

EVANDRO DE BARROS COSTA

**MOL3D : UM SISTEMA GRAFICO DE APOIO
A VISUALIZAÇÃO DO COMPORTAMENTO
ESPACIAL DE ESTRUTURAS MOLECULARES**

Dissertação apresentada ao curso de MESTRADO EM
ENGENHARIA ELETRICA da Universidade Federal da
Paraíba, em cumprimento às exigências para obten-
ção do Grau de Mestre.

PAULO ROBERTO C. DE ARAUJO
Orientador (DEE)

JOSE HAMURABI NOBREGA DE MEDEIROS
Co-orientador (DSC)

MARIA DO SOCORRO DE LACERDA
Co-orientadora (DEQ)

CAMPINA GRANDE
OUTUBRO - 1989



C837m Costa, Evandro de Barros
Mol3D : um sistema grafico de apoio a visualizacao do comportamento espacial de estruturas moleculares / Evandro de Barros Costa. - Campina Grande, 1989.
76 f. : il.

Dissertacao (Mestrado em Engenharia Eletrica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

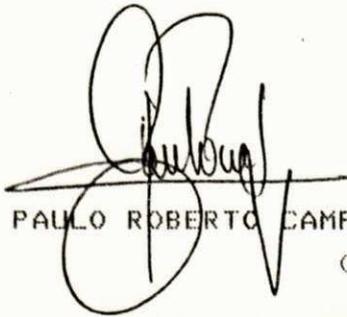
1. Software Grafico 2. Dissertacao I. Araujo, Paulo Roberto Campos de, M.Sc. II. Medeiros, Jose Hamurabi Nobrega de, M.Sc. III. Lacerda, Maria do Socorro de IV. Universidade Federal da Paraiba - Campina Grande (PB) V. Título

CDU 004.41(043)

UM SISTEMA GRÁFICO DE APOIO À VISUALIZAÇÃO
DO COMPORTAMENTO ESPACIAL DE ESTRUTURAS MOLECULARES

EVANDRO DE BARROS COSTA

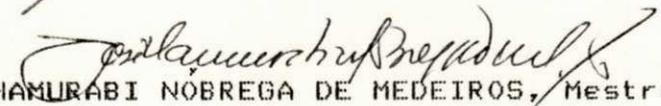
DISSERTAÇÃO APROVADA 17-11-89



PAULO ROBERTO CAMPOS DE ARAÚJO, Mestre, UFPB
Orientador



JOÃO MARQUES DE CARVALHO, Ph.D., UFPB
Componente da Banca



JOSÉ HAMURABI NOBREGA DE MEDEIROS, Mestre, UFPB
Componente da Banca

Maria de Fatima R.V. Turrell
MARIA DE FATIMA QUEIROZ VIEIRA, Ph.D., UFPB
Componente da Banca

CAMPINA GRANDE - PB

NOVEMBRO - 1989

224 24-11-89

AGRADECIMENTOS,

Agradeço em primeiro lugar , ao criador do universo e aos meus pais.

Agradeço a todos aqueles que , de alguma forma, contribuíram para a realização deste trabalho particularmente aos colegas de graduação e pós-graduação em informática.

Agradeço, em particular, ao Prof. Paulo Roberto Campos de Araújo, pelo incentivo e pela orientação segura e eficiente no desenrolar do trabalho.

RESUMO

O presente trabalho descreve a concepção e a implementação de um "software" gráfico. Trata-se do "Sistema Gráfico de Apoio à Visualização do Comportamento Espacial de Estruturas Moleculares" (MOL3D). MOL3D tem seus algoritmos implementados na Linguagem Pascal, sob o Sistema Turbo Pascal, rodando em microcomputadores da linha IBM-PC.

ABSTRACT

The present work describes the conception and implementation of a graphical software, here called a "Graphical System to Help visualizing the Spacial Behavior of Molecular Structures" (MOL3D). MOL3D has its algorithms implemented in Pascal (Turbo Pascal), running on an IBM-PC compatible micro-computer.

SUMARIO

1. INTRODUÇÃO	1
1.1. Objetivos do Trabalho.....	1
1.2. Visão Geral.....	2
1.3. Descrição Geral de Funcionamento.....	6
1.4. Conteúdo do Trabalho.....	10
2. TECNICAS DE VISUALIZAÇÃO DE SUPERFÍCIES MOLECULARES.....	12
2.1. Uniões de Esferas.....	12
3. CONCEITOS FUNDAMENTAIS.....	19
3.1. Arquivos Externos.....	19
3.2. Estruturas de Dados.....	21
3.3. Interface com o Usuário.....	23
4. OPERAÇÕES SOBRE ARQUIVOS.....	28
4.1. Operações Sobre Arquivos de Objetos.....	28
4.2. Operações Sobre Arquivos de Telas.....	30
5. OPERAÇÕES DE VISUALIZAÇÃO.....	32
5.1. Posicionamento do Observador.....	32
5.2. Projeções.....	34
5.3. Apagamento de Esferas e Conexões Não Visíveis	39
5.4. Deslizamento.....	47
5.5. Ampliação e Redução("ZOOM").....	48
5.6. Sistemas de Coordenadas.....	50

5.7. Pintura de Esferas.....	53
5.8. Visualização de Eixos.....	57
5.9. Rolamento de Tela.....	57
5.10. Traçado das Conexões Interatômicas.....	60
5.11. Atribuição de Cores.....	63
5.12. Animação.....	64
6. CONCLUSÃO.....	66
7. REFERENCIAS BIBLIOGRAFICAS.....	68
Apêndice A.....	70
Apêndice B.....	72
Apêndice C.....	73

CAPITULO I

INTRODUÇÃO

Este capítulo se propõe a apresentar ao leitor o propósito geral do trabalho. Para isto, o capítulo foi dividido em quatro partes : Objetivos do Trabalho, Visão Geral do Trabalho, Descrição de Funcionamento do Sistema e Conteúdo do Trabalho.

1.1. Objetivos do Trabalho

Sistemas computacionais para visualização de estruturas moleculares estão se tornando muito importantes como ferramenta auxiliar no ensino e na pesquisa, nas áreas de química, de bioquímica e outras áreas correlacionadas.

Dada a importância acima, resolveu-se partir para a especificação e implementação de um software, cujo propósito geral seria o de fornecer ao usuário recursos adequados para se lidar com o fenômeno espacial das moléculas (*).

Procurou-se, assim, desenvolver um ambiente confortável, onde o usuário contasse com uma interface amigável, biblioteca de aminoácidos e de moléculas, recursos para armazenamento e leitura de telas, visualização e simulação dinâmica dos aminoácidos envolvidos na molécula, ou simulação dinâmica da própria molécula.

(*) O termo molécula será utilizado, indistintamente, em todo este texto para denotar todas as moléculas em geral, especificamente as macromoléculas, e muito especialmente as proteínas (que são casos particulares de macromoléculas).

Com o advento da evolução das estações gráficas, surgem possibilidades de implementação de modelos físicos de representação de estruturas moleculares de diferentes tipos. O uso destes recursos é comum em grandes centros de pesquisa.

A maior parte do "software" existente (até onde se tem informação) tem sido desenvolvido nos próprios laboratórios de pesquisa em química, havendo pouco "software" para este fim, disponível no mercado.

Tratando-se de "software" com esta finalidade no Brasil, MOL3D configura-se, até que se cheguem informações contrárias, como pioneiro.

1.2. Visão Geral

O MOL3D, apesar de ser especialmente dedicado às proteínas, pode ser usado no estudo da geometria de qualquer tipo de molécula. Entende-se por proteínas as macromoléculas resultantes da condensação de moléculas de aminoácidos, através de ligações peptídicas. Os aminoácidos são unidades estruturais básicas das proteínas [1].

Para representar a estrutura tridimensional das moléculas, profissionais das áreas de química, bioquímica e suas áreas correlacionadas recorrem a modelos. Entre os modelos utilizados estão o de "esfera-e-bastão" e o "esquelético" [2].

No modelo "esfera-e-bastão", como o próprio nome sugere, átomos ou aminoácidos (resíduos) são representados por esferas e suas ligações são indicadas por bastões (fig. 1.2.1). Neste modelo o comprimento de ligação de duas esferas corresponde à distância entre os seus centros.



Figura 1.2.1 : Estrutura molecular representada pelo modelo esfera-e-bastão.

O modelo "esquelético" pode ser visto, do ponto de vista geométrico, como uma forma de representação simplificada do modelo "esfera-e-bastão" (fig. 1.2.2). Nele a geometria da molécula é obtida pela composição de segmentos de reta. Uma vez montada a estrutura, observar-se-ão figuras como hexágonos, pentágonos, segmentos de reta, distribuídos no espaço. Os átomos ou aminoácidos (resíduos) ficam distribuídos nos vértices das figuras formadas. Estas figuras estão unidas, umas às outras, numa disposição espacial bem definida.

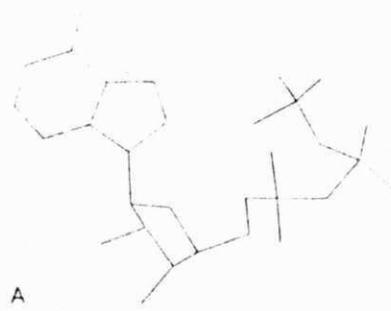


Figura 1.2.2 : Estrutura molecular representada pelo modelo esquelético.

A estrutura espacial das moléculas é conhecida através da análise cristalográfica em RAIO-X, onde se obtém as coordenadas espaciais (x,y,z).

Do ponto de vista computacional, MOL3D é um sistema de telas hierarquizadas de acordo com a sequência do funcionamento do programa. Neste sistema, encontram-se as telas com área de menus, área de mensagens e área de desenhos, como será visto adiante, no item INTERFACE COM O USUARIO.

MOL3D conta com duas categorias de bibliotecas e um conjunto de operações que atuam sobre elas. Nestas categorias encontram-se a biblioteca de aminoácidos e a de macromoléculas. Na primeira, têm-se todos os aminoácidos que figuram entre as proteínas, e na segunda, têm-se moléculas em geral. Com o conjunto de operações mencionado, permitem-se utilizar os

recursos necessários à visualização e as possíveis manipulações das figuras selecionadas. A visualização destas estruturas pode ser feita usando os modelos já citados (esfera-e-bastão e esquelético).

1.2.1. Ambiente de Trabalho

MOL3D foi implementado no sistema Turbo Pascal, num microcomputador da linha IBM-PC e placa de interface gráfica CGA (Color Graphics Adapter). A placa CGA possui uma resolução gráfica de 320 x 200 (320 pontos por linha, 200 linhas) e 4 cores simultâneas (incluindo a cor de fundo). Foram utilizados dois tipos de monitores : RGB e monocromático. Visou-se com isso, avaliar os resultados obtidos em duas situações diferentes, pois o uso do monitor monocromático é mais frequente em ambientes universitários. Embora MOL3D seja desenvolvido no padrão CGA, pode-se utilizá-lo no padrão EGA (Enhanced Graphics Adapter).

Como dispositivo físico de entrada de informações, utiliza-se o teclado. Uma impressora é utilizada para emissão de cópias de tela e outros dados, em papel.

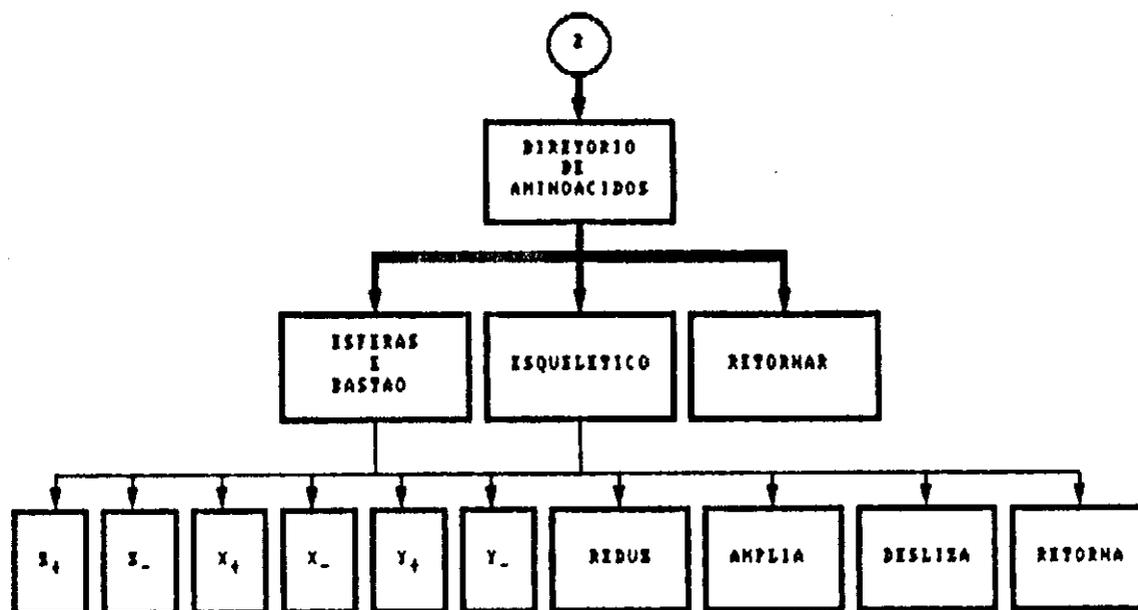
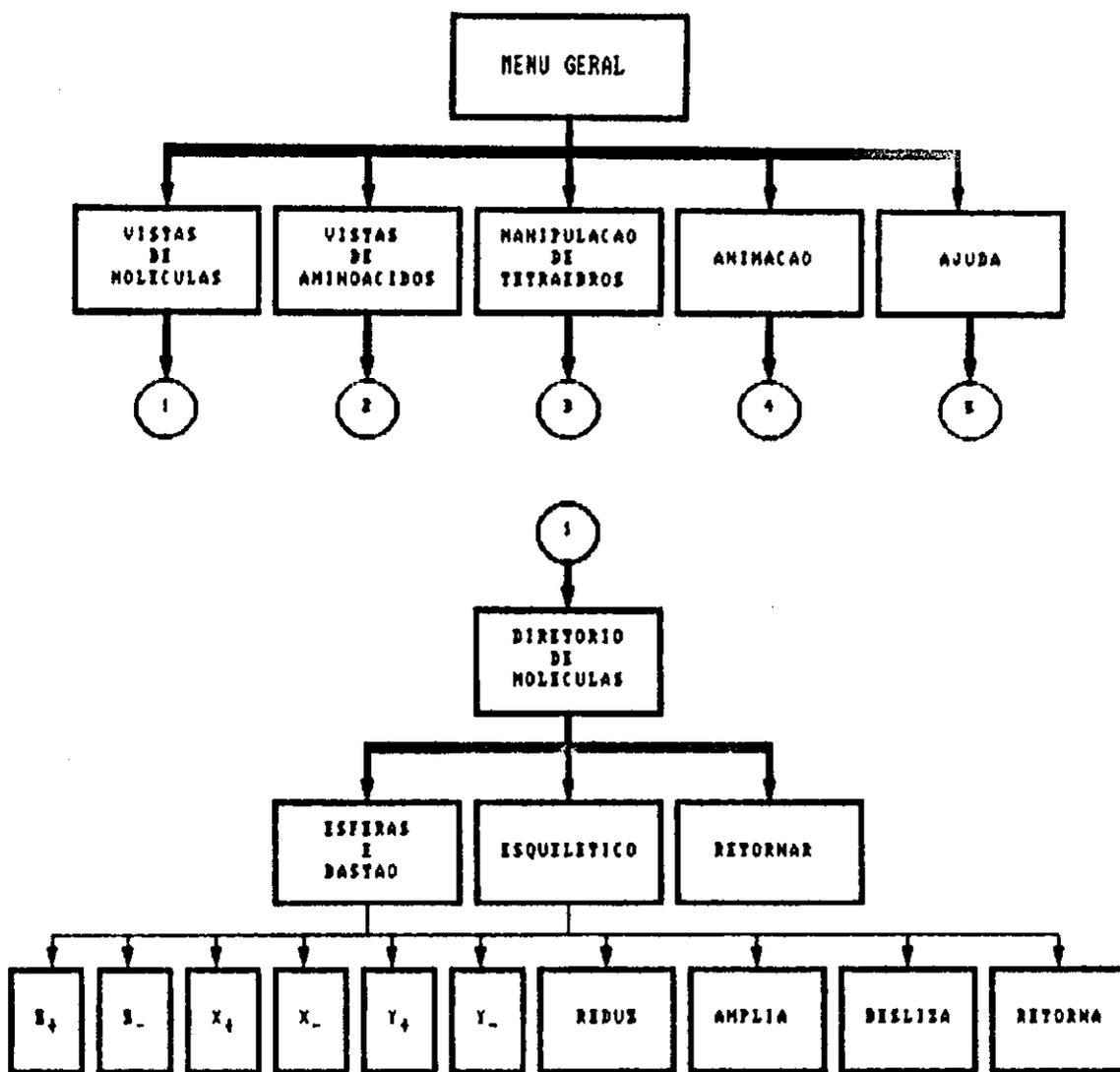
Existem duas razões pelas quais se utiliza o ambiente de trabalho nas condições precárias, supracitadas. A primeira e mais direta é a de serem as condições citadas, as disponíveis no local onde se desenvolveu o sistema. A segunda está vinculada ao

fato de que a maioria dos microcomputadores nacionais de 16 bits utiliza esta configuração como padrão. Esta configuração é a mais frequentemente disponível em instituições de ensino no Brasil. O modelo concebido, entretanto, permite, mediante pequenas alterações, adaptações a um ambiente mais robusto.

