FERNANDO NUNES FERREIRA

UMA NOVA ARQUITECTURA PARA SISTEMAS GRÁFICOS UTILIZADOS NA SÍNTESE INTERACTIVA DE IMAGENS

FACULDADE DE ENGENHARIA
PORTO - 1982

PUBLICAÇÃO SUBSIDIADA PELO
INSTITUTO NACIONAL DE INVESTIGAÇÃO CIENTIFICA

UMA NOVA ARQUITECTURA PARA SISTEMAS GRÁFICOS UTILIZADOS NA SÍNTESE INTERACTIVA DE IMAGENS

FERNANDO NUNES FERREIRA

DISSERTAÇÃO DE DOUTORAMENTO APRESENTADA À

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

5/733-A
UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA
N.º 12536-01
CDU 621.38:681.14:744(043)
Data 23 12 1982
N.N. 6212

681.3(043)
FERF/NOV

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

FACULDADE DE ENGENHARIA

UNIVERSIDADE DO PORTO

043D
F441m
2x.2

PORTO 1982

Agradecimentos

Pretende-se expressar sinceros agradecimentos às seguintes pessoas e entidades:

— Sr. F. Martinez, responsável pela equipa de Comunicação Gráfica do Laboratório IMAG (Laboratoire d'Informatique et de Mathématiques Appliquées de Grenoble), pela orientação prestada, durante os três últimos anos, na execução do trabalho que serviu de base à elaboração desta tese,

— Sr. R. Bouttaz, responsável pelo Laboratório de Micro-electrónica do IMAG, pelas facilidades concedidas, desde o primeiro momento, e que permitiram a realização do protótipo HELIOS,

— Dr. Marques dos Santos da F.E.U.P., pela discussão contínua facultada durante a elaboração desta tese,

— todas as pessoas (e foram muitas) que contribuíram das formas mais diversas, nomeadamente com um estímulo ou um conselho competente,

— Ministério dos Negócios Estrangeiros francês pela bolsa de estudos concedida.

