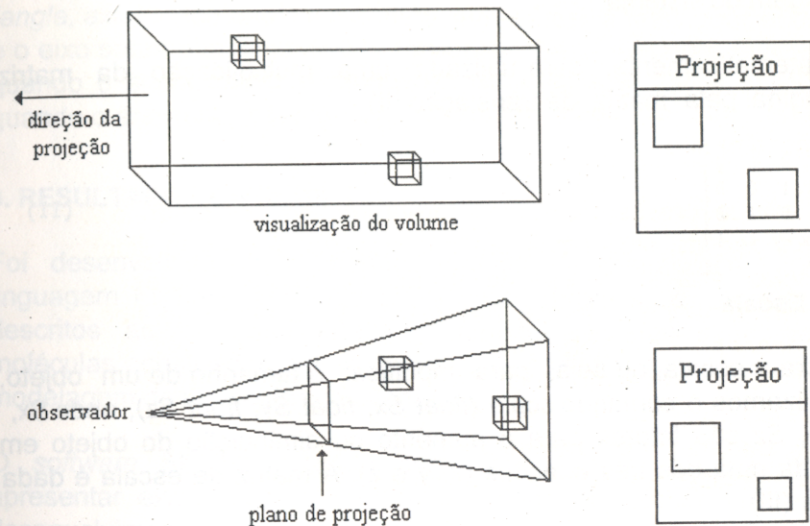
O estudo prosseguiu com a programação, utilizando o *Window Manager*, de programas gráficos que utilizam as facilidades das janelas, através do uso de características particulares da janela como tamanho e forma, tratamento de eventos do tipo *redraw* (redesenho do objeto gráfico na janela quando esta é exposta, movida ou sofre mudança de forma ou dimensão), do tipo *input change* (quando o cursor do *mouse* entra ou sai da janela) e mapeamento do movimento do *mouse* para o sistema de coordenadas real (*World Coordinate System*).

Pensando na visualização, estudou-se como criar, visualizar e animar modelos tridimensionais, usando transformações matemáticas para estabelecer um ambiente de visualização, gerando o seguinte algoritmo :

FAZ A PROJEÇÃO → APLICA A → DESENHA A MATRIZ
TRANSFORMAÇÃO MODELADA

Existem dois tipos de projeção: a projeção em perspectiva e a projeção paralela ortográfica. Na projeção em perspectiva os objetos mais distantes aparecem com tamanho menor do que os mais próximos, e na projeção paralela ortográfica os objetos aparecem como se estivessem à mesma distância do observador. Assim, as projeções em perspectiva são mais realistas, uma vez que permitem a ilusão de profundidade.

Existem duas maneiras de posicionar o ponto de vista (a posição do observador) e o volume a ser visto: através da definição das posições do observador e do ponto de interesse, e através da definição da distância do observador em relação à origem e orientação.



Todas as transformações são representadas por matrizes 4X4. Um vértice pode ser transformado por uma matriz 4X4 resultando em um novo vértice. Desta forma, uma transformação trata-se de multiplicação da matriz de transformação pela matriz original (I).

$$[x' y' z' w'] = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{32} & m_{33} \end{bmatrix} [x y z w] \quad (I)$$

Isto permite que sejam executados os seguintes procedimentos :

a) Translação

Estando todos os pontos no formato de uma matriz criada pelo sistema, para executar uma translação utilizamos o comando *translate (float Tx, float Ty, float Tz)*, onde *Tx*, *Ty*, *Tz* são as coordenadas que representam o deslocamento a partir do ponto de origem do sistema.

Matematicamente, será utilizada uma multiplicação da matriz original pela matriz de translação (II).

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \quad (II)$$

b) Escala

Para a escala, ou seja, para modificar o tamanho de um objeto, utilizamos o comando *scale (float Sx, float Sy, float Sz)*, onde *Sx*, *Sy*, *Sz* são fatores para o aumento ou diminuição do objeto em cada uma das três direções (x, y e z). A matriz de escala é dada por (III):

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III)$$

c) Rotação

O processo de rotação funciona de maneira semelhante ao processo de escala, só que em relação a cada um dos três eixos, ou seja, um objeto pode ser rotacionado em relação ao eixo das coordenadas X ou em relação ao eixo das coordenadas Y ou em relação ao eixo das coordenadas Z. Logo, existem três matrizes para solucionar esse problema (IV).

$$R_x(\theta) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad R_y(\theta) \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_z(\theta) \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (IV)$$

Dois comandos são utilizados para efetuar a rotação: o *rotate (angle, axis)*, onde *angle* é o ângulo que se deseja rotacionar e *axis* é o eixo sobre o qual se deseja rotacionar. Esse comando é usado quando o ângulo está em dezenas de graus e *rot (angle, axis)*, quando o ângulo está em graus.