1.3. Descrição Geral de Funcionamento

Em linhas gerais, o funcionamento de MOL3D pode ser descrito, conforme mostra o diagrama hierárquico da figura 1.3 : inicialmente é exibido um "Menu Geral", com cinco opções disponíveis ao usuário. Essas opções são :

1. Vistas de molécula ;
2. Vistas de aminoácido ;
3. Manipulação de tetraedros ;
4. Animação ;
5. Ajuda.



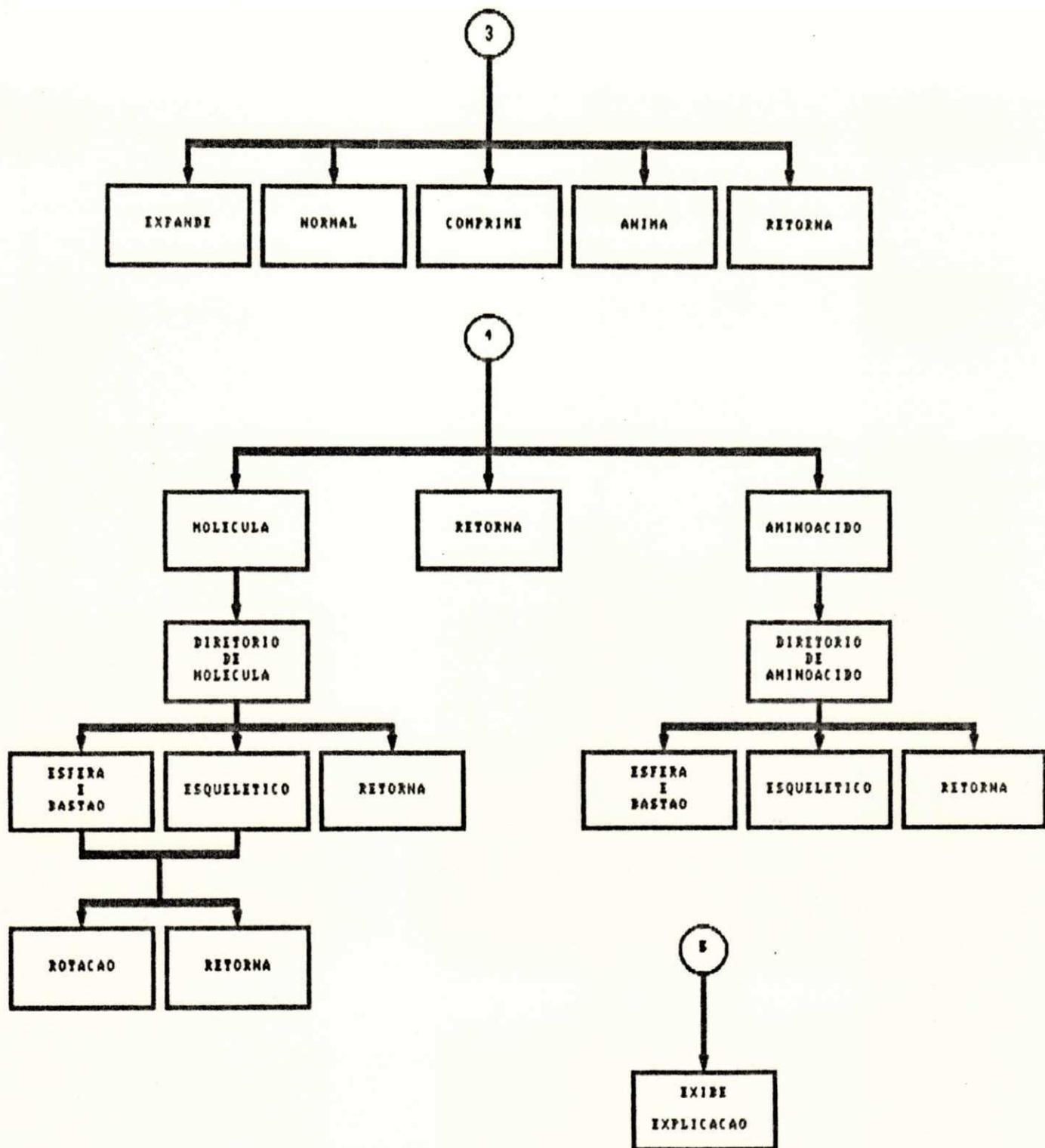


Figura 1.3 : Diagrama hierárquico do funcionamento de MOL3D.

Escolhendo-se a opção 1 do "Menu Geral", será exibido um outro menu, com a biblioteca de moléculas, onde o usuário selecionará uma molécula para trabalho. Escolhida uma molécula aparecerá outro menu, exibindo os modelos geométricos que o sistema põe à disposição do usuário para representar as moléculas. Uma vez escolhido o modelo geométrico desejado, aparecerá o último menu desta sequência, com a apresentação de seis diferentes pontos de visualização da molécula (Z+, Z-, X+, X-, Y+, Y-) e as opções de ampliação ("Zoom+") e redução ("Zoom-") e deslizamento da molécula.

Escolhendo-se a opção 2 do "Menu Geral", aparecerá um outro menu, correspondente à biblioteca de aminoácidos, onde o usuário selecionará um aminoácido. Escolhido o aminoácido, apresentar-se-á um outro menu, exibindo os modelos geométricos para representar os aminoácidos. Uma vez escolhido o modelo geométrico desejado, aparecerá o último menu desta sequência, com a apresentação de seis pontos diferentes de visualização da aminoácido (Z+, Z-, X+, X-, Y+, Y-) e as opções de ampliação ("Zoom+"), redução ("Zoom-") e deslizamento do aminoácido.

Escolhendo-se a opção 3 do "Menu Geral", aparecerá um outro menu, com as operações que podem ser aplicadas no tetraedro, que são compressão, equilíbrio e expansão, além da animação que pode ser feita no mesmo.

Escolhendo-se a opção 4 do "Menu Geral", será exibido um outro menu, com as opções de pergunta sobre o que a animação será feita, sobre moléculas ou sobre aminoácidos. Daí, aparecerá o diretório correspondente (de moléculas ou de aminoácidos), onde se fará a escolha desejada. Em seguida, vem o menu com os modelos, para depois aparecer a animação propriamente dita da molécula ou aminoácido, conforme escolha anterior. Esta animação é obtida das vistas provenientes de rotações, de 10 em 10 graus, da figura em trabalho.

A opção 5, "AJUDA", fornece explicação detalhada acerca das opções anteriores (de 1 a 4). Pressionando-se uma tecla, volta-se ao nível anterior.

Qualquer módulo em que o usuário esteja trabalhando permite alternativa de retorno ao módulo de nível imediatamente anterior.

1.4. Conteúdo do Trabalho

No capítulo 2, serão discutidas algumas técnicas para representação de superfícies moleculares no computador.

No capítulo 3, serão apresentadas as formas de organização externa e interna dos arquivos adotadas por MOL3D. Além disso, serão enunciadas as características envolvidas no projeto da interface de usuário.

Apresentam-se no capítulo 4, os tipos de arquivos utilizados, abordando-se, sobretudo, as operações envolvidas no tratamento deles. Apresentam-se, ainda, os diversos procedimentos utilizados, com vistas à viabilização dessas operações. Nessa ocasião, demonstram-se os cálculos matemáticos presentes na operação de mapeamento de coordenadas de uma dada janela para outra.

No capítulo 5, surgem as denominadas operações de visualização. Nelas estão inseridos os recursos para apresentação e visualização de uma dada figura na tela. Incluem-se aí os pontos de vista do observador, a forma de se projetar a figura na tela e, assim por diante, até o apagamento de esferas e conexões não visíveis ao observador. Além disso, são incluídas as operações que visam fazer alterações básicas de figuras na tela, seguidas de demonstrações dos cálculos matemáticos presentes na operação de mapeamento de coordenadas de uma dada janela para outra. Encontram-se aí as operações de deslizamento e de ampliação e redução ("Zoom") de figuras na tela. Outros tópicos também abordados nesse capítulo, são: rolamento de tela, traçado das conexões interatômicas, atribuição de cores e animação.

No capítulo 6, apresenta-se uma conclusão do trabalho, inclusive uma avaliação do mesmo e sugestões para futuros aperfeiçoamentos do MOL3D.

TECNICAS DE VISUALIZAÇÃO DE SUPERFÍCIES MOLECULARES

Encontra-se em [3] uma grande quantidade de algoritmos para representação de superfícies moleculares no computador. Esses algoritmos representam estas superfícies, ou como uniões de esferas ou como outras superfícies suavizadas. Nesses discutem-se, fundamentalmente, problemas vinculados à eliminação de partes do desenho(molécula) invisíveis ao observador, além disso, apresentam-se algumas técnicas de sombreamento, visando a simulação de efeitos de tridimensionalidade no desenho.

2.1 Uniões de Esferas

A idéia de visualização de superfícies moleculares como uniões de esferas provém da inspiração no "plastic CPK space-filling models", que foi inventado por Corey e Pauling, e posteriormente melhorado por Koltun.

Nestes modelos, cada átomo é representado por uma esfera com raio igual ao raio atômico de Van der Waals, e uma molécula é representada pela união de suas esferas interceptadas.

A seguir, são descritas as idéias envolvidas em alguns algoritmos apresentados em [3].

2.1.1. Algoritmo de Smith e Gund

Smith e Gund desenvolveram um algoritmo que se baseia no modelo "space-filling", onde as linhas de contorno das esferas são desenhadas. As linhas de contorno que são exibidas no desenho da molécula correspondem aos arcos provenientes da interseção de esferas. Para solucionar o problema das linhas não visíveis ao observador, utiliza-se a idéia de aproximar cada arco por um polígono, e para o cálculo de linha escondida, compara-se cada aresta do polígono com cada esfera para eliminar os arcos escondidos. O resultado da implementação desse algoritmo pode ser observado na figura 2.1.1.

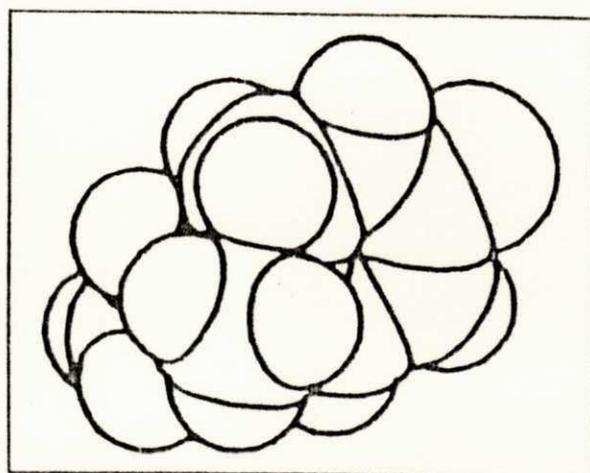


Figura 2.1.1 : Molécula obtida do algoritmo de Smith e Gund.

2.1.2. Algoritmo de Gary Quigley

Em seu programa denominado PLT1, Gary Quigley utiliza círculos concêntricos, com espaçamento diferenciado, tentando com

isso produzir sombras nas esferas, como mostra a figura 2.1.2. Para isso ele utiliza um traçador gráfico ("plotter") com canetas multicoloridas.

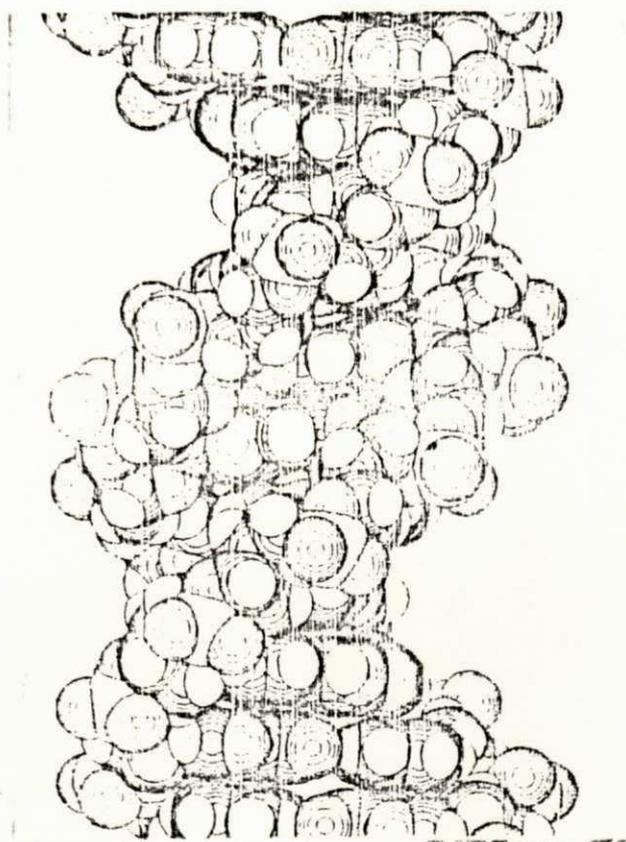


Figura 2.1.2 : Molécula desenhada pelo PLT1.

2.1.3. Algoritmo de MotherWell

Em seu programa denominado PLUTO, MotherWell utiliza, além das linhas de contorno utilizadas por Smith e Gund, hachuras

as partes de cada esfera, cujas faces se ocultam da suposta fonte luminosa. A figura 2.1.3, mostra um desenho obtido de PLUTO.

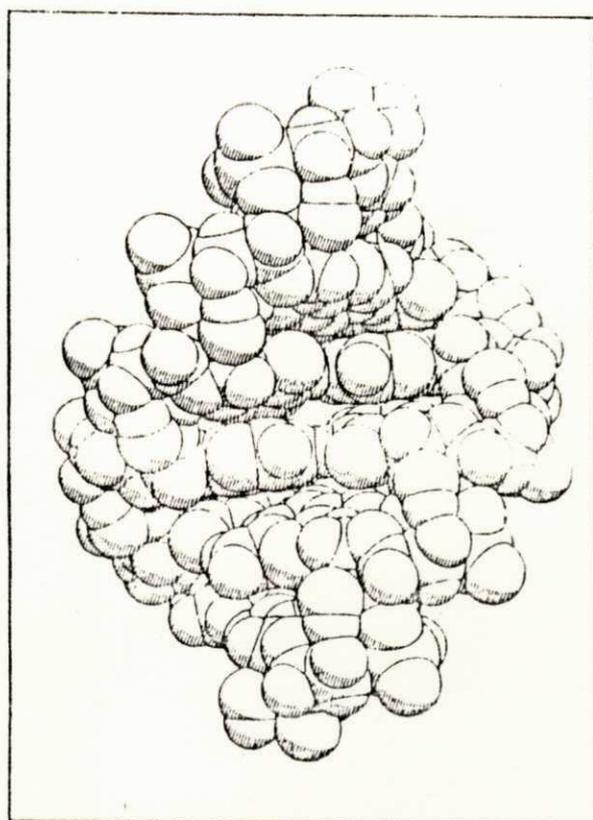


Figura 2.1.1 : Molécula desenhada pelo programa PLUTO.