ÍNDICE

ÍNDICE

EXEMPLOS DE IMAGENS PRODUZIDAS COM HELIOS 9

INTRODUÇÃO 13

CAPÍTULO I: CONCEITOS DE BASE LIGADOS À SÍNTESE DE IMAGENS REALISTAS 16

I. 1. INTRODUÇÃO 17

I. 2. AS APLICAÇÕES DAS IMAGENS SINTETIZADAS 17

I. 2.1 — Aplicações para as quais a imagem é uma ferramenta de trabalho 18

I. 2.2 — Aplicações para as quais a imagem é uma finalidade 19

I. 3. A ARQUITECTURA DOS SISTEMAS GRÁFICOS 21

I. 3.1 — Os diferentes tipos de informação contidos numa cena 22

I. 3.2 — As arquitecturas de base dos sistemas gráficos 23

I. 3.2.1 — A arquitectura clássica 25

I. 3.2.2 — A arquitectura ideal 27

I. 3.2.3 — A arquitectura escolhida 27

I. 4. AS OPERAÇÕES ELEMENTARES DA SÍNTESE DE IMAGENS REALISTAS 29

I. 4.1 — A descrição 30

I. 4.2 — A visualização das formas 31

I. 4.2.1 — Projecção paralela aos eixos 32

I. 4.2.2 — Projecção perspectiva 33

I. 4.2.3 — Profundidade modulada em intensidade 33

I. 4.2.4 — Imagens estereoscópicas 34

I. 4.2.5 — Eliminação das linhas escondidas 34

I. 4.2.6 — Eliminação das superfícies escondidas 35

I. 4.3 — A determinação das superfícies visíveis 35

I. 4.3.1 — Algoritmo que utiliza uma memória de profundidade (Z-buffer) 37

I. 4.3.2 — Algoritmo de Atherton e Weiler 39

I. 4.3.3 — Algoritmo de Newell, Newell e Sancha 40

I. 4.3.4 — Algoritmo de Watkins 41

I. 4.4 — O cálculo de iluminação 43

I. 4.4.1 — A modelização dos efeitos da iluminação 43

I. 4.4.2 — O modelo de iluminação de um ponto 44

I. 4.4.3 — Modelos de iluminação de superfícies 48

I. 4.5 — Outros factores a considerar na síntese de imagens realistas 50

I. 4.5.1 — A cor 50

I. 4.5.2 — A modelização da transparência 52

I. 4.5.3 — As sombras 53

I. 4.6 — Síntese final 56

I. 5. CONCLUSÃO 58

CAPÍTULO II - EVOLUÇÃO E TENDÊNCIAS DO HARDWARE GRÁFICO APLICADO À SINTESE DE IMAGENS REALISTAS 59

II. 1. INTRODUÇÃO 60

II. 2. EVOLUÇÃO TECNOLÓGICA DOS DISPOSITIVOS DE VISUALIZAÇÃO 61

II. 2.1 — Os tubos catódicos de varrimento incremental 61

II. 2.2 — Os tubos catódicos de varrimento do tipo TV 65

II. 2.3 — Os tubos de memória 68

II. 2.4 — Os ecrans a plasma 69

II. 2.5 — Os ecrans a laser 70

II. 2.6 — Conclusão 71

II. 3. TENDENCIAIS ACTUAIS DA ARQUITECTURA DOS SISTEMAS GRÁFICOS 72

II. 4. ANÁLISE DE ALGUNS SISTEMAS EXISTENTES 74

II. 4.1 — Metodologia de análise de sistemas gráficos 75

II. 4.2 — Alguns sistemas comercializados 77

II. 4.2.1 — Um sistema simples 78

II. 4.2.2 — Sistemas com software interno 78

II. 4.2.3 — Visualização de sequências animadas pré-calculadas 80

II. 4.2.4 — Simulação de condução 83

II. 4.3 — A investigação actual: alguns sistemas característicos 86

II. 4.3.1 — Visualização de moléculas em tempo real 87

II. 4.3.2 — Visualização de sequências animadas em tempo real com um elevado nível de realismo 88

II. 4.3.3 — Produção de animação interactiva em tempo real 90

II. 5. UMA APRESENTAÇÃO GERAL DE HELIOS 95

II. 5.1 — Características principais 95

II. 5.2 — Outras características 96

II. 5.3 — As facilidades de animação 98

II. 5.4 — O posto de trabalho 99

II. 6. CONCLUSÃO 100

CAPÍTULO III — HELIOS VISTO DO UTILIZADOR 101

III. 1. APRESENTAÇÃO E PRINCÍPIO DE FUNCIONAMENTO 102

III. 1.1 — Relações software/hardware 102

III. 1.2 — Princípio de funcionamento de HELIOS 104

III. 1.3 — Caminho percorrido pela informação em modo de refrescamento do ecran 105

III. 2. MODELIZAÇÃO DAS INFORMAÇÕES PRÉ-SINTETIZADAS 108

III. 2.1 — Os planos de identificação 108

III. 2.1.1 — Natureza das informações memorizadas 108

III. 2.1.2 — A capacidade dos planos 109

III. 2.1.3 — A função JANELA 110

III. 2.1.4 — O cálculo de visibilidade 113

III. 2.2 — O plano de fundo 115

III. 3. AS ETAPAS DA SÍNTESE INTERNA 116

III. 3.1 — A tabela das faces 117

III. 3.1.1 — O brilho 118

III. 3.1.2 — O apontador de texturas 118

III. 3.1.3 — O vector normal 122

III. 3.1.4 — A matriz de projecção 123

III. 3.2 — As texturas coloridas 124

III. 3.2.1 — A memória de texturas 124

III. 3.3 — O cálculo de iluminação 125

III. 3.4 — O módulo de visualização 130

III. 4. A INTERFACE COM O EXTERIOR 130

III. 4.1 — A arquitectura de HELIOS vista do utilizador 130

III. 4.2 — Os formatos de informação utilizados 133

III. 4.3 — O módulo interface 134

III. 4.4 — O módulo de comunicação 136

III. 4.5 — O processo de comunicação 138

III. 4.5.1 — A verificação do estado interno 138

III. 4.5.2 — O modo de acesso 140

III. 4.5.3 — O endereçamento interno 141

III. 4.5.4 — O acesso em leitura 147

III. 4.5.5 — O acesso em escrita 147

III. 4.5.6 — O controlo do cursor 151

III. 5. ALGUMAS CONSIDERAÇÕES SOBRE O SOFTWARE DE VISUALIZAÇÃO 152

III. 6. CONCLUSÃO 155

CAPÍTULO IV — realização de um protótipo 156

IV. 1. INTRODUÇÃO 157

IV. 2. ALGUMAS CONSIDERAÇÕES GERAIS 158

IV. 2.1 — Os conceitos de paralelismo e de "pipe-line" utilizados 158

IV. 2.2 — As tecnologias utilizadas 161

IV. 2.2.1 — Os circuitos integrados utilizados 162

IV. 2.3 — A arquitectura geral de HELIOS 163

IV. 3. O MÓDULO INTERFACE 165

IV. 4. O MÓDULO DE COMUNICAÇÃO 169

IV. 5. A MEMÓRIA DE QUADRO 171

IV. 5.1 — Os planos de identificação 171

IV. 5.1.1 — A memorização das informações de identificação 172

IV. 5.1.2 — A função janela 176

IV. 5.1.3 — O cálculo de visibilidade 181

IV. 5.2 — O plano de fundo 184

IV. 6. A TABELA DAS FACES 184

IV. 7. O CÁLCULO DE TEXTURAS 186

IV. 8. O CÁLCULO DE ILUMINAÇÃO 190

IV. 9. O MÓDULO DE VISUALIZAÇÃO 192

IV. 10. O CURSOR 195

IV. 11. O MÓDULO DE CONTROLO 195

IV. 12. CONCLUSÃO 197

CAPÍTULO V — CONCLUSÃO 200

REFERÊNCIAS BIBLIOGRÁFICAS 204

EXEMPLOS DE IMAGENS OBTIDAS COM HELIOS

As passagens entre as imagens representadas nas fotografias 1 e 2, por um lado, e 3, 4, 5 e 6, por outro, não exigiram nenhum tempo de cálculo à unidade de tratamento associada a HELIOS. Estas modificações foram calculadas localmente, em tempo real, pelo processador gráfico.

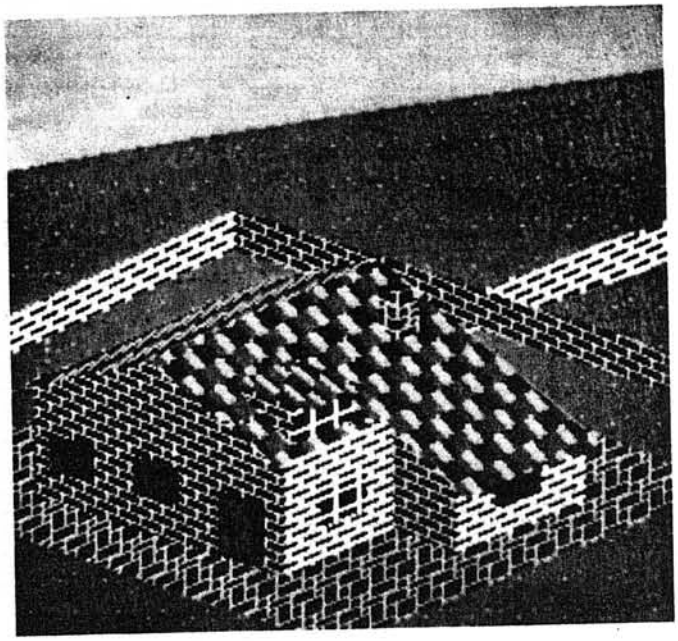
As fotografias referidas (originalmente a cores) ilustram algumas das possibilidades oferecidas:

— texturas (paredes da moradia, fotografias 1 e 2; "abat-jour" do candeeiro, fotografias 3, 4, 5 e 6, ...)