3. RESULTADOS OBTIDOS

Foi desenvolvida uma interface gráfica de visualização, em linguagem Fortran, utilizando a *Graphics Library* e os conceitos descritos acima. Essa interface permite a visualização das moléculas construídas pelo *software SD - software acadêmico de modelagem de moléculas covalentes*.

O *software SD* foi utilizado como base deste trabalho por apresentar exatamente o caso mais crítico da necessidade do desenvolvimento da interface gráfica. Sendo um *software* desenvolvido em um *mainframe IBM*, ele necessita dessa interface para que os pesquisadores possam visualizar suas moléculas de forma interativa, em uma *workstation Silicon Graphics*.

A próxima etapa deste trabalho é o desenvolvimento dessa interface segundo o padrão da Open GL - nova biblioteca gráfica da *Silicon Graphics* (2, 3) que segue o padrão OSF - e utilizando o *software Motif* para o gerenciamento de janelas (4).

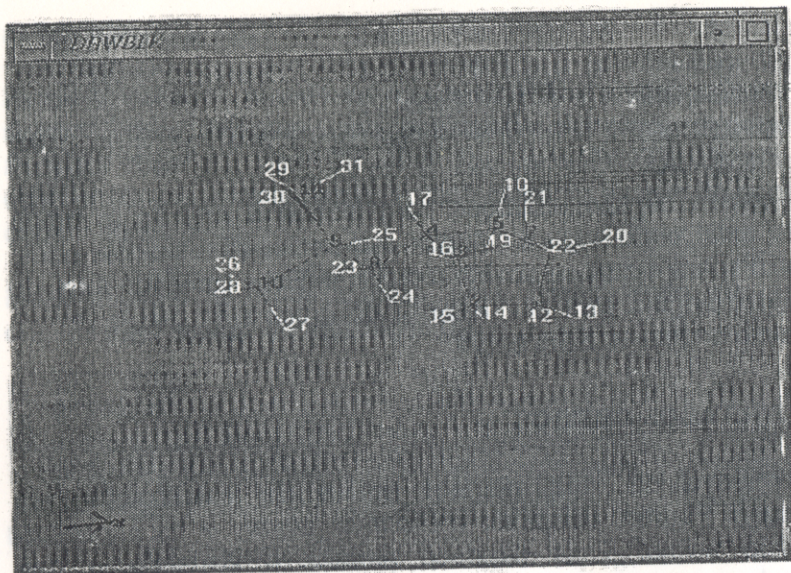


Figura 1 - Impressão da tela do software desenvolvido editando uma molécula construída no software SD

BIBLIOGRAFIA

1. FOLEY, J.A., VAN DAM, S.F. and HUGHES, J. Computer Graphics - Principles and Practice, Addison-Wesley Publishing Company, Reading, MA.; (1990) pp. 858 - 859
2. GL Programming 1, Supplement 1, Fortran Code Supplement - Student's Workbook, Silicon Graphics Computer Systems (1991)
3. GL Programming 2 - Student's Workbook, Silicon Graphics Computer Systems (1991)
4. PowerVision Graphics Programming - Student's Workbook, Silicon Graphics Computer Systems (1990)
5. NYE, A. - Xlib Programming Manual, O'Reilly & Associates, Inc. (1990)

PAINEL 16

*Banco de Estruturas
Moleculares*

DESTAQUE

Vanessa Paranhos Türner
Bolsista de Inic. Científica, Informática,
UFRJ

Peter Rudolf Seidl
Orientador, Químico, D.sc.

Márcia Viana de Sá Earp
Co-orientadora, Analista de Sistemas

1. INTRODUÇÃO

A modelagem molecular por computador é atualmente considerada um importante método de pesquisa química. A moderna tecnologia existente torna possível a simulação e estimativa de estruturas moleculares. Porém, a grande quantidade de informações tratadas durante o estudo de uma família de moléculas torna inviável a utilização de meios não computacionais de armazenamento e recuperação desses dados.

Isto foi observado durante a execução do trabalho "Construção, Cálculo e Catalogação de Famílias de Moléculas Tensionadas Derivadas do Norbonano Através da Modelagem Molecular" (1). Este trabalho catalogou dezessete moléculas e gerou um catálogo em papel impresso com aproximadamente 500 páginas, fato que dificulta consideravelmente o acesso às informações catalogadas.