2.1.4. Algoritmo de Keith Daves

Keith Daves desenvolveu um pacote denominado CHEMGRAPH, o qual utiliza sobreposições de três círculos preenchidos com intensidades diferentes para representar cada esfera.

Objetivando uma representação mais exata das superfícies obtidas de uniões de esfera, este pacote utiliza um algoritmo para remoção de linhas escondidas, gerando contornos poligonais para as partes visíveis de cada esfera. A figura 2.1.4, apresenta um desenho obtido por este algoritmo.

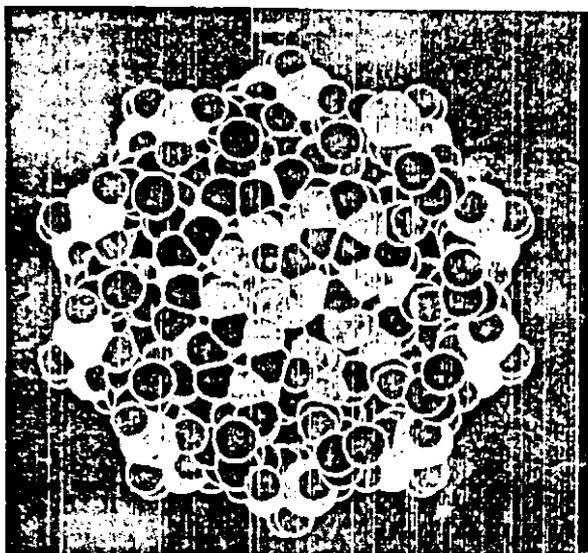


Figura 2.1.4. Uma vista do topo de DNA, obtida pelo CHENGRAPH de Keith Davis.

2.1.5. Algoritmo de Pete Harris

Este algoritmo utiliza a idéia de representar esfera como círculos concêntricos, originados de vários cortes planares aplicados à mesma.

Para isso, supõe-se que r é o raio em pixels(*) da esfera projetada, com isso desenham-se círculos concêntricos com raios de $r, r-1, r-2, \dots, 1, 0$, adotando-se a política de o exterior ser mais escuro e interior ser mais brilhante. As esferas aparecem como se fossem iluminadas por uma fonte de luz atrás do observador. Adicionalmente, se Z é a profundidade em pixels da esfera, esses círculos podem ser vistos como círculos situados em planos com profundidades $Z, Z-1, Z-2, \dots, Z-r$, respectivamente, formando um cone no espaço. A figura 2.1.5 exibe um desenho gerado por este algoritmo.

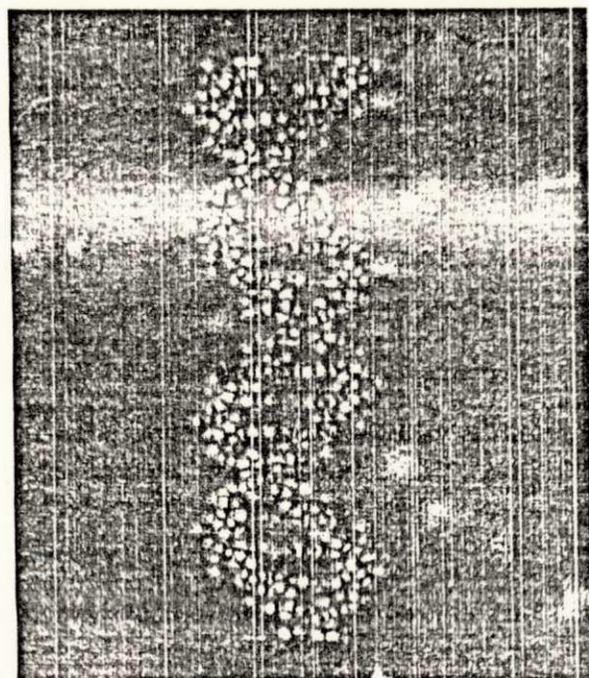


Figura 2.1.5. Um desenho de DNA gerado pelo Algoritmo de Pete Harris.

(*) Pixels (pontos) - deriva de picture element.

Evidenciam-se nos algoritmos baseados no "plastic CPK space-filling models", apresentados acima, o primordial objetivo de melhor caracterização da superfície externa da molécula.

Em MOL3D, entretanto, não há uma preocupação direta com a geometria da superfície externa da molécula, mas com a topologia global da mesma.

As técnicas envolvidas nos algoritmos citados foram, no entanto, valiosas na elaboração dos algoritmos de MOL3D, especificamente no módulo relacionado com a pintura de esferas.

CONCEITOS FUNDAMENTAIS

Este capítulo apresenta características do sistema MOL3D, no que se refere à forma de organização externa e interna dos seus dados e os aspectos da interface de usuário.

3.1. Arquivos Externos

Referem-se aos arquivos em memória secundária. Consideram-se em MOL3D duas classes de arquivos externos. Na primeira classe incluem-se os arquivos com informações da geometria e topologia de objetos (moléculas ou aminoácidos). Na segunda classe, tem-se os arquivos de telas.

Os arquivos pertencentes à primeira classe formam as bibliotecas básicas de aminoácidos e de moléculas. Eles são, por convenção, denotados por um nome seguido de sua extensão, que deve ser, respectivamente, do tipo AMI e MOL. Estes arquivos contêm uma descrição geométrica e topológica de suas figuras, de acordo com o seguinte padrão :

X1	Y1	Z1	R1	(C12	C13	...	C1k)
X2	Y2	Z2	R2	(C21	C23	...	C2k)
.
.
Xk	Yk	Zk	Rk	(Ck1	Ck2	...	Ck-1)

onde:

C_{ki} , significa que o elemento k tem o elemento i como uma de suas conexões, na condição de k, i pertencerem ao conjunto dos naturais, com $(k \neq i)$. (X_k, Y_k, Z_k) e R_k são, respectivamente, as coordenadas espaciais e o raio do elemento k .

Estes arquivos (AMI e MOL) são criados com o auxílio do editor do sistema Turbo Pascal [4]. Os passos necessários à criação destes arquivos, usando o editor Turbo Pascal, serão apresentados no **Apêndice A**. O programa possui rotinas que conseguem ter acesso aos mesmos e trazê-los para o seu "ambiente interativo".

Os arquivos da segunda classe são aqueles em que seus conteúdos são telas preparadas. Cada arquivo corresponde a um conjunto de telas (registros) cuja extensão fica convencionalizada por .TEL. Cada tela pode ser vista como uma fotografia obtida de uma dada figura. Uma vez armazenada uma dada tela, pode-se visualizá-la posteriormente com facilidades, entretanto não se pode manipulá-la. Estes arquivos são criados dentro do próprio programa. Para isso, o usuário informa algum nome com terminação Tel e começa a gravar as telas que deverão compor este arquivo. Há, no entanto, uma proteção contra a inclusão em arquivos já existentes.

3.2. Estrutura de Dados

Referem-se à organização interna dos dados, ou seja, organização a nível de memória principal. A estruturação dos dados é dinâmica [5], [6], na forma de uma "multilista linear simplesmente encadeada", tal que a cada elemento de uma primeira lista corresponde uma sub-lista linear encadeada conforme mostrado na figura 3.2. A primeira lista contém informações acerca dos átomos ou aminoácidos (resíduos), na forma de agregados heterogêneos (registro). Cada registro contém campos envolvendo coordenadas espaciais (x,y,z), o raio, a cor, número indicativo da posição do elemento na molécula, e os ponteiros (um que aponta para o próximo elemento e o outro é o "cabeça" de uma sub-lista). Já na segunda lista ("lista das conexões"), que indica quais as ligações de um dado elemento, as informações também estão em registros, cada um deles com dois campos, um par indicar o número do elemento de conexão em relação à posição na molécula o outro é um ponteiro para o próximo elemento.

Detalhando e declarando em pseudo-código essas listas têm-se :

```
tipo    PontElemento    =  ^Elemento
        PontLig         =  ^Conexão
        Elemento        =  Registro
                           início
                           x,y,z,
                           raio,
                           cor,
```

```

PosiElemento,
ProxElemento,
PrimConexão
fim

```

```

Conexão = Registro
         inicio
         Num-Elemento : inteiro
         ProxConexão : Pontlig
         fim

```

Obs.: O acento circunflexo que aparece no código acima equivale a "aponta para".

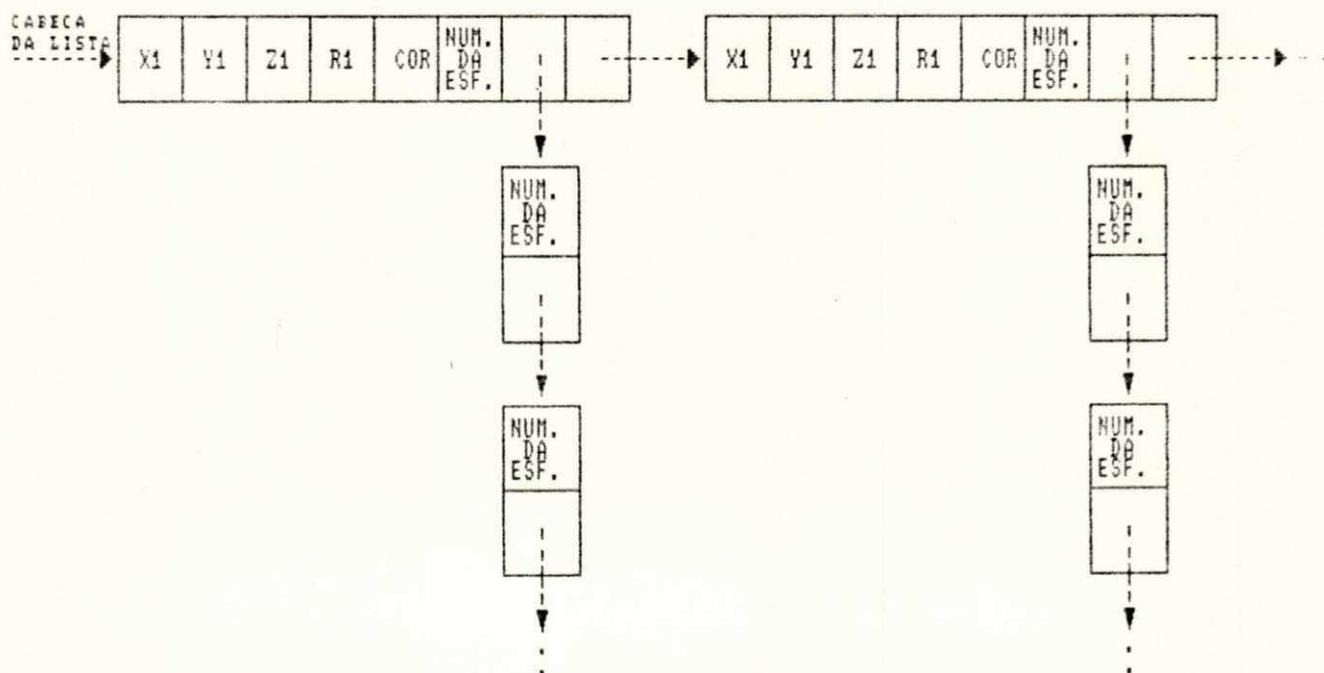


Figura 3.2 : Estrutura de dados dinâmica para representação de estrutura molecular.

3.3. Interface com o Usuário

Procurou-se durante o projeto da interface do MOL3D, que é um sistema interativo, proporcionar ao usuário um ambiente confortável, procurando-se estabelecer uma forma adequada de diálogo. O tipo de diálogo utilizado é feito via "menu", onde as informações estão, na maioria das vezes, distribuídas de forma linear, com as opções em forma de múltipla escolha. Os menus utilizados estão organizados de acordo com uma estrutura de árvore, tendo o usuário a oportunidade de percorrê-la em vários níveis. Além disso se oferece ao mesmo, a opção de sair do programa para o sistema operacional.

As telas foram projetadas de modo a tentarem, sempre que possível, responder ao usuário três perguntas fundamentais : "Onde estou?", "O que posso fazer agora?", ou "De onde venho?"

O formato básico das telas que compõem MOL3D será mostrado a seguir. Serão apresentados os três formatos existentes, que serão denotadas, na ordem de exibição, por tela 1, tela 2 e tela 3.

MENU GERAL

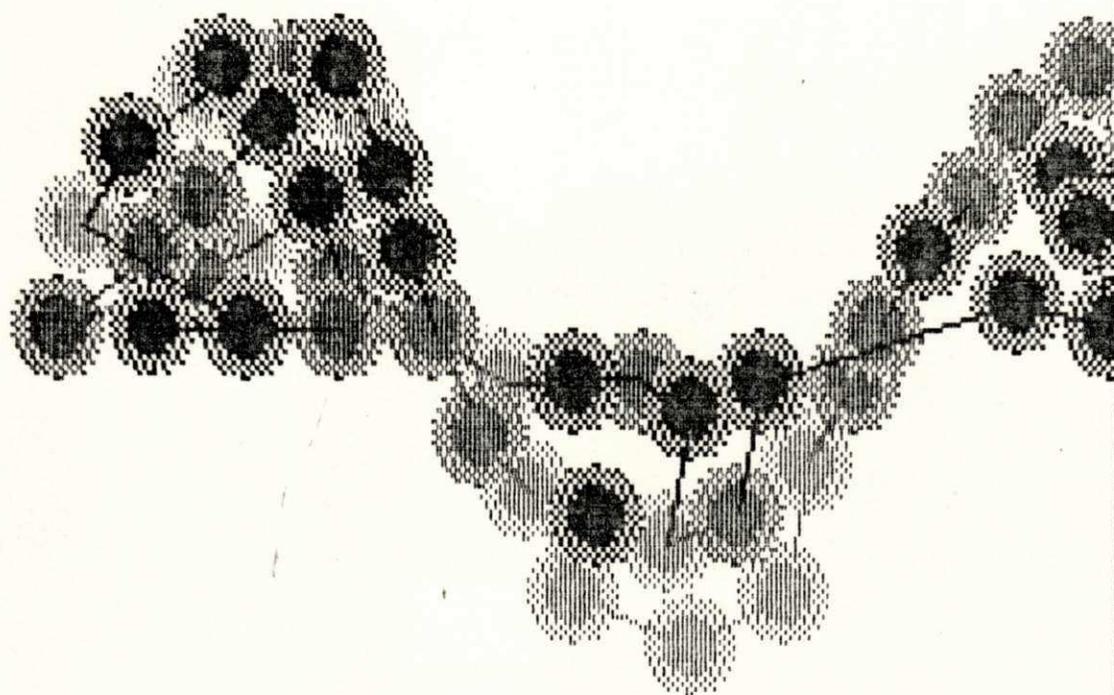
1. Vistas de Molecula
2. Vistas de Aminoacido
3. Manipulacao de Tetraedros
4. Animacao
5. Ajuda
6. Abandona a execucao

Escolha a sua opcao : [_]

Figura 3.3a : Formato da tela 1.

P_DNA.MOL

VISTAS



1. Z+
2. Z-
3. X+
4. X-
5. Y+
6. Y-
7. Red
8. Amp
9. Fin

Escolha a sua opcao ou use ↑↓←→ []

Figura 3.3c : Formato da tela 3.

BASE1.MOL
P_DNA.MOL

BASE10.MOL

BASE2.MOL

BASE4.MOL

BASE6.MOL

Diretorio Atual -> D:\EVANDRO

Nome da Base de Dados (digite FIM para retornar) --->

Figura 3.3b : Formato da tela 2.