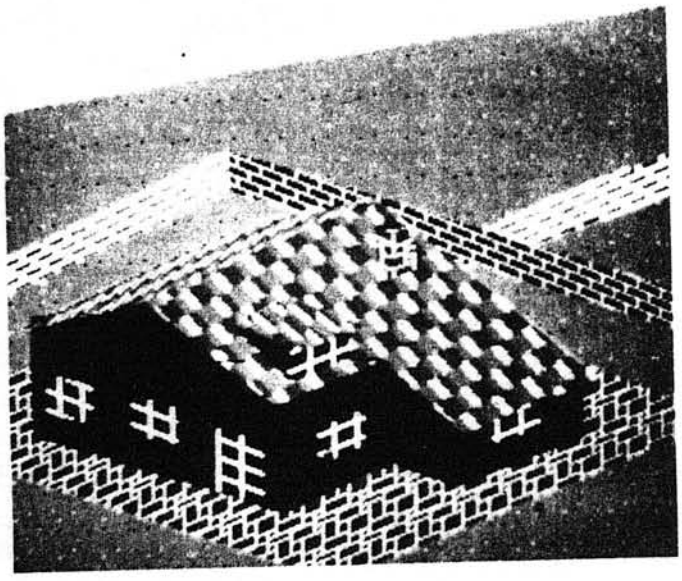
— posição da fonte luminosa (fotografias 5 e 6, ...)

— brilho dos materiais (pé do candeeiro, fotografias 5 e 6, ...)

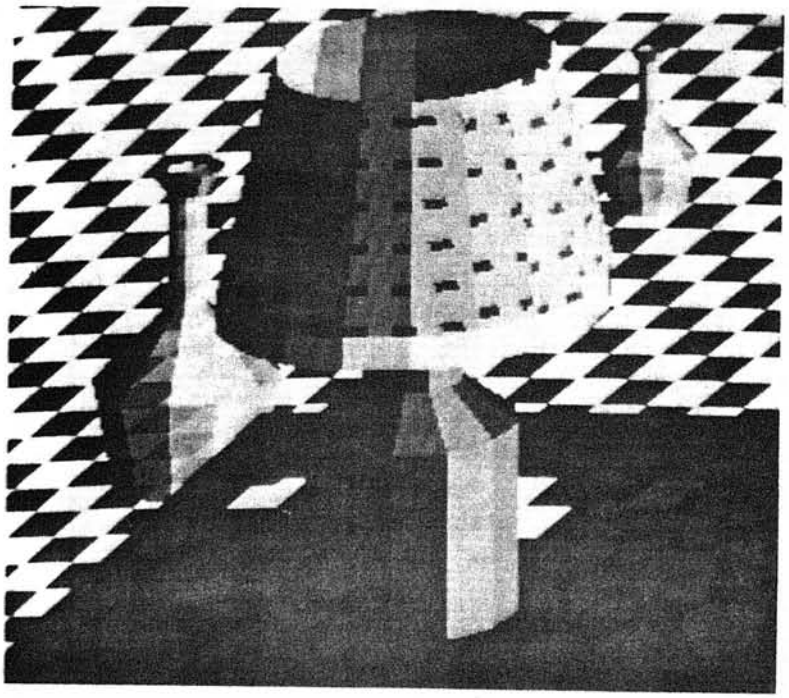
— simulação da transparência (garrafas, fotografias 3 e 6).