3.3.1. Descrição das Telas

A tela 1, como mostra a figura 3.3a, está dividida em três áreas : área I, área II e área III. A área I apresenta uma mensagem que se propõe a informar o usuário, sua localização atual. Na área II (área de menu), surgem informações sobre as diferentes operações que podem ser executadas. A área III tem duas finalidades : a primeira é a de apresentação de mensagem indicando decisão a ser tomada no atual instante; a segunda está reservada a exibição de mensagem de erro, no caso de haver necessidade.

A tela 2, como mostrada na figura 3.3 b, apresenta um formato bem diferente das telas 3.3 a e 3.3 c. Isto é justificado por sua finalidade, que é a de apresentar na tela de vídeo, a biblioteca de molécula ou de aminoácido, a depender da escolha do usuário. Além disso, a distribuição das opções passa,

agora, a ser colunar, devido ao grande número de opções e à possibilidade de acréscimo de novas opções à medida que se criam novos arquivos que pertençam a uma ou a outra biblioteca. Apesar de ser uma tela um pouco pobre em relação às telas 1 e 3, ela apresenta, em essência, as mesmas características de conteúdo de informação da tela 1.

A tela 3, como mostrada na figura 3.3 c, está dividida em quatro áreas : área I, área II, área III e área IV. A área I serve para identificação do desenho, indicando, assim, o nome do mesmo. A área II corresponde ao espaço de desenho da figura. A área III propõe-se a dar uma idéia ao usuário em que módulo ele se encontra e, ao mesmo tempo mostra um menu com as opções de ação disponíveis. A área IV, possui as mesmas finalidades, da área III apresentada na tela 1.

obs.: No momento em que o sistema emite uma mensagem de erro, nas telas 1 e 3, a área da tela de vídeo onde se exhibe esta mensagem fica no modo reverso. Isso é colocado com o objetivo de realce da mesma.

CAPITULO IV

OPERAÇÕES SOBRE ARQUIVOS

Neste capítulo são tratadas as operações que tornam possíveis a criação e a manipulação de arquivos. Esses arquivos são os de objetos (tipos AMI e MOL) e os de telas (tipo TEL).

4.1. Operações Sobre Arquivos de Objetos

Os arquivos de objetos são criados fora do ambiente do programa. Para criá-los, utiliza-se o Editor do Sistema Turbo Pascal. Para que o arquivo a ser criado faça sentido para o programa, deve obedecer à convenção adotada, que é a de possuir uma terminação AMI para aminoácidos ou MOL para moléculas. Uma vez criado, este arquivo passará a fazer parte de sua biblioteca correspondente.

Para manipulação destes arquivos, MOL3D oferece três procedimentos básicos: o procedimento "diretório" (de moléculas ou de aminoácidos), o "Escolha_Arquivo" e o "Leia_Arquivo". O primeiro objetiva trazer para tela (no momento da execução do programa) todos os arquivos que possuem uma determinada terminação [7], seja ela AMI ou MOL. O segundo seleciona o arquivo desejado para trabalho e fixa a correspondência entre o nome externo e o interno deste arquivo. O terceiro faz a leitura do conteúdo do arquivo selecionado, preenchendo assim a estrutura de dados no programa.

Os procedimentos em linguagem algorítmica, são, na ordem que foram mencionados, os seguintes :

Procedimento Diretório_de_Moléculas

```
início
  Obtenha o nome do primeiro arquivo de molécula
  Enquanto houver arquivo de molécula faça
    início
      Escreva o nome do arquivo obtido
      Obtenha o próximo arquivo de molécula
    fim
  fim
fim.
```

obs.: Caso esteja manipulando aminoácidos, substituem-se todas as ocorrências de moléculas no procedimento acima, por aminoácidos.

Procedimento Escolha_Arquivo

```
início
  Obtenha o nome do arquivo desejado
  Enquanto não existe tal arquivo faça
    início
      emitir mensagem de erro
      informe o nome do arquivo novamente
    fim
  Senão
    selecione o arquivo para operação
fim.
```

Procedimento Leia_Arquivo

```
início
  Enquanto não for fim do arquivo faça
    início
      Leia (x,y,z)
      Leia (raio)
      Atribua o nº de posição da esfera no arquivo
      ( Percorrer a lista de conexões desta esfera )
      Enquanto não terminar a lista de conexões
        início
          Leia (Num_Conexão)
        fim
      fim
    fim
fim.
```

4.2. Operações Sobre Arquivo de Telas

Esta classe de operações permite ao usuário o armazenamento de telas em um dado arquivo e leitura dos mesmos. Um arquivo de telas pode conter um ou mais registros. Cada registro corresponde a uma tela de média resolução (320 x 200 pixels) da interface gráfica CGA de micros da linha IBM-PC, tendo, portanto, um comprimento de 16Kbytes. Através destes arquivos o usuário pode, sempre que desejar, rever o conteúdo de telas previamente armazenadas.

4.2.1. Armazenamento de Telas

Através do procedimento Grava-Tela, o usuário tem a opção de guardar uma dada tela num arquivo. Para isso, ativa-se o procedimento, daí este solicita o nome do arquivo e, dependendo da situação, inicia-se o armazenamento das telas naquele arquivo (conforme o Procedimento Grava-Tela-No-Arquivo, a seguir).

Procedimento Grava-Tela-No-Arquivo

início

obtenha o nome do arquivo
se já existe tal arquivo
então confirme gravação
senão

início

abra arquivo com nome informado
enquanto desejar : execute Grava-Tela obtida
fecha arquivo

fim

fim.

4.2.2. Recuperação de Telas

Através do procedimento Obtenha-Tela-do-Arquivo, o usuário tem a oportunidade de observar no vídeo, as telas armazenadas num dado arquivo.

Procedimento Obtenha-Tela-do-Arquivo

Início

obtenha o nome do arquivo

Procura este arquivo

Se arquivo não existe

então

emita mensagem

senão

início

Abra arquivo com nome informado

Leia arquivo e apresente suas telas

Fecha arquivo

fim

fim.

CAPÍTULO V

OPERAÇÕES DE VISUALIZAÇÃO

Estas operações relacionam-se com as etapas pertinentes ao processo de apresentação e visualização de uma figura na tela. Isso vai desde as formas de posicionamento do observador até o posicionamento da figura em si. Incluem-se, ainda, as operações que visam fazer alterações básicas de figura na tela. Uma dessas operações faz o "deslizamento da figura na tela", a outra faz a ampliação ou redução da figura na tela ("ZOOM").

5.1. Posicionamento do Observador

Esta operação fornece as diferentes posições em que o observador pode visualizar a figura. Dispõem-se de seis posições de visualização distintas. Nessas posições, estão os eixos x, y, z nos semi-eixos positivos e negativos. Os procedimentos que implementam estas operações são os seguintes :

Procedimento Zpositivo

início

percorra Lista de Coordenadas
enquanto não percorrer toda a lista
obtenha (x, y, z) da esfera
retorne (x, y, z)

fim.

Procedimento Znegativo

início

percorra Lista de Coordenadas
enquanto não percorrer toda a lista
obtenha (x, y, z) da esfera
retorne $(-x, y, -z)$

fim.

Procedimento Xpositivo

inicio

percorra Lista de Coordenadas
enquanto nao percorrer toda a lista
obtenha (x,y,z) da esfera
retorne $(-z,y,x)$

fim.

Procedimento Xnegativo

inicio

percorra Lista de Coordenadas
enquanto nao percorrer toda a lista
obtenha (x,y,z) da esfera
retorne $(z,y,-x)$

fim.

Procedimento Ypositivo

inicio

percorra Lista de Coordenadas
enquanto nao percorrer toda a lista
obtenha (x,y,z) da esfera
retorne (z,x,y)

fim.

Procedimento Ynegativo

inicio

percorra Lista de Coordenadas
enquanto nao percorrer toda a lista
obtenha (x,y,z) da esfera
retorne $(-z,x,y)$

fim.

Com retorno do terno de coordenadas ao programa, este utiliza as duas primeiras para o traçado da esfera projetada, e a terceira servirá para indicar a sua profundidade.

5.2. Projeções

Matematicamente, uma projeção equivale a um mapeamento de entidades de uma dimensão de uma dada ordem, para outra de ordem inferior [8]. Neste trabalho, tem-se um caso particular de projeção, que corresponde à operação de mapeamento de uma dada imagem espacial no plano de um dispositivo de visualização (tela gráfica do computador).

5.2.1. Tipos de Projeções

Uma projeção fica definida por dois elementos básicos : o plano de projeção ou quadro (plano sobre o qual a imagem será projetada), e o centro de projeção ou ponto de vista [9]. A projeção de uma imagem tridimensional fica determinada pelo conjunto de pontos do plano (quadro), obtidos da sua interseção com as linhas visuais que partem do objeto e atingem o centro de projeção (fig. 5.2.1 a).

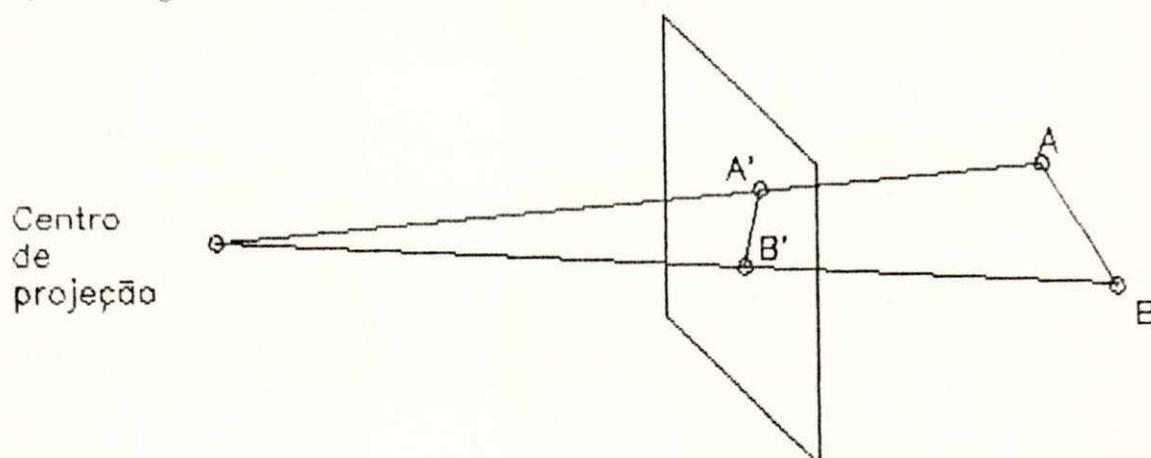


Figura 5.2.1 a : Elementos de projeção na projeção perspectiva da linha AB.

A projeções (***) podem ser classificadas, de acordo com a posição do centro de projeção, em duas classes básicas : perspectiva e paralela.

A projeção perspectiva é aquela em que a distância entre o centro de projeção e o plano de projeção é finita, como é o caso da figura 5.2.1 a. Nesta classe de projeção, a imagem do objeto é representada com maior realismo, apresentando as dimensões das partes do objeto mais afastadas do centro de projeção reduzidas em relação às dimensões das partes mais próximas. Este tipo de projeção cria um efeito semelhante aos dos sistemas fotográficos e ao sistema visual humano.

A projeção paralela é aquela em que a distância entre o centro de projeção e o plano de projeção é infinita, como mostra a figura 5.2.1 b. Numa projeção paralela, as dimensões do objeto não são alteradas em função de sua distância ao centro de projeção, pois os raios projetores, neste caso, são paralelos entre si e, portanto encontram-se no infinito.

(***) O termo projeções aparece neste caso, para denotar as projeções geométricas planares, isto é, aquelas em que a superfície de projeção é um plano e os projetores são linhas retas.

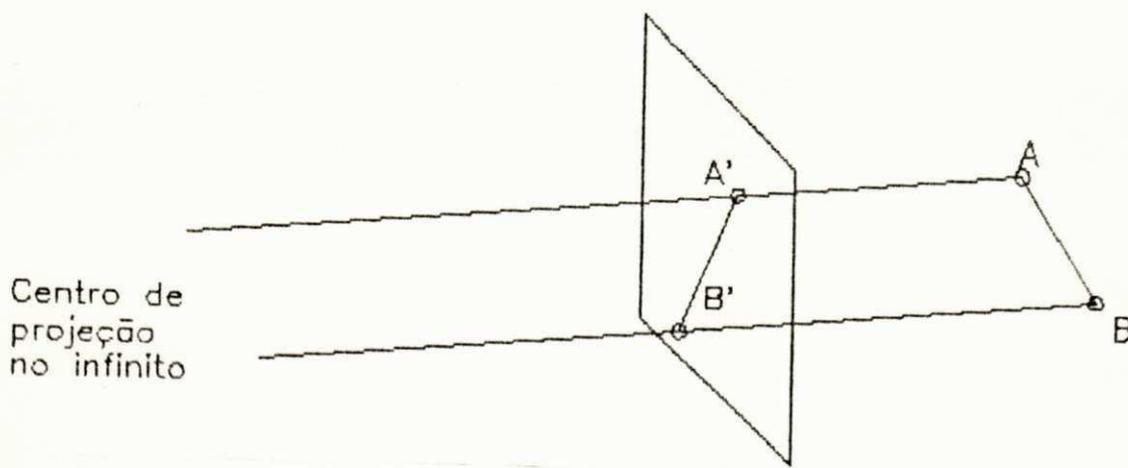


Figura 5.2.1 b : Projeção paralela da linha AB.

5.2.2. Transformações de Projeção

As operações utilizadas para projeção de um objeto tridimensional no plano da tela de visualização do computador são denominadas Transformações de Projeção. Os cálculos envolvidos nas transformações de projeção perspectiva e paralela são apresentados a seguir.

Assume-se, no cálculo da transformação de perspectiva, que o plano de projeção é paralelo ao plano XY (normal ao eixo Z) e está situado a uma distância D do observador (centro de projeção), como mostra a figura 5.2.2 . Utiliza-se um sistema de coordenadas "left-handed", isto é, um sistema com eixo X orientado para direita, eixo Y dirigido para cima e eixo Z orientado para dentro do plano da tela de visualização, como mostra a figura 5.2.2 .