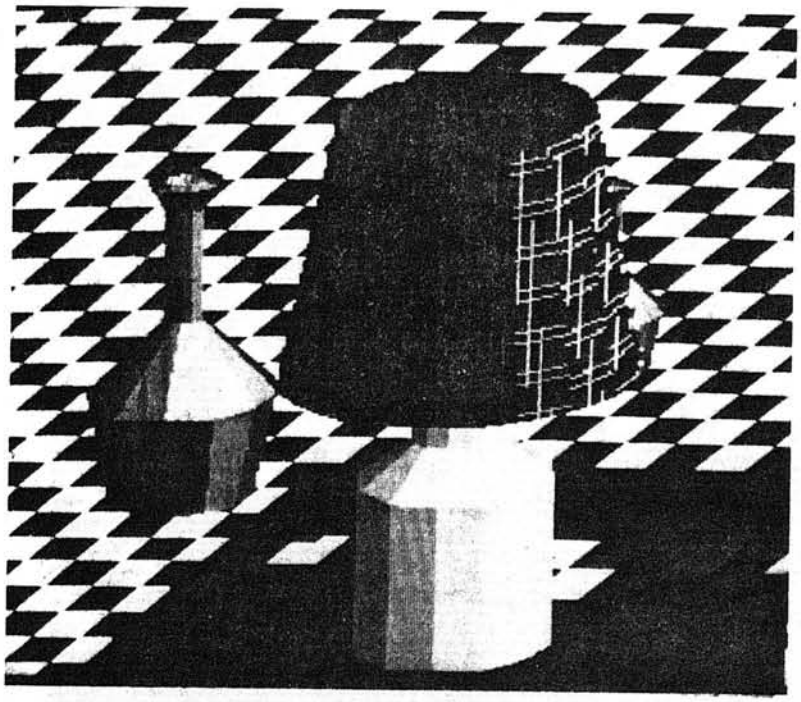
Fotografia 1



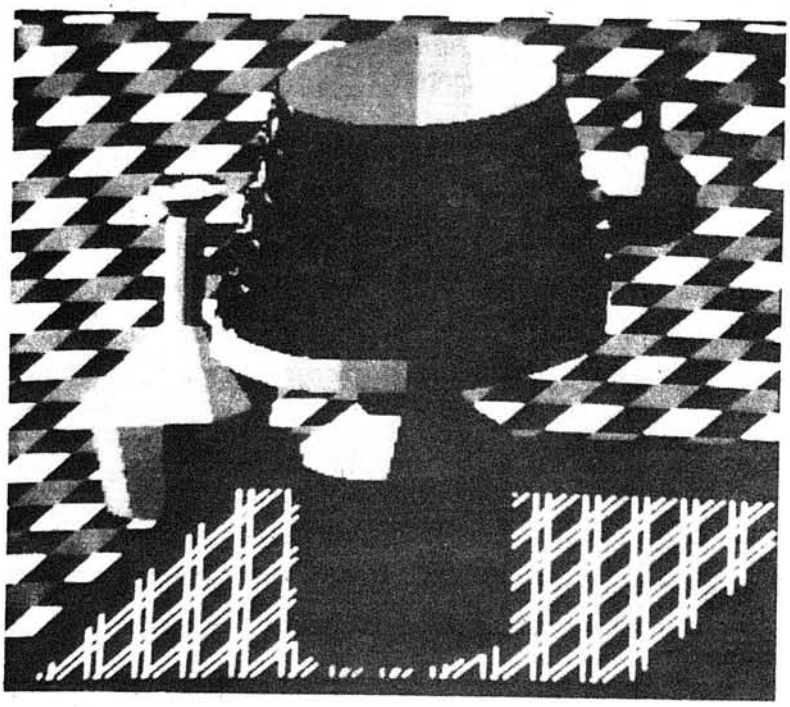
Fotografia 2



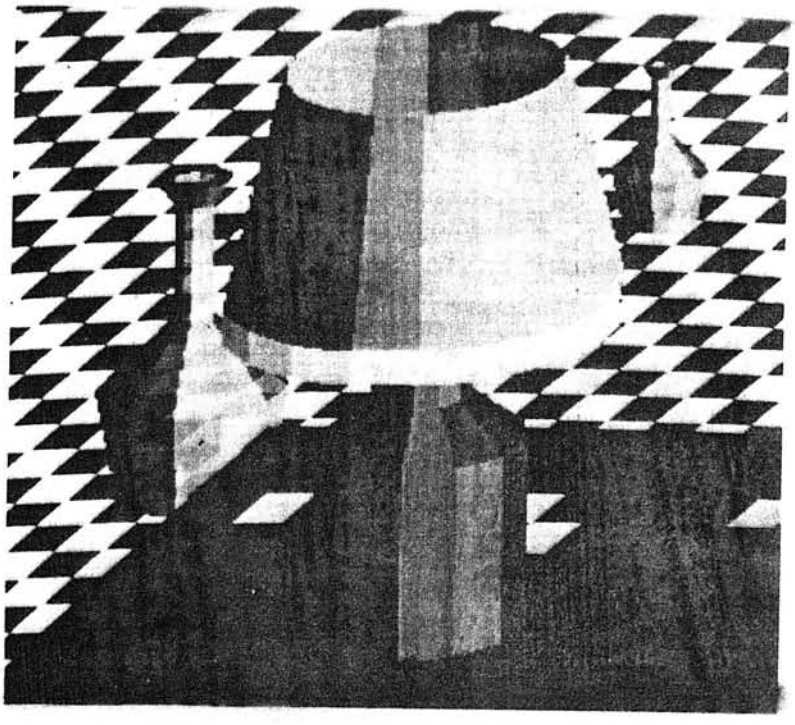
Fotografia 3



Fotografia 4



Fotografia 5



Fotografia 6

INTRODUÇÃO

INTRODUÇÃO

A influência da Computação Gráfica num número sempre crescente de actividades humanas tem a sua principal justificação na facilidade de compreensão, por parte do utilizador, da informação contida numa imagem. Outras razões, em particular a evolução tecnológica constante dos circuitos integrados e dos monitores de varrimento do tipo TV, têm facilitado a integração deste ramo da Informática nas aplicações mais diversas. Citamos, como exemplos, a realização de filmes didácticos e publicitários, projecto de auto-estradas, concepção de peças mecânicas, de tecidos e de máscaras de circuitos integrados e simulação de condução, esta última exigindo, simultaneamente, imagens com aspecto realista e um nível de interacção elevado.

A investigação actual neste domínio está essencialmente orientada para a pesquisa de soluções satisfatórias para os inúmeros problemas levantados pelas aplicações que exigem, como suporte ou como finalidade, a criação de *imagens realistas em tempo real.* Neste tipo de aplicações, o realismo e o tempo real exibem um carácter antagónico muito evidente: De um lado, toda uma série de tratamentos complexos e de longa duração pedidos por uma visualização realista e que englobam os cálculos relativos à visibilidade de uma cena, à cor, às texturas, à transparência e à iluminação; de outro lado, as condições impostas pelo tempo real, que fazem realçar o curto intervalo de tempo para as respostas necessárias.

As actividades de investigação que merecem especial atenção da Equipa de Computação Gráfica do Laboratório I.M.A.G. (Laboratoire d'Informatique et de Mathématiques Appliquées de Grenoble, onde foram realizados os trabalhos que fundamentam esta tese) integram este domínio actual, no quadro de um estudo cuja ideia de base é devida a F. Martinez [Mar 81]. Esta ideia, que se apoia num *sistema de computação gráfica estruturado de maneira original,* visa a criação, a manipulação e a

visualização interactiva de imagens com aspecto realista e põe em evidência dois campos de investigação:

— O estudo dos problemas que podem ser resolvidos por via "software" e que constitui o projecto CLOVIS [Mar 80],

— O estudo dos problemas que resultam do projecto e realização material (hardware) de um sistema destinado à síntese de imagens realistas, projecto HELIOS, que traduz o *objectivo fundamental desta tese.*

Inicia-se o Capítulo I com uma breve apresentação das actividades humanas onde a integração das técnicas gráficas é, ou pode vir a ser, muito importante. Esta integração cria ainda sérios problemas em actividades onde imagens com aspecto realista devem ser usadas interactivamente. Um estudo sobre a síntese destas imagens dará uma ideia dos tratamentos, por vezes complexos, que são geralmente exigidos.

A resposta para este género de problemas afasta-se de soluções programadas (software), harmonizando-se em geral com soluções de natureza "hardware". O Capítulo II considera a evolução e a tendência actuais neste sentido e situa HELIOS nesta linha evolutiva, apresentando-o como uma primeira resposta para esta situação.

O Capítulo III agrupa a informação necessária aos potenciais utilizadores de HELIOS, ao facultar-lhes o seu princípio de funcionamento e o seu modo de utilização.

A realização do protótipo HELIOS constitui o tema do Capítulo IV. Neste se apresentam, para cada módulo, a respectiva estrutura, os algoritmos utilizados, alguns pormenores de realização e a justificação das opções tomadas.

O Capítulo V contém uma análise final dos resultados obtidos, permitindo definir alguns temas originais a investigar futuramente.

CAPÍTULO I

CONCEITOS DE BASE LIGADOS À SÍNTESE DE IMAGENS REALISTAS

CAPÍTULO I: CONCEITOS DE BASE LIGADOS À SÍNTESE DE IMAGENS REALISTAS

I. 1. INTRODUÇÃO

A utilização de sistemas de computação gráfica em certas aplicações exige, cada vez mais, a visualização de imagens que exibam um nível de realismo importante. Esta necessidade levanta vários problemas aos projectistas de tais sistemas, uma vez que a síntese de imagens realistas envolve tratamentos vários que, na ausência de material especializado, torna defícil, ou até mesmo impossível, a sua utilização interactiva.

O presente capítulo principia com uma breve panorâmica das aplicações onde a imagem é utilizada como ferramenta de trabalho e/ou como finalidade. Seguidamente analisam-se diferentes hipóteses para a estruturação de um sistema de computação gráfica face ao problema da síntese de imagens realistas; é posta em evidência a necessidade de soluções que possam ser realizadas por via "hardware". Finalmente, percorre-se as diferentes operações elementares da síntese de imagens, o que permite deduzir as possíveis incidências do "hardware" sobre cada uma delas.

I. 2. AS APLICAÇÕES DAS IMAGENS SINTETIZADAS

As aplicações que envolvem uma comunicação visual com um computador multiplicam-se constantemente, fenómeno que se justifica pela incontestável supremacia da imagem como meio de comunicação. A potência de cálculo de um computador associada a um dispositivo de visualização permite aumentar consideravelmente as

informações veiculadas pelas imagens, ao minimizar o esforço de compreensão exercido pelo utilizador.

Estas aplicações podem ser agrupadas em duas grandes classes:

I. 2.1 — **Aplicações para as quais a imagem é uma ferramenta de trabalho**

Esta classe comporta, em geral, o conjunto de aplicações que utilizam a Concepção Assistida por Computador (C.A.D.), para as quais a comunicação visual é uma ferramenta fundamental.

Podemos dizer que, nos nossos dias, este género de aplicações produz unicamente os esquemas do produto a realizar, sendo as imagens assim obtidas do tipo "desenho" [fig. 1.1]. Esta forma de visualização, onde os objectos são representados, em geral, pelos seus contornos, exibe um nível de realismo extremamente baixo.

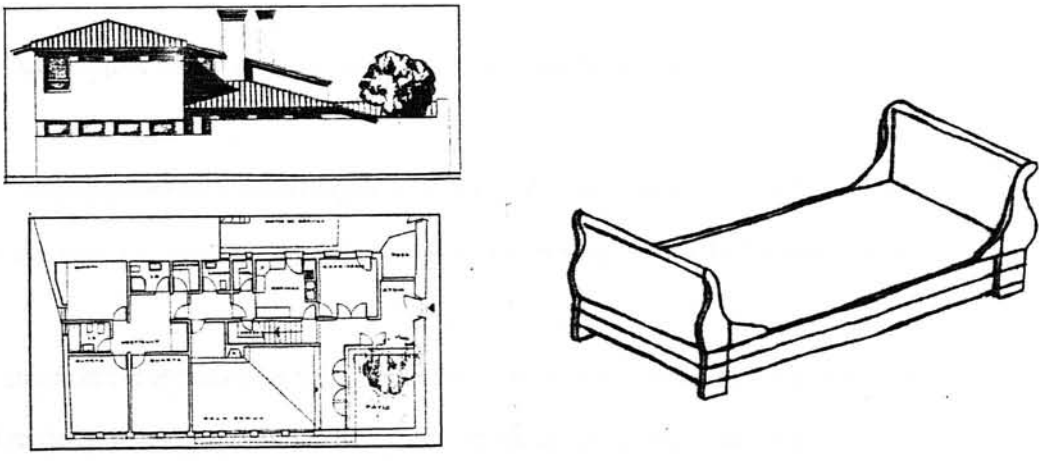


Fig. 1.1 – Exemplos de representações do tipo "desenho".