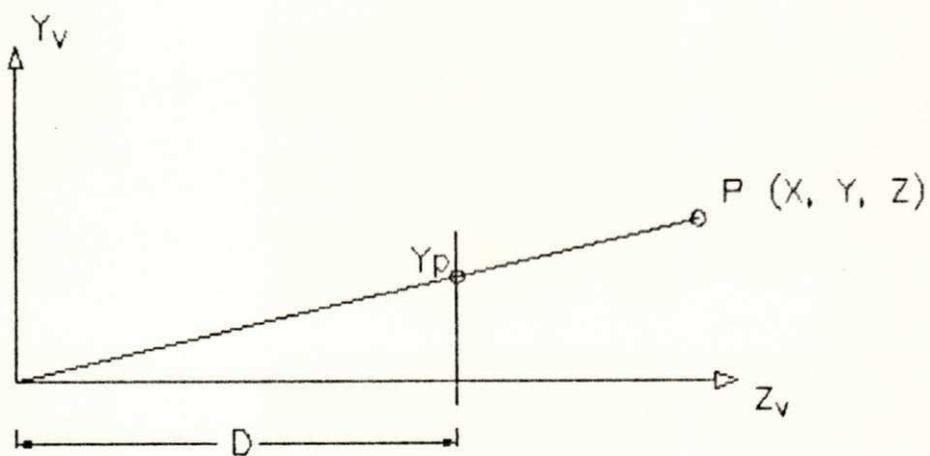
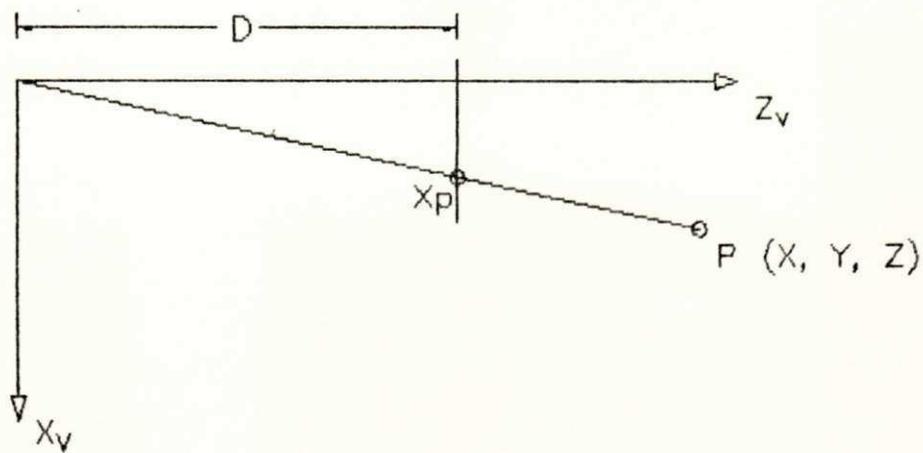
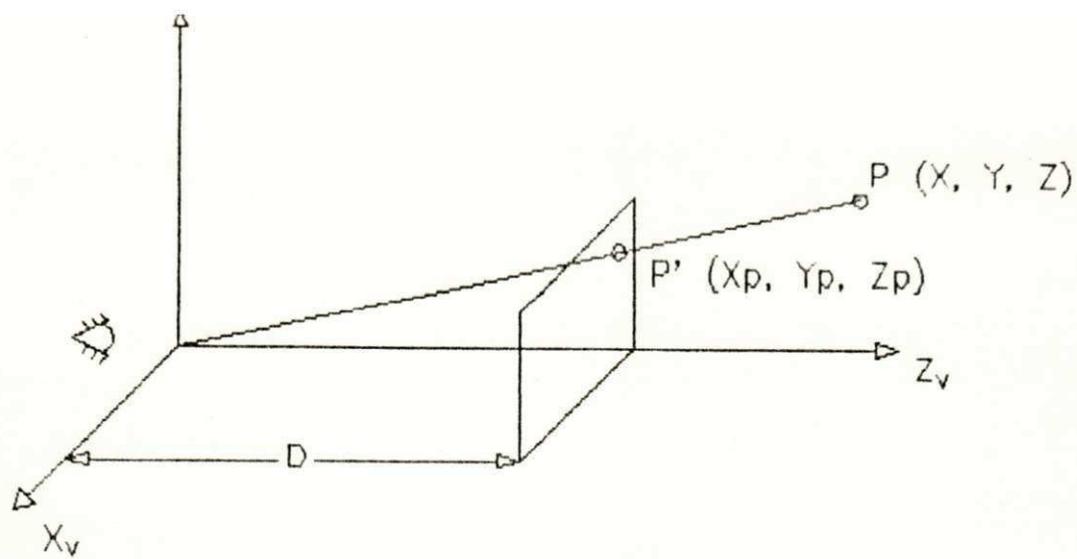


Figura 5.2.2 : Projeção perspectiva com vistas do topo e de lado.

Seja $P = (X, Y, Z)$ o ponto a ser projetado no plano de projeção, e $P_p = (X_p, Y_p, Z_p)$ sua projeção, tal como indicado na figura 5.2.2 . Utilizando-se propriedades de semelhança de triângulos aplicadas nos triângulos das figuras 5.2.2a e 5.2.2b, tem-se :

$$X_p/D = X/Z \quad \leftrightarrow \quad X_p = D * (X/Z) ;$$

$$Y_p/D = Y/Z \quad \leftrightarrow \quad Y_p = D * (Y/Z) .$$

Assume-se, para o cálculo da transformação paralela, que o plano de projeção seja normal ao eixo Z para $Z = 0$. Nesta, o ponto $P = (X_p, Y_p, Z_p)$ a ser projetado, corresponderá a $P_p = (X_p, Y_p, Z_p)$, com $X_p = X$, $Y_p = Y$ e $Z_p = 0$.

De acordo com os resultados obtidos acima, percebe-se na transformação perspectiva a necessidade das coordenadas (X, Y) serem divididas por Z e multiplicadas por D. Por outro lado, verifica-se que a transformação de projeção paralela não requer nenhum cálculo aritmético para obtenção de suas coordenadas de projeção. Em consequência disto, o esforço computacional solicitado pela transformação perspectiva é bem maior que o solicitado pela transformação paralela. Por essa análise, e pelas limitações de velocidade impostas pelo equipamento em uso, trabalha-se em MOL3D com projeção paralela.

5.3. Apagamento de Esferas e Conexões Não Visíveis ao Observador

A tarefa de representação de objetos tridimensionais sem ambiguidades no computador requer a eliminação de partes do objeto não visíveis ao observador. A determinação de porções invisíveis depende, basicamente, da posição do observador.

Existem dois tipos de abordagem para resolver o problema acima mencionado.

O primeiro visa percorrer o conjunto de objetos de uma cena, explorar as relações geométricas que existem entre esses objetos visando a determinação das partes visíveis de cada um, para posterior projeção das mesmas em uma tela. Os algoritmos que atacam o problema deste modo são chamados "Algoritmos de Espaço do Objeto".

O segundo tipo de abordagem objetiva determinar para cada pixel da tela, qual objeto deve oferecer os atributos de intensidade ou cor do referido pixel. Os algoritmos que abordam o problema deste modo são chamados "Algoritmos de Espaço de Imagem".

Todos os algoritmos destinados a resolver o problema do apagamento de linhas ou superfícies escondidas (de espaço do objeto ou de imagem) têm como entrada : a descrição do(s) objeto(s) e posição do observador hipotético. Nestes algoritmos

são utilizados dois princípios : o de ordenação geométrica ("geometric sorting") para distinguir partes visíveis de um objeto de todas aquelas que estão escondidas, e o de coerência que explora a propriedade de semelhança entre elementos geometricamente próximos.

5.3.1. Algoritmos de Espaço do Objeto

Dentro dos algoritmos de espaço do objeto, encontram-se duas classes de algoritmos consagrados na literatura de Computação Gráfica. Na primeira está o Algoritmo de Roberts e na segunda estão os algoritmos de Appel, Loutrel e Montanari [10].

5.3.1.1. Algoritmo de Roberts

O algoritmo de Roberts foi a primeira solução dada ao problema de linhas escondidas ("hidden line"). Este algoritmo consiste na eliminação das arestas ou planos de cada volume que são ocultados por seu próprio volume. Subseqüentemente, cada aresta remanescente de cada volume é comparada com cada um dos volumes remanescentes para determinar que porção ou porções, se houver alguma, são escondidas por esses volumes [11]. O algoritmo de Roberts exige que todos os objetos(volumes) do desenho sejam convexos. Apesar de que objetos côncavos podem ser representados por uma coleção de objetos convexos, e conseqüentemente, tratáveis adequadamente por este algoritmo; computar esta

decomposição eficientemente é difícil.

5.3.1.2. Algoritmos de Appel, Loutrel e Montanari

Nestes algoritmos o teste da visibilidade é feito para cada aresta em relação a todas as outras. A idéia é testar a visibilidade ou invisibilidade total da aresta, fazendo isso através da verificação dos vértices que determinam esta aresta. Assim, se a visibilidade de uma aresta é calculada, ela pode ser usada para evitar cálculos nas outras aresta que têm o mesmo vértice daquela aresta calculada. Desta forma, a maioria das decisões de visibilidade tornam-se incrementais.

5.3.2. Algoritmos de Espaço da Imagem

Estes algoritmos também são chamados de algoritmos de prioridade de profundidade. Eles se dividem bem em duas categorias : aqueles que trabalham com áreas na tela e aqueles que trabalham com pontos na tela [10].

5.3.2.1. Algoritmo de Warnock

Este algoritmo está incluído na primeira categoria (trabalha com área na tela) e tem como finalidade computar a apropriada intensidade para cada área da tela. Neste, parte-se do princípio que a área da tela, chamada de janela, pode ser

declarada como homogênea (uma só tonalidade), e então pode ser exibida com um simples cálculo de tonalidade. Esta hipótese é considerada correta se uma face cobre totalmente a janela e não existe nenhuma outra entre esta e o observador. Se a hipótese for provada como falsa, a janela é dividida em quatro outras janelas iguais, e o processo será repetido.

5.3.2.2. Algoritmos de Romney, Bouknight e Watkins

Estes algoritmos são disponíveis para displays do tipo raster-scan, ou seja, de rastreamento. Eles computam as interseções entre o plano horizontal determinado por cada linha de varredura e cada face dos objetos existentes na cena (supõe-se que cada objeto é um poliedro de faces planas). Posteriormente destas interseções (segmentos de reta) têm suas profundidades (distâncias ao observador) comparadas.

Estes três algoritmos têm em comum uma filosofia básica: primeiramente eles fazem um ordenamento em Y dos vértices de cada objeto, visando minimizar o tempo de cálculo das interseções. Calculadas as interseções se procede a um ordenamento em X das mesmas. Finalmente é feita uma pesquisa na profundidade Z para estabelecer a face visível.

Outras características desses algoritmos é o uso de conceito de coerência de linha, ou seja : arestas que interceptam

uma linha de varredura muito provavelmente interceptarão a uma linha adjacente e, permitindo assim que se mantenha uma lista de arestas ativas.

Em seguida os algoritmos examinam a lista reduzida de arestas para computarem quais faces são visíveis e em que porções da linha de tela. Este processo envolve a divisão da linha de tela em seções menores chamadas "spans". A comparação da profundidade dos segmentos será comentada no **Apêndice B**.

Como podemos deduzir, o problema é reduzido aqui a duas dimensões : X e Z.

As soluções apresentadas nestes algoritmos são de propósito geral. Desejou-se em MOL3D uma solução adequada para um problema específico, onde os objetos a serem apagados são as esferas e suas conexões. Assim, buscou-se uma solução bem particular, visando uma maior otimização na mesma. Para isto, empregaram-se algoritmos mistos, utilizando elementos de algoritmos de espaço do objeto, bem como de espaço da imagem. A seguir serão apresentadas as etapas de soluções envolvidas no projeto destes algoritmos.

Antes de iniciar-se a descrição do apagamento em si, faz-se necessário ressaltar uma etapa precedente. Nesta etapa, que será denominada de pré-processamento, faz-se uma ordenação das esferas em relação à coordenada que mede a proximidade das

mesmas ao observador, convencionando-se neste programa a utilização da coordenada z. No momento de se fazer a conexão, verifica-se qual esfera, no par que compõe a ligação, tem o maior valor de z, pois é nela que começa o traçado da aresta. Para cada aresta (conexão), é testada a visibilidade contra cada esfera potencialmente na frente da mesma. E utilizado, nesse caso, o conhecido teste da "bounding box" [9] : neste teste são verificados quais esferas se encontram dentro da "bounding box" da aresta (ver figura 5.3a). As esferas acima, abaixo, à esquerda e à direita, serão desprezadas; isto é verificado do seguinte modo : Toma-se o maior e o menor valores de x e de y (obtendo-se, respectivamente, $maxx$, $minx$, $maxy$ e $miny$); em seguida, testam-se os valores de x, y, z e r de cada esfera (entre as que não compõem a aresta). O Teste é feito da seguinte forma : se $(y + r < miny)$ ou $(y - r > maxy)$ ou $(x + r < minx)$ ou $(x - r > maxx)$ então a esfera testada está fora da "bounding box"; caso contrário a esfera está dentro dela.

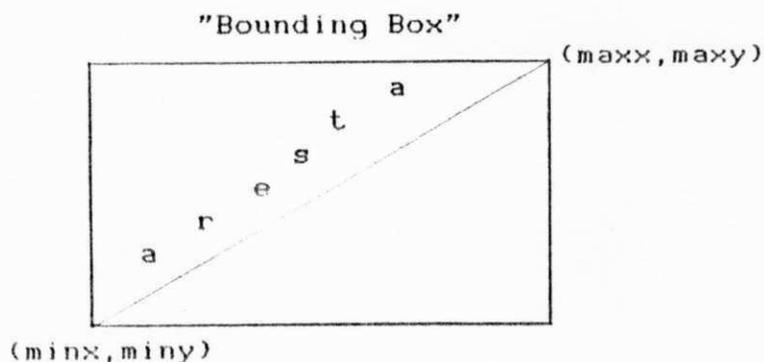


Figura 5.3a : Teste da "Bounding Box".

Para o apagamento de objetos propriamente dito, ainda há um outro teste dentro da "bounding box" a ser feito : é a vez de se observar quando uma dada esfera encobre ou não a aresta; o problema passa a ser o de verificar se a esfera que está sendo testada pertence ou não ao triângulo projetado no plano xz, formado pelos pontos extremos (P_i e P_f) da aresta e um outro ponto (P), como mostrado no esquema da figura 5.3b . Caso a esfera esteja no interior do triângulo, ela não encobrirá a aresta; caso contrário ela a encobrirá. O teste será feito do seguinte modo : toma-se o ponto mais afastado do observador (P_f), entre os pontos extremos da aresta, e une-se-o ao centro da esfera em teste (como indica a figura 5.3b). Agora, a questão restringe-se a verificar se o valor do ângulo alfa é menor do que o valor do ângulo beta (indicados na figura 5.3b), ou equivalentemente, verificar se a tangente do ângulo alfa é menor do que a tangente ângulo beta. Em caso afirmativo, a esfera em teste não encobrirá a aresta; em caso contrário, a encobrirá.

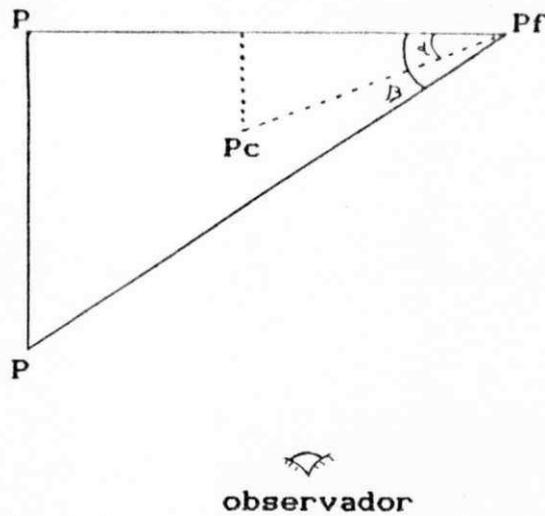


Figura 5.3b : Triângulo de visualização.

Obs.: Na figura 5.3b, o ponto Pc equivale ao centro da esfera em teste. Nesta situação, ela mostra o caso em que a esfera em teste está dentro do triângulo.

No caso de esferas dentro da "bounding box", testa-se cada uma delas, verificando se ela está na frente ou atrás da aresta. No caso de haver esfera na frente da aresta (há uma lista no programa destinada a guardar estas esferas), esta aresta será construída ponto a ponto (como será discutido mais abaixo, no item **traçado das conexões interatômicas**, calculando para cada ponto, a distância do mesmo ao centro de cada esfera na frente. Se para pelo menos uma esfera, a distância do ponto ao seu centro for menor que seu raio, esse ponto não será plotado na tela; caso contrário o mesmo será plotado. Não havendo esferas

obscurecendo a aresta, esta será construída diretamente, através da função "Line" do Turbo Pascal.

5.4. Deslizamento

Para executar uma operação do tipo deslizamento, toma-se cada par de coordenadas de tela (xt, yt) e adiciona-se um deslocamento desejado. Tem-se assim que : $xt = xt + Dxt$ e $yt = yt + Dyt$. Sua utilidade resume-se à movimentação da figura na tela. A operação de deslizamento em MOL3D, assume uma forma particular. Nesta combinam-se a operação de Deslocamento com a operação de Rolamento de Tela (descrita no item 5.9), visando uma possível recuperação de porções do desenho que estão fora da janela de tela.