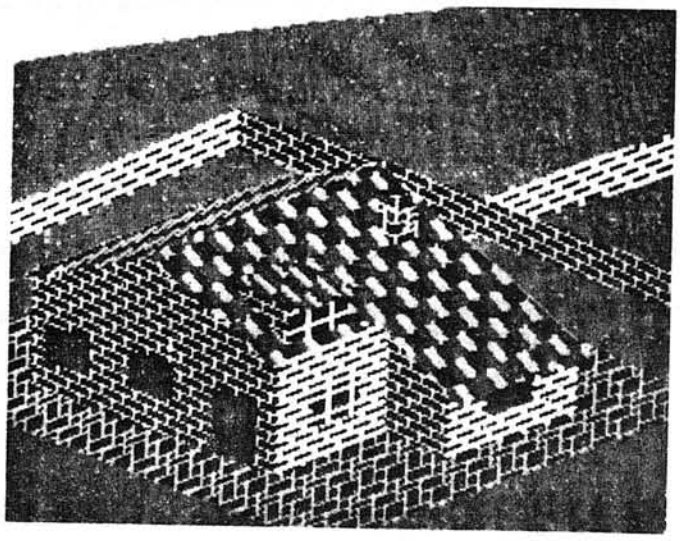


Fig. 1.3 – Uma imagem realista sintetizada com HELIOS.

Podemos considerar nesta classe:

— As aplicações que produzem imagens estáticas utilizadas, por exemplo, na concepção de tapeçarias, de tecidos ou de moradias.

Aqui, o factor tempo não cria problemas de difícil resolução, pois é medido à escala do tempo de resposta do utilizador.

— As aplicações que produzem imagens dinâmicas em tempo real, entre as quais se pode distinguir a simulação de condução.

É precisamente neste domínio que surgem os grandes problemas da computação gráfica, pois deve tomar-se em linha de conta, simultaneamente, o realismo da imagem e o factor tempo real. Esta última condição significa que o sistema deve ser suficientemente rápido para efectuar as modificações pedidas em 40 ms, o que não é nada evidente quando se considera a longa lista de parâmetros envolvidos na visualização de imagens realistas.

I. 3. A ARQUITECTURA DOS SISTEMAS GRÁFICOS

A transmissão e o tratamento das informações gráficas entre o utilizador e o programa de aplicação efectua-se geralmente através de dois módulos "software":

— Descrição da cena e Visualização [fig. 1.4]. Com estes dois módulos pretende-se introduzir uma certa independência da aplicação em relação aos dispositivos de interacção disponíveis.

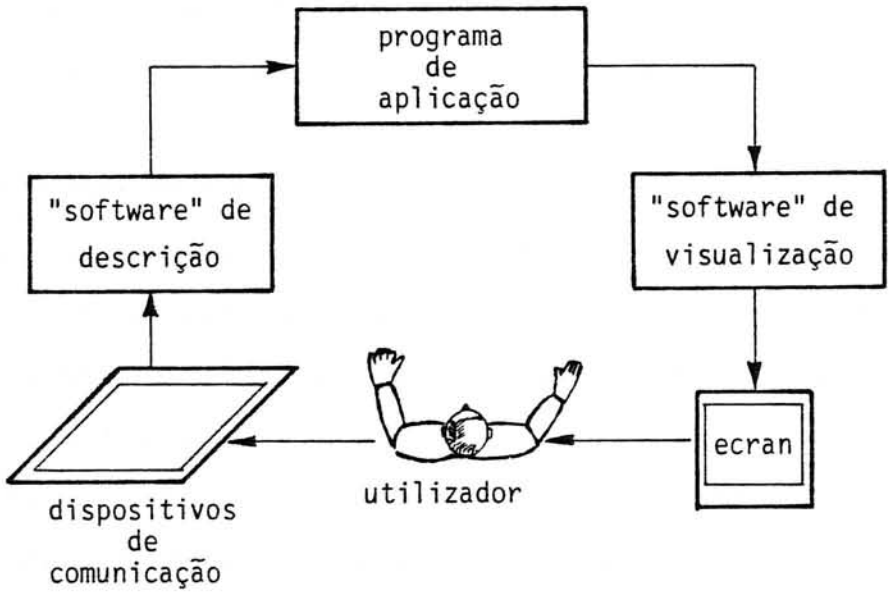


Fig. 1.4 – A circulação das informações gráficas.

Comecemos por examinar os diferentes tipos de informações gráficas em jogo e, seguidamente, pormenorizemos os caminhos percorridos por cada um deles no seio do sistema. Estas acções permitem introduzir naturalmente as arquitecturas de base dos sistemas gráficos interactivos.

I. 3.1 — **Os diferentes tipos de informação contidos numa cena**

Uma imagem sintetizada com aspecto realista deve suscitar num eventual observador, uma impressão visual idêntica à que ele teria face à cena real correspondente. A síntese de imagens realistas deve então criar e aglutinar os mesmos tipos de informação que são veiculados durante a análise de uma cena real. Estas informações podem ser vantajosamente postas em evidência através de um exemplo simples de análise visual.