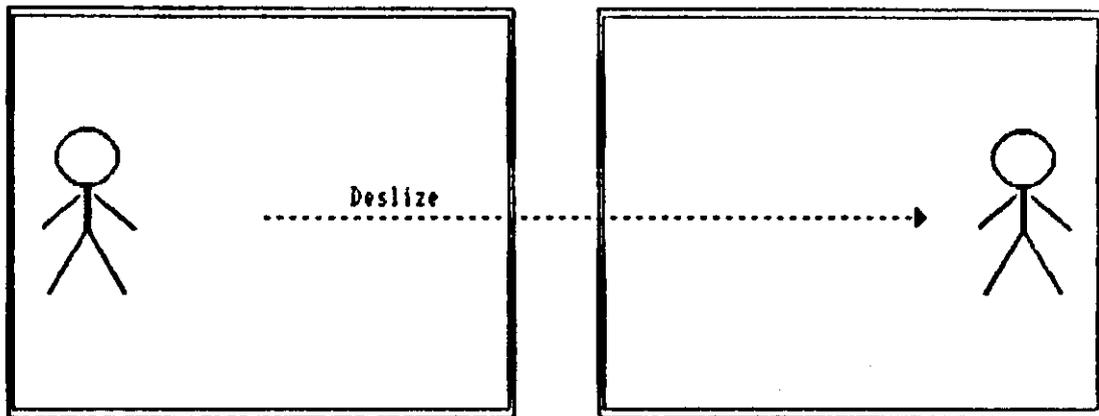


Figura 5.4 : Deslizamento horizontal do desenho

5.5. Ampliação e Redução ("ZOOM")

Em muitas situações é bastante útil visualizar uma dada figura, podendo aproximá-la (para perceber detalhes particulares, "Zoom+") ou afastá-la (para obter uma visão mais global, "Zoom-"). A operação envolvida para efetuar um "ZOOM", resume-se a um mapeamento de janelas.

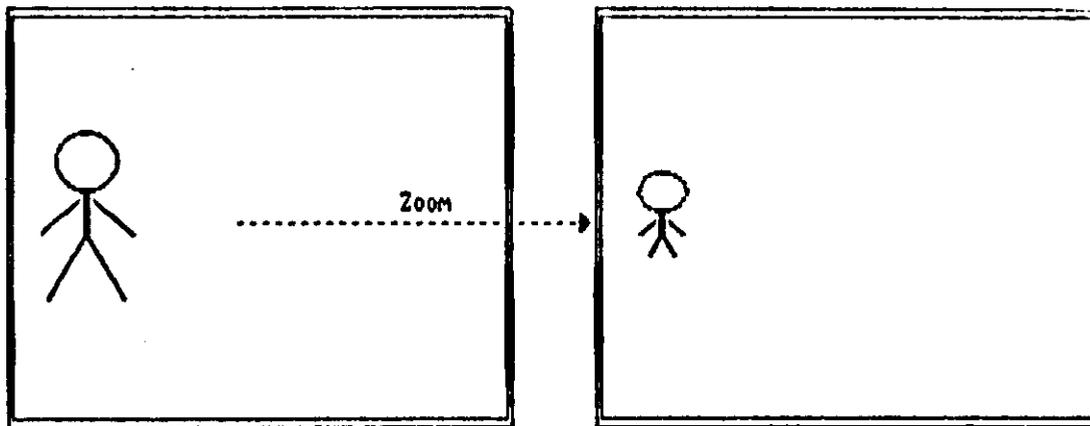


Figura 5.5 : Operação de "zoom" (afastamento do desenho em relação ao observador).

5.5.1. Operações de Mapeamento

Esta operação permite transformar coordenadas de uma janela de visualização W ("Window") para janela de tela V ("Viewport"). Para isso, deve-se manter uma correspondência de

proporcionalidade entre estas coordenadas, visando preservar as proporções de dimensão da figura em questão.

Considerem-se as duas janelas W e V abaixo. Seja (x_w, y_w) um ponto qualquer da "Window" e (x_v, y_v) , o seu correspondente na "ViewPort". Suponha que (x_{wmin}, y_{wmin}) e (x_{wmax}, y_{wmax}) sejam, respectivamente, o canto inferior esquerdo e o canto superior direito da "Window" e (x_{vmin}, y_{vmin}) e (x_{vmax}, y_{vmax}) sejam, respectivamente, o canto inferior esquerdo e o canto superior direito da "ViewPort", como mostra a figura 5.5.1.

Este mapeamento satisfaz as seguintes condições :

a) A distância de x_v até a aresta vertical esquerda da "ViewPort" dividida pelo comprimento da aresta horizontal da "ViewPort" é igual a distância de x_w até a aresta vertical esquerda da "Window" dividida pelo comprimento da aresta horizontal da "Window". Assim, tem-se :

$$(x_v - x_{vmin}) / (x_{vmax} - x_{vmin}) = (x_w - x_{wmin}) / (x_{wmax} - x_{wmin})$$

$$\rightarrow x_v = x_{vmin} + ((x_{vmax} - x_{vmin}) / (x_{wmax} - x_{wmin}) * (x_w - x_{wmin})).$$

b) A distância de y_v até a aresta horizontal inferior da "ViewPort" dividida pelo comprimento da aresta vertical da "ViewPort" é igual a distância de y_w até a aresta horizontal inferior "Window" dividida pelo comprimento da aresta vertical da "Window". Assim, tem-se :

$$(yv - yvmin)/(yvmax - yvmin) = (yw - ywmin)/(ywmax - ywmin)$$

$$\rightarrow yv = yvmin + ((yvmax - yvmin)/(ywmax - ywmin) * (yw - ywmin)).$$

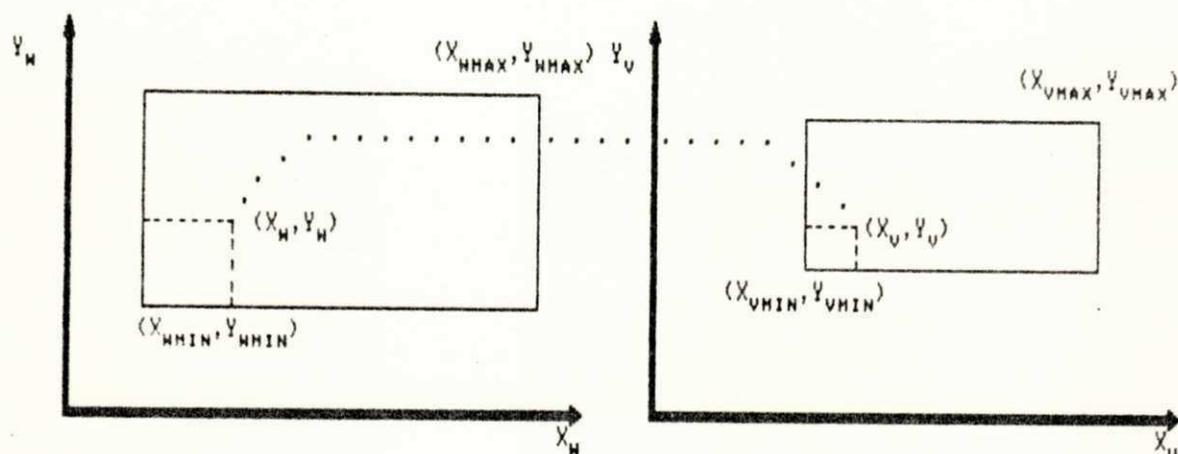


Figura 5.5.1 : Mapeamento de coordenadas.

5.6. Sistemas de Coordenadas

Geralmente um objeto que se deseja representar na tela do vídeo não preserva suas dimensões reais. Isto se deve, a necessidade de se transformar suas dimensões para um tamanho compatível com o tamanho da tela. Para se conseguir esta compatibilidade, o sistema MOL3D utiliza três sistemas de coordenadas: universal, visual e de tela (figura 5.6).

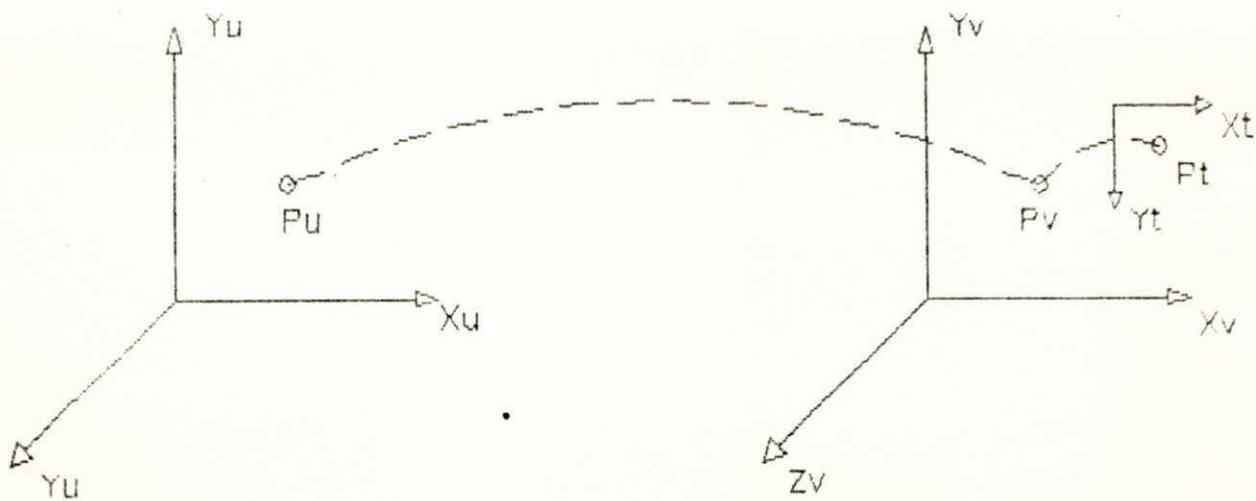


Figura 5.6 : Sistemas de coordenadas.

No sistema de coordenadas universais (x_u, y_u, z_u) o objeto é definido em suas dimensões reais (independente do dispositivo de visualização). Fixando-se a posição do observador em algum ponto do espaço, necessita-se associar a este, um sistema de coordenadas, no qual o referido ponto será a origem. Feito isto, transformam-se as coordenadas do sistema universal para este novo sistema. O sistema em questão é denominado de coordenadas visuais. Estando-se no sistema de coordenadas visuais (x_v, y_v, z_v) onde está posicionado o olho do observador, faz-se a projeção do objeto. Após a transformação de sistemas de coordenadas é então escolhida a área da molécula a ser visualizada. Isto é feito mediante a definição de uma janela retangular de visualização ("window"). Paralelamente é definida uma área na tela do dispositivo de visualização onde será apresentada a parte da molécula escolhida. Uma das operações

envolvidas neste processo é a transformação das coordenadas da porção do objeto enquadrado na janela de visualização ("window") para a janela de tela ("viewport"). Entre o mapeamento de coordenadas da janela de visualização para janela de tela, existe uma transformação intermediária. Trata-se da transformação de normalização, onde se mapeiam as coordenadas da "window" numa janela, cujas coordenadas x, y variam entre 0 e 1, ou seja, $(0 \leq x \leq 1)$ e $(0 \leq y \leq 1)$. A utilidade da janela normalizada é a de tornar genérico o mapeamento na visualização, tornando-o independente da resolução do dispositivo de exibição. Uma vez na janela normalizada, a transformação das coordenadas destas coordenadas para as coordenadas físicas da tela, é feita pela multiplicação das coordenadas normalizadas pelas coordenadas máximas da tela ou da de alguma "viewport" ativa.

Há uma particularidade na utilização dos sistemas de coordenadas deste sistema, quando se deseja trabalhar com a biblioteca de aminoácidos. Neste caso, a cada aminoácido corresponde uma base de dados contendo informações geométricas e topológicas dos seus elementos (átomos). Para cada aminoácido é fixado um elemento de referência (um átomo) em sua base de dados, estando as posições dos outros átomos representadas em relação ao mesmo, ou seja, em coordenadas relativas. O posicionamento de um dado aminoácido em qualquer localização dentro da janela de visualização, poderá ser feito aplicando-se a esta base inicial uma transformação de translação em x e em y .

5.7. Pintura de Esferas

Visando o preenchimento de esferas, utilizaram-se em MOL3D algumas rotinas que viessem a cumprir esse papel. Basicamente as técnicas envolvidas no preenchimento de áreas delimitadas de tela, dividem-se em duas categorias [11] : "scan conversion" e "seed fill". A primeira consiste na determinação dos pontos extremos de interseção da linha de rastreamento com o contorno da figura. O preenchimento será então feito através da ligação destes pontos.

A segunda técnica inicia o preenchimento, partindo de um ponto interior ao contorno fechado, denominado ponto semente. Daí, o algoritmo prossegue na pesquisa de pontos adjacentes ao semente que estão no interior do contorno.

Ambas as técnicas são utilizadas na pintura de regiões planares, onde o atributo de cor de todos os pixels é constante ou varia de acordo com um padrão de textura pré-estabelecido.

Em computação gráfica tridimensional, a pintura de objetos se utiliza de métodos bem mais complexos, visto que uma esfera iluminada por uma fonte de luz qualquer, se apresenta ao observador, não mais como uma área de cor homogênea. Neste caso, a cada ponto de sua superfície será atribuído um valor próprio de intensidade ou de cor, em função da posição do mesmo nesta

superfície. A simulação deste efeito requer uma "palheta de cores" relativamente rica.

A placa gráfica CGA só dispõe de 4 cores distintas, não havendo nenhuma possibilidade da implementação de técnicas de iluminação e sombreamento. Assim, buscou-se a implementação de outras técnicas de pintura que pudessem passar ao observador a melhor impressão de curvatura nas diferentes esferas pintadas.

Abordam-se a seguir as técnicas de preenchimento testadas para MOL3D. Elas apresentam elementos de ambas as técnicas acima mencionadas.

Antes de se implementar alguma técnica de preenchimento, utilizou-se a rotina do Turbo Pascal (baseada na idéia de "seed fill") destinada a esse fim. Trata-se da rotina "FloodFill", utilizável no preenchimento de áreas fechadas, a partir da informação de um ponto (x,y) interior a essa área e de uma cor que caracteriza a borda (fronteira) da mesma. Apesar de interessante, no sistema MOL3D essa rotina veio a apresentar problemas no que se refere ao preenchimento de área fechada. O problema aconteceria no momento em que duas esferas projetadas se sobrepuham (o ponto semente de uma delas estava contido na outra), e a cor de suas bordas eram a mesma cor. Assim, haveria situações na qual algum setor de uma das áreas deixava de ser preenchido. Com isso, tinha-se que o algoritmo prever essas anomalias e selecionar um ponto semente adequado.

Em face desse problema foram testadas algumas alternativas que viessem a cumprir com sucesso esta tarefa. Assim, inicialmente implementou-se um algoritmo, cujo propósito era utilizar círculos concêntricos, cujos tamanhos dos raios variavam desde o tamanho do raio da esfera até um raio de comprimento unitário. Esse algoritmo, devido a baixa resolução da placa gráfica em uso, apresentou um efeito indesejável : a ocorrência de pontos escuros no interior da região.

Partiu-se em seguida para implementação de um algoritmo de preenchimento por varredura da área ("Scan Line"), onde a idéia era a seguinte : Traçava-se a corda máxima da circunferência (diâmetro), e a partir daí ia-se incrementando o valor de y , de uma unidade, obtendo com isso os valores correspondentes de x (x a esquerda e x a direita), de acordo com equação da circunferência em questão (ver figura 5.7). Este algoritmo "Scan line" cumpriu bem a tarefa do preenchimento de região.

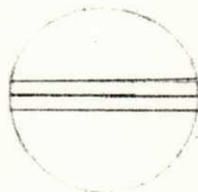


Figura 5.7 : Preenchimento por varredura de área.

Visando a introdução de melhorias, no que se refere à simulação de um efeito de profundidade mais significativo, o algoritmo anteriormente citado ("Scan Line") sofreu algumas melhorias. Assim, elaboraram-se mais dois algoritmos : o primeiro é uma imediata melhoria do "Scan line" citado, onde se utiliza o padrão "InterleaveFill" (padrão reticulado) do turbo pascal e o preenchimento é efetuado por meio da utilização de arcos interiores à região. No segundo utiliza-se a técnica de "halftoning", onde o preenchimento da região é feito por varredura. Neste caso, a região será preenchida através do preenchimento sequencial de células de 2x2 pixels adjacentes, podendo a cada uma delas serem atribuídos os seguintes valores :

Fundo	Fundo
Fundo	Cor

(a)

Cor	Fundo
Fundo	Cor

(b)

Cor	Cor
Fundo	Cor

(c)

Cor	Cor
Cor	Cor

(d)

Cada uma destas 4 células permite simular diferentes intensidades da mesma cor. Em MOL3D foram testadas diferentes combinações, sendo escolhidas apenas duas células : c e d. Foi verificado que a queda de resolução da imagem não ficou criticamente comprometida com o uso desta estratégia (vide figura 3.3 c).

Pelo exposto acima, deu-se a preferência à utilização

definitiva desta última técnica, por ela ter fornecido uma melhor idéia de profundidade.

Serão apresentadas no **Apêndice C** figuras com utilização das técnicas de preenchimento acima utilizadas.

5.8. Visualização de Eixos

Visando proporcionar uma melhor orientação para o observador, implementaram-se procedimentos para desenhar o sistema de eixos na tela. Para evitar confusão entre linhas da figura e linhas do sistema de eixos, utilizaram-se padrões de linha diferenciados ("LineFill" do Turbo Pascal).

5.9. Rolamento de Tela

O sistema permite o rolamento em duas direções (cada uma com dois sentidos) distintos. Para isso, dois procedimentos estão presentes, a saber : scroll-up-down e scroll-right-left.

Antes de uma apresentação mais detalhada desses procedimentos, são convenientes alguns comentários sobre o modo de vídeo em que os gráficos são mostrados e sobre a maneira como os pixels são mapeados na memória pelo mesmo.

A filosofia pretendida para MOL3D é que deva funcionar perfeitamente em qualquer microcomputador compatível com a

família IBM-PC equipado com placa EGA, CGA ou qualquer outra que suporte o modo 4 (gráfico de resolução 320 x 200 a quatro cores). Nesse modo, a tela se apresenta como um "quadriculado" de 320 x 200 pontos, podendo cada um desses pontos assumir qualquer uma entre quatro cores : branco, magenta, ciano e preto. A tela gráfica é mapeada na memória entre os endereços 000B:8000 h e 000B:BE7F h de modo que se pode manipular diretamente os gráficos mostrados através da manipulação de dados dessa área de memória. As 200 linhas são armazenadas em duas regiões distintas: as linhas pares (0,2,..., 198) são armazenadas entre 000B:8000 h e 000B:9F3F h, enquanto as linhas ímpares (1,3,...,199) são armazenadas entre 000B:A000 h e 000B:BF3F h. Tanto as linhas pares quanto as ímpares são armazenadas continuamente, da menor para a maior.

Os procedimentos de rolamento de tela fazem transferências de informações dentro dessas regiões de vídeo. Vejamos cada um deles :

Scroll-up-down (N) -> Produz um rolamento vertical que pode ser para cima ou para baixo, de acordo com o sinal do argumento N. O deslocamento dado às linhas é igual a 2 multiplicado pelo módulo de N. Isso deve-se ao fato de o procedimento não transferir informações entre as duas regiões de memória distintas, reservadas às linhas pares e ímpares. Desse

modo, as informações referentes a uma linha par não poderão ser transferidas para a região referente às linhas ímpares, nem vice-versa. Por esse motivo, uma linha par pode ser transferida ou "rolada", para cima ou para baixo, para o lugar de outra linha par. O mesmo acontece com as linhas ímpares, que podem ser deslocadas, para cima ou para baixo, para o lugar de outra linha ímpar.

Scroll-left-right (N) -> Produz um rolamento horizontal, que pode ser para a direita ou para a esquerda, de acordo com o sinal do argumento N. O deslocamento dado às colunas é igual a 16 multiplicado pelo módulo de N. Isso se deve ao fato de as informações serem transferidas em blocos de 4 bytes. Isto é feito com o objetivo de se aumentar a velocidade da transferência das informações, utilizando-se para isso variáveis do tipo "word", que suportam um inteiro de 4 bytes sem sinal. Como cada pixel é mapeado na memória por 2 bits, 4 bytes mapeiam $(4 * 8 / 2) = 16$ pixels.

Depois que os deslocamentos forem efetuados por cada um dos procedimentos, eles se encarregam de "limpar" a região da tela cujas informações foram deslocadas para outro lugar, sem que após isso nenhuma outra informação tenha sido colocada em seu lugar.

5.10. Traçado das Conexões Interatômicas

A primitiva gráfica para traçar linhas retas está presente na maioria dos desenhos gerados por computador. Em MOL3D o segmento de reta é utilizado para indicar as conexões interatômicas ou interaminoácidos. Neste utilizam-se para o traçado de segmentos de reta, dois recursos básicos : a função "line" do Turbo Pascal e uma função que permite o traçado do referido segmento, ponto a ponto. A necessidade do traçado do segmento ponto a ponto, surge com o propósito de auxiliar as operações de apagamento de esferas e conexões não visíveis. Na tentativa de se resolver este problema do traçado, recorreu-se aos algoritmos já consagrados para esse fim.

Dentre os algoritmos utilizados para traçado de segmentos de reta, destacam-se : o algoritmo de Bresenham e o algoritmo DDA ("Digital Differential Analyzer") [11],[12]. O algoritmo de Bresenham baseia-se em incrementos unitários nas coordenadas (x,y) na direção de maior deslocamento. Isto significa que coeficiente angular (inclinação) da suposta reta ($y = mx + n$) tem ($0 \leq m \leq 1$). Com isso, a cada incremento na direção maior, o valor da outra coordenada pode ser incrementado ou permanecer o mesmo. O algoritmo DDA é também um algoritmo incremental, como será visto mais adiante. O algoritmo DDA na sua forma otimizada e o algoritmo de Bresenham são fáceis para

serem implementados e, além disso não utilizam operações de multiplicação e divisão, sendo assim adequados à implementação em microcomputadores.

Considerando-se a simplicidade e eficiência de ambos os algoritmos citados, na implementação em microcomputadores, torna-se indiferente a escolha de um ou de outro algoritmo. Assim, em MOL3D, a opção foi feita pelo DDA.

O algoritmo utilizado, em MOL3D, para o traçado de segmentos de reta, ponto a ponto, baseou-se no método DDA (Digital Differential Analyzer), numa forma mais otimizada na tentativa de eliminar duas desvantagens básicas : adição com ponto flutuante e a consequente operação de arredondamento que este algoritmo apresenta quando se encontra na sua forma mais trivial. Para resolver isso, usar-se-á apenas a aritmética inteira. A intenção é substituir o valor real da inclinação (coeficiente angular ($m = \text{delta}_y / \text{delta}_x$)) do segmento de reta, pelas suas componentes inteiras , isto é, os valores de $\text{delta}_y = y_f - y_i$ e $\text{delta}_x = x_f - x_i$ (x_i, y_i e x_f, y_f são os pontos extremos do segmento de reta).

O algoritmo será dividido em 4 casos, dependendo da inclinação do segmento de reta. O termo "erro" que aparecerá no algoritmo, indica a distância do ponto plotado à linha atual. Têm-se então :

1) A inclinação está entre 0 e 1 com $(\text{delta}_x \text{ e } \text{delta}_y) \geq 0$:

Se o erro for negativo, o ponto será plotado uma unidade direita de ponto anterior ($x := x + 1$) e será adicionado delta_y ao erro; senão, plotar o ponto em questão, uma unidade para cima e uma unidade à direita de ponto anterior ($x := x + 1$ e $y := y + 1$) e adicionar $\text{delta}_y - \text{delta}_x$ ao erro.

2) Inclinação maior que 1 com $(\text{delta}_x \text{ e } \text{delta}_y) \geq 0$:

Se o erro for negativo, plotar o ponto em questão uma unidade para cima e uma unidade à direita do ponto anterior ($x := x + 1$ e $y := y + 1$) e adicionar $\text{delta}_y - \text{delta}_x$ ao erro; senão, plotar o ponto uma unidade para cima do ponto anterior ($y := y + 1$) e subtrair delta_x do erro.

3) Inclinação entre -1 e 0 com $(\text{delta}_x < 0 \text{ e } \text{delta}_y) \geq 0$:

Se o erro for negativo, plotar o ponto em questão uma unidade à esquerda do ponto anterior ($x := x - 1$) e adicionar delta_y ao erro; senão, plotar o ponto uma unidade para cima e uma unidade à esquerda do ponto anterior ($y := y + 1$ e $x := x - 1$) e adicionar $\text{delta}_x + \text{delta}_y$ ao erro.

4) Inclinação menor que -1 com $(\text{delta}_x < 0 \text{ e } \text{delta}_y) \geq 0$:

Se o erro for negativo, plotar o ponto em questão uma unidade acima e uma unidade à esquerda do ponto anterior ($x := x - 1$ e $y := y + 1$) e adicionar $\text{delta}_y + \text{delta}_x$ ao erro, senão plotar o ponto uma unidade para cima do ponto anterior ($y := y + 1$) e adicionar delta_x ao erro.

5.11. Atribuição de Cores

Em vista dos poucos recursos de cores oferecidos pelo padrão gráfico CGA, procurou-se a utilização de uma adequada política de distribuição de cores que melhor se adequasse ao referido padrão. Para isto, empregou-se a técnica conhecida por "Depth Cueing". A seguir será descrito o mecanismo de funcionamento desta, no escopo de MOL3D.

Antes da apresentação da técnica propriamente dita, serão definidos alguns termos utilizados, que são :

1) Regiões de Profundidade -> o número de regiões de profundidade corresponde ao número de cores que são apresentadas simultaneamente na tela, excluindo-se a cor de fundo. Em MOL3D o número de regiões de profundidade é igual a 3, devido a placa CGA (3 cores simultaneas).

2) Largura de Faixa de Profundidade -> corresponde ao quociente entre a diferença do maior e menor coordenada Z entre as esferas da estrutura e o número de cores simultâneas na tela $(Z_{max} - Z_{min}) / (n_{q} \text{ de cores})$.

Estando de posse das destas definições, o algoritmo pode ser entendido do seguinte modo :

1. Determina-se a primeira região de profundidade (convencionada como sendo a mais escura), tomando-se o intervalo de Z_{min} até $Z_{min} + \Delta Z / 3$, onde $\Delta Z = (Z_{max} - Z_{min})$;

2. Determina-se a segunda região de profundidade (convencionada como sendo a intermediária), tomando-se o intervalo de $Z_{min} + \Delta Z + 1$ até esse valor $+ \Delta Z/3$;

3. Determina-se a terceira região de profundidade (convencionada como sendo a mais clara), tomando-se o intervalo complementar que vai até Z_{max} .

Estando com estes intervalos, pesquisa-se o Z de cada esfera da estrutura, para assim atribuir-lhe sua cor correspondente.

A utilização de monitor RGB permite o uso de uma estratégia diferente, qual seja : a associação de uma cor a cada tipo de elemento atômico presente na estrutura, por exemplo, o usuário pode atribuir a cor verde aos átomos de carbono, vermelho aos de oxigênio, amarelo aos de hidrogênio, e assim por diante. Visando aumentar a palheta de cores foram criadas três cores adicionais através do uso da técnica de " halftoning" usada para a pintura de esferas. Estas cores são : verde + vermelho, verde + amarelo, amarelo + vermelho.

5.12. Animação

Esta operação corresponde à simulação, em tempo real, do passeio do observador em torno da molécula visualizada.

A técnica empregada nesta simulação de animação, consiste, essencialmente, em duas etapas : criação do arquivo de telas e recuperação do mesmo através de uma operação de leitura.

Estando no ambiente interativo de MOL3D, o usuário dispõe, no módulo de animação, da opção de observação da trajetória circular da molécula em torno de um plano, por intermédio das trinta e seis rotações sucessivas, de 10 em 10 graus, aplicadas na mesma. Ressalta-se, entretanto, que os arquivos de telas já estão previamente criados para cada molécula disponível para trabalho, a nível das 36 rotações. O usuário pode utilizar esse módulo, seguindo-se os passos apresentados no item 1.4 do capítulo primeiro desta dissertação.

Salienta-se, por outro lado, que a geração de um arquivo de telas, em princípio, está sendo efetuada fora do ambiente interativo. Consegue-se isso, mediante a ativação da rotina Grava-Tela-No_Arquivo (já descrita no capítulo 4). Caso o usuário introduza uma nova molécula na biblioteca e deseje uma operação de animação sobre esta, terá que seguir alguns critérios. Em primeiro lugar, deve-se ativar de dentro do procedimento que executa rotação, a rotina que efetua a gravação de telas em arquivo. Com isso serão gravadas as referidas 36 rotações no correspondente arquivo.

CONCLUSÃO

A partir da experiência obtida durante o desenvolvimento do sistema MOL3D e durante as análises efetuadas, podem-se apontar alguns problemas, e algumas sugestões podem ser apresentadas para o prosseguimento deste trabalho.

MOL3D foi desenvolvido num microcomputador da linha IBM-PC, com monitor monocromático e placa gráfica CGA. Não é difícil perceber, pelo que já foi exposto, que a magnitude desse trabalho sugere recursos de "hardware" mais sofisticados do que os utilizados, principalmente em relação à placa gráfica e ao monitor. Isso ocorre em função da necessidade de uma maior resolução de pontos, bem como de uma "palheta de cores" mais rica, permitindo assim, uma melhor qualidade na visualização das moléculas, como a simulação de efeitos de profundidade na cena e iluminação mais realista nas superfícies das esferas.

Objetivando a adição de possíveis melhorias posteriores a MOL3D, as sugestões seguintes, são importantes :

- 1) Incorporação de algoritmo para compactação de imagens, na condição de analisar sua eficiência em termos de tempo de processamento ;
- 2) Otimização dos algoritmos, visando redução no tempo de processamento ;

- 3) Ampliação no módulo de animação, objetivando a introdução de opção do observador passear por todo o interior da molécula ;
- 4) Ampliação no projeto da interface de usuário, adicionando-se ícones aos menus, recursos de cores, etc. ;
- 5) Implementação de rotina para fazer projeção perspectiva ;
- 6) Introdução de recursos que permitam realçar a ocorrência de algum tipo de aminoácido no desenho da molécula.

obs.: As sugestões 3), 4) e 5) são dadas para um ambiente de trabalho mais sofisticado do que o utilizado.

Enfim, espera-se que MOL3D cumpra os objetivos ao qual se destina e, ao mesmo tempo contribua para a disseminação de pacotes de "software" voltados para o ensino e pesquisa nas áreas de química, bioquímica e outras correlacionadas.

CAPITULO VII

7. REFERENCIAS BIBLIOGRAFICAS

- [1] Feltre, R., " Química Orgânica ", vol. 3, Editora Moderna (1982).
- [2] Stryer L., "Bioquímica", Editorial Reverté (1979).
- [3] Max, N.L.; " Computer Representation of Molecular Surfaces ", IEEE Computer Graphics and Applications, August 1983, pp. 21-29.
- [4] Turbo Pascal Reference Manual, Borland International Inc., 1988.
- [5] Wirth, N.; " Algorithms + Data Structures = Programs ", Prentice Hall Inc., 1976.
- [6] Schildt, Herbert, " Turbo Pascal Avançado, Guia do Usuário", McGraw-Hill (1988).
- [7] Carroll, David W., " Programação em Turbo Pascal ", McGraw-Hill (1988).
- [8] Tori, R.; Arakaki, R.; Massola A.; Filgueiras, L. , " Fundamentos de Computação Gráfica ",LTC (1987).
- [9] Foley, J.D.; Van Dam, A., " Fundamentals of Interactive Computer Graphics ", Addison-Wesley (1982).