Um observador começa por constatar que uma cena é composta por um certo número de objectos, agrupados de uma certa maneira, o que faz imediatamente ressaltar a existência de uma estrutura; esta observação evidencia as *informações Estruturais,* que podem estar igualmente presentes ao nível do objecto (caso de um objecto composto por outros estruturalmente mais simples).

Depois desta análise global, o observador concentra a atenção sobre os objectos individualmente. Vê, por exemplo, uma forma cónica e aberta na parte superior — *informações Morfológicas* — de cor esbranquiçada e ligeiramente brilhante — *informações de Aspecto* —.

Analisando com um pouco mais de atenção vê, que o objecto em causa, apresenta reflexos, devidos certamente a uma fonte luminosa, não muito forte, situada sobre a direita e um pouco no alto — *informações de Iluminação* —. Ao tentar definir a posição deste objecto em relação aos outros, isto é, ao determinar a sua atitude no espaço, põe em evidência as *informações Geométricas.*

O observador pode agora concluir esta primeira análise, pois reconheceu o objecto "copo" — *informações de Identificação* — [fig. 1.5].

Assim, sucessivamente, todos os objectos da cena podem ser identificados.

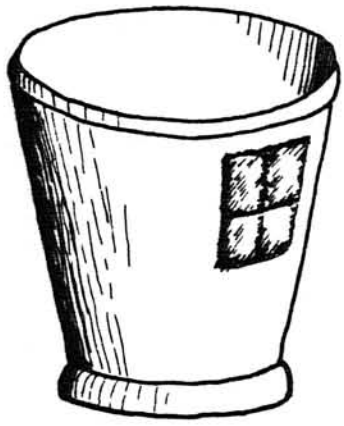


Fig. 1.5 – O objecto identificado.

I. 3.2 — **As arquitecturas de base dos sistemas gráficos**

Quando se analisam as diferentes tarefas atribuídas aos módulos de Descrição e de Visualização [fig. 1.4], distingue-se três fases fundamentais [fig. 1.6] [Mar 79]:

— A *Descrição* que, a partir das informações fornecidas pelo utilizador, gera o *modelo de descrição* da cena (maquete).

— A *Preparação à Visualização* que, a partir do modelo de descrição, cria um *negativo* (analogia com o processo fotográfico), ou seja, uma imagem latente num estado não visualizável.

— A *Síntese Final* que, ao aglutinar todas as informações que lhe chegam, produz a *imagem* num dispositivo de visualização.

Em geral, é durante esta última fase que se resolvem as incompatibilidades entre as informações contidas no "negativo" e as possibilidades reais do dispositivo de visualização em jogo, ao introduzir-se uma camada suplementar de "software" — software de Visualização —. Este realiza a interface entre as funções lógicas desejadas e as funções físicas disponíveis ao nível do processador gráfico [fig. 1.7].

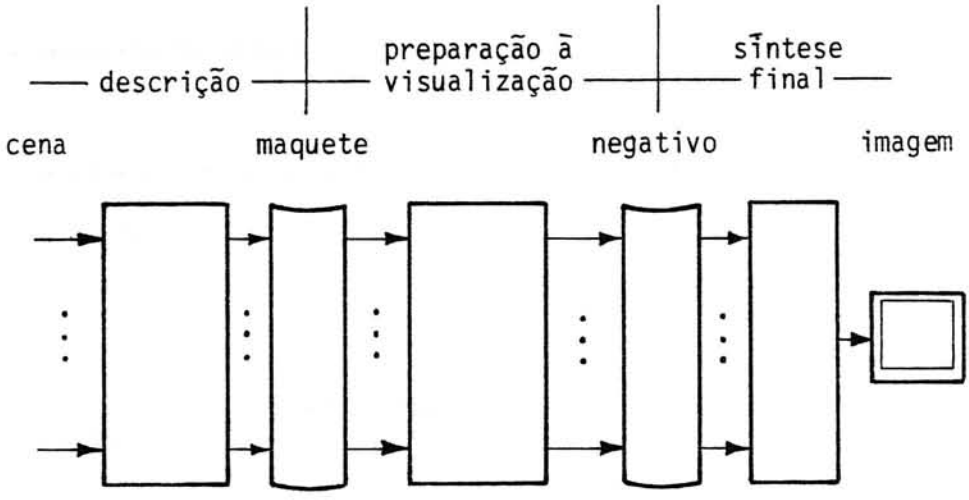


Fig. 1.6 – As três fases fundamentais a considerar num sistema gráfico.

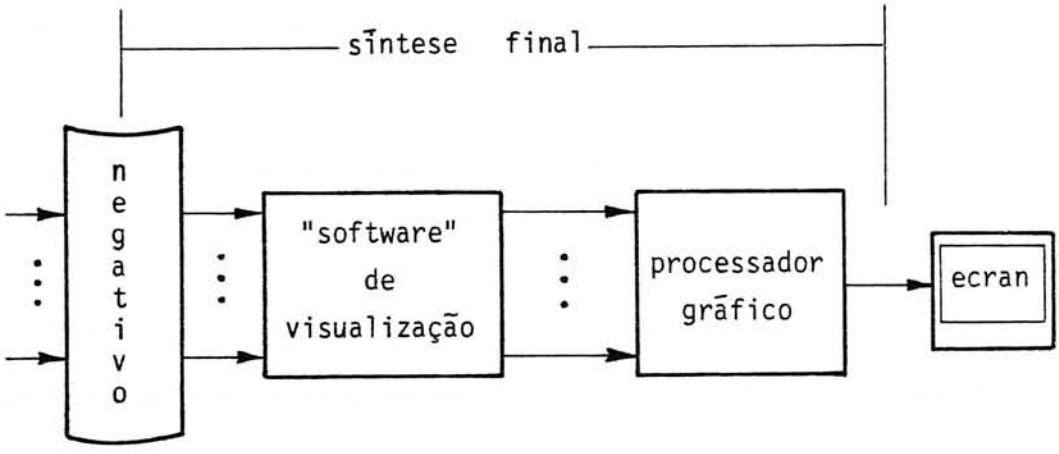


Fig. 1.7 – A função do software de visualização.