[10] Sutherland, I.; Sproull, R.F.; Schumacker, R.A. , "A Characterization of Ten Hidden-Surfaces Algorithms", Computing Surveys, vol. 6, no 1, march/1974.

[11] Rogers, David F., "Procedural Elements For Computer Graphics", McGraw-Hill (1985).

[12] Newman, W.M.; Sproull, R.f., " Principles of Interactive Computer Graphics ", McGraw-Hill (1984).

Criação de arquivos no Editor Turbo Pascal

A seguir serão descritos os passos necessários para a criação de arquivo no ambiente do editor Turbo Pascal. Estando de posse do Sistema Turbo Pascal versão 5.0 no computador, digita-se < Turbo >, em seguida aparecerá uma tela com as opções do menu principal. Essas opções são : FILE, EDIT, RUN, OPTIONS, DEBUG e BREAK/WATCH. Para criação de arquivo o usuário terá que escolher a opção FILE. Escolhida esta opção, será exibida uma janela com sub-opções :LOAD, PICK, NEW, SAVE e WRITE TO. Escolhe-se então a opção <LOAD> , o Turbo pedirá o nome do arquivo a ser criado, fornecido este nome, o usuário poderá digitar o conteúdo do arquivo, em seguida gravá-lo, utilizando a tecla F2 ou <Alt> seguido da letra S. Uma outra alternativa para criação de arquivo seria a escolha da opção <NEW>, onde imediatamente o usuário digitará o conteúdo do arquivo. Em seguida manda gravá-lo, antes da gravação ser efetuada, o sistema pede o nome do arquivo e, posteriormente executa a gravação do mesmo.

Visando um maior detalhamento do processo descrito acima, considere o seguinte exemplo : suponha que se deseje criar um arquivo cujo nome seja hemoglobina. Suponha que o mesmo pertença à classe das moléculas. Os passos envolvidos na criação deste arquivo serão :

- 1) Carregar o Sistema Turbo Pascal, utilizando o comando <Turbo> ;
- 2) No menu principal do Turbo Pascal selecione a opção <File> e tecle <Enter> ;
- 3) Na janela, que aparece após teclar <Enter>, com as sub-opções : LOAD, PICK, NEW, SAVE e WRITE TO selecione <LOAD> e tecle <Enter>. Feito isso, o Turbo Pascal pedirá o nome do arquivo. No caso desse exemplo, digite hemoglobina.mol . Posteriormente pressione a tecla <Enter>.
- 4) Digite o conteúdo do arquivo ;
- 5) Grave o arquivo. Utilize para isso a tecla F2 ou <Alt> seguido da letra S.

Apêndice B

Informações Complementares para o item 5.3

Teste de Minimax (ou de Bounding Box) : é um teste que verifica a sobreposição possível de duas faces em um dos eixos, verificando as máximas e mínimas coordenadas dos objetos.

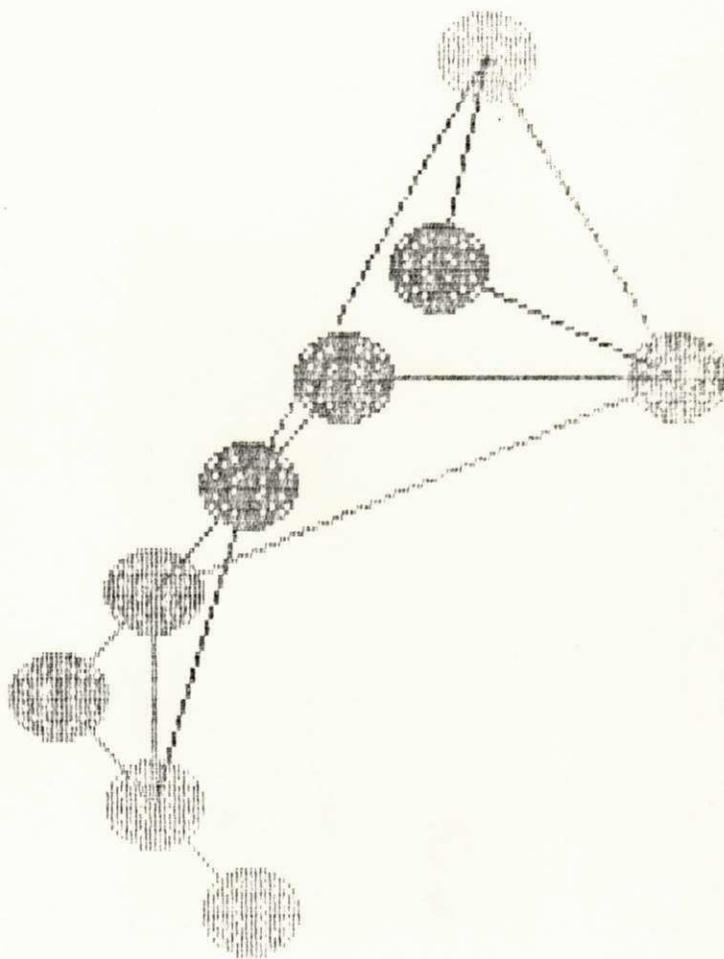
Comparações de Segmentos : é uma operação muito comum nos algoritmos e que tem como resultado a indicação de quais os segmentos de reta dos diversos objetos que estão visíveis numa determinada linha de tela (scan-line). Geralmente esta opção é feita dividindo a linha de vídeo em "spans" delimitados pela projeção na linha de tela dos vértices iniciais e finais dos objetos. Posteriormente cada span tem as profundidades dos seus segmentos testadas, para se descobrir aquele mais próximo do observador.

Face : é um polígono ,geralmente planar, que tem linhas retas como limites.

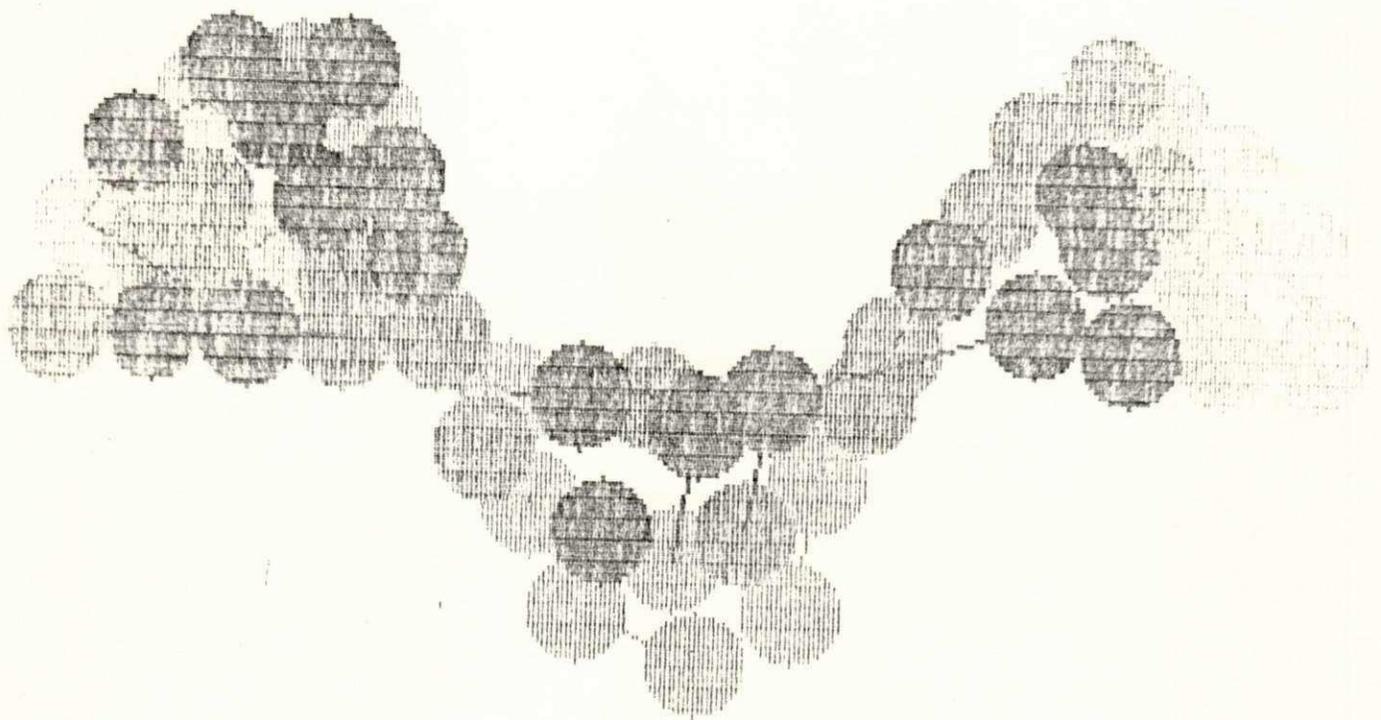
Aresta : é um segmento de linha reta que liga dois vértices adjacentes de uma face.

Apêndice C

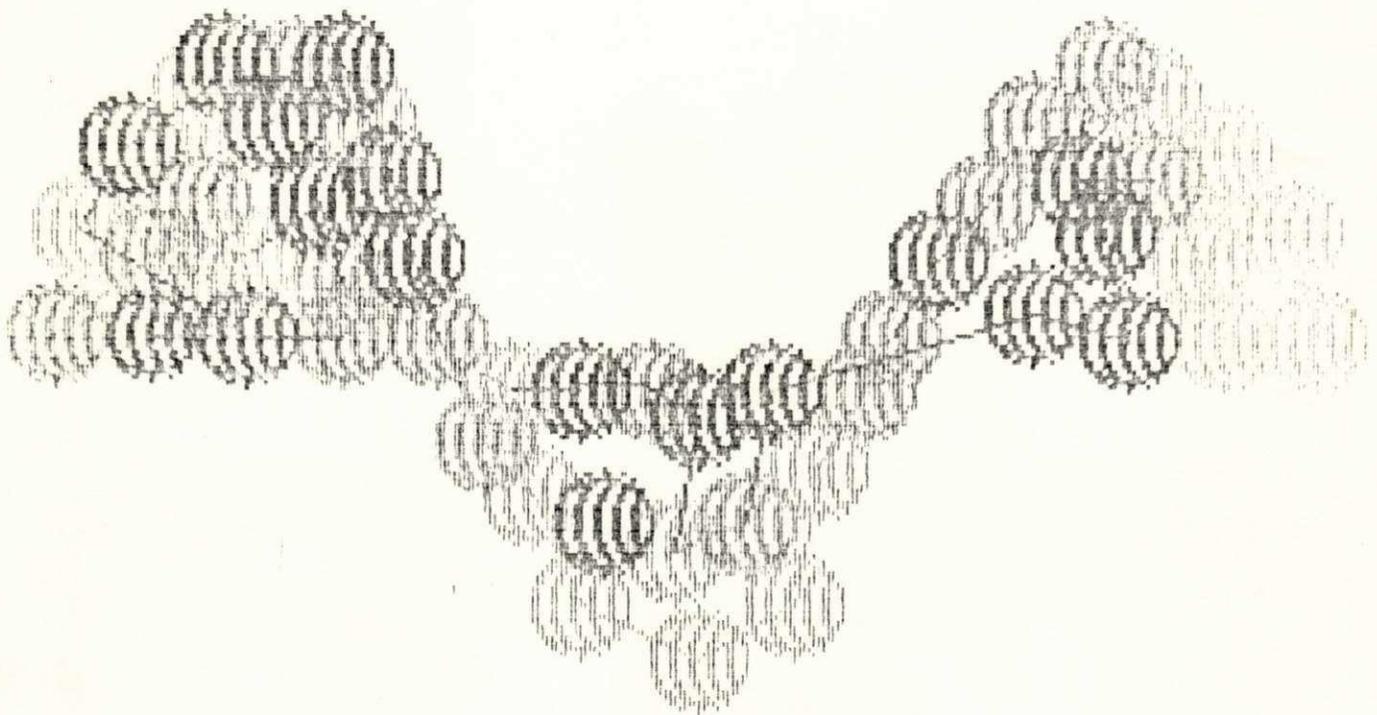
Figuras com as Técnicas de Preenchimento Utilizadas



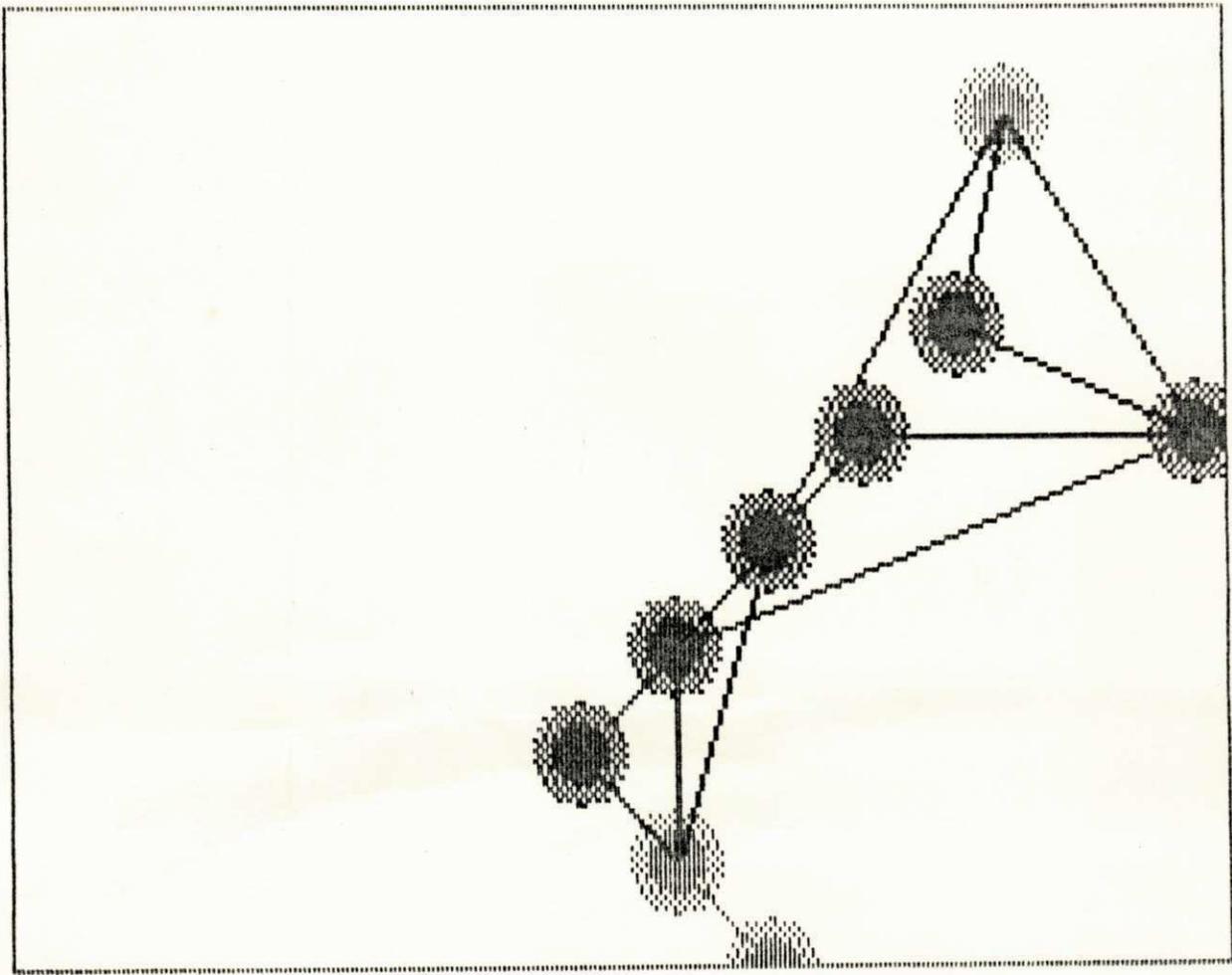
Técnica de pintura da esfera, por meio de círculos concêntricos, com espaçamentos unitários.



Técnica de pintura por varredura da área



Técnica de pintura de esfera usando arcos interiores



Técnica de pintura de esfera usando halftoning