Quando se consideram os diferentes tipos de informações gráficas tratadas em cada uma destas fases, encontramos várias arquitecturas possíveis para os sistemas gráficos.

I. 3.2.1 — **A arquitectura clássica**

Esta arquitectura é caracterizada por uma "síntese distribuída" pelas diferentes fases do processo [fig. 1.8].

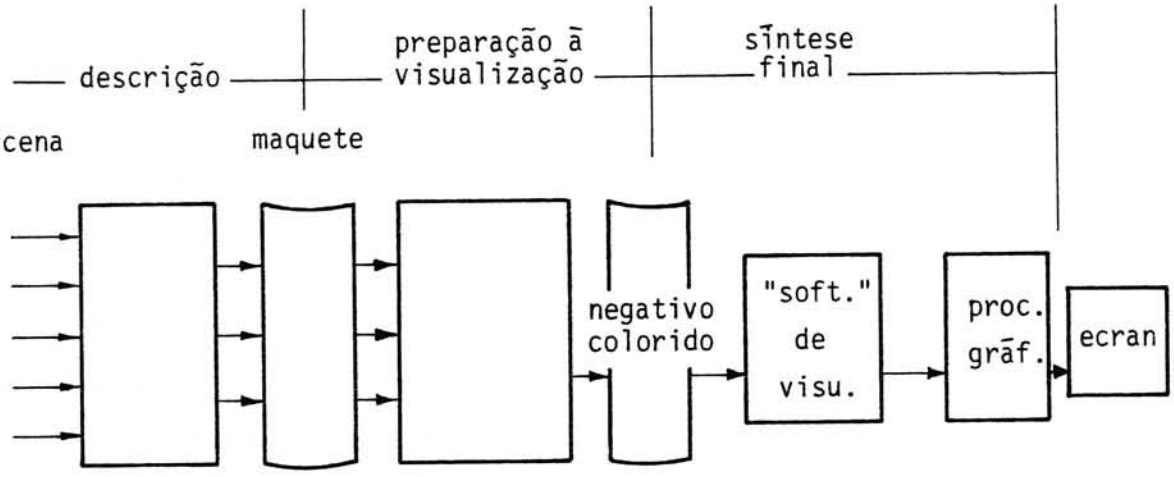


Fig. 1.8 – A arquitectura clássica.

Com efeito, o número de tipos de informações gráficas em jogo decresce no sentido cena-imagem, ao passar de fase em fase; a síntese final torna-se, desta maneira, insignificante.

Do ponto de vista da visualização esta solução harmoniza-se perfeitamente com um *processador gráfico* extremamente simples e a função do "software" de Visualização resume-se a carregar a *memória de quadro,* neste caso, designada por *memória imagem* [fig. 1.9]. As possibilidades oferecidas por um processador gráfico evoluído vêem-se, neste caso, desaproveitadas, uma vez que a maior parte do tratamento é feita pelos módulos de descrição e de preparação à visualização, portanto de uma maneira totalmente independente daquele dispositivo.

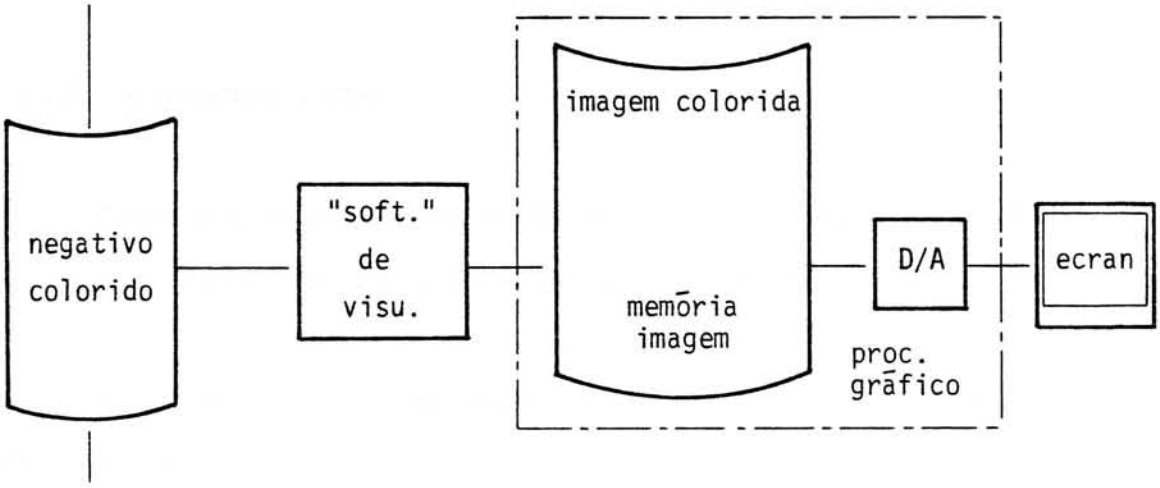


Fig. 1.9 – Um processador gráfico extremamente simples.

Do ponto de vista da interacção homem-máquina, esta solução mostra-se igualmente pobre, pois:

— As informações de identificação não chegam ao "negativo", o que prejudica a implementação de uma função de designação sobre o écran.

— A modificação de um tipo de informação exige um novo cálculo da imagem, que tomará em linha de conta todos os outros tipos de informação em jogo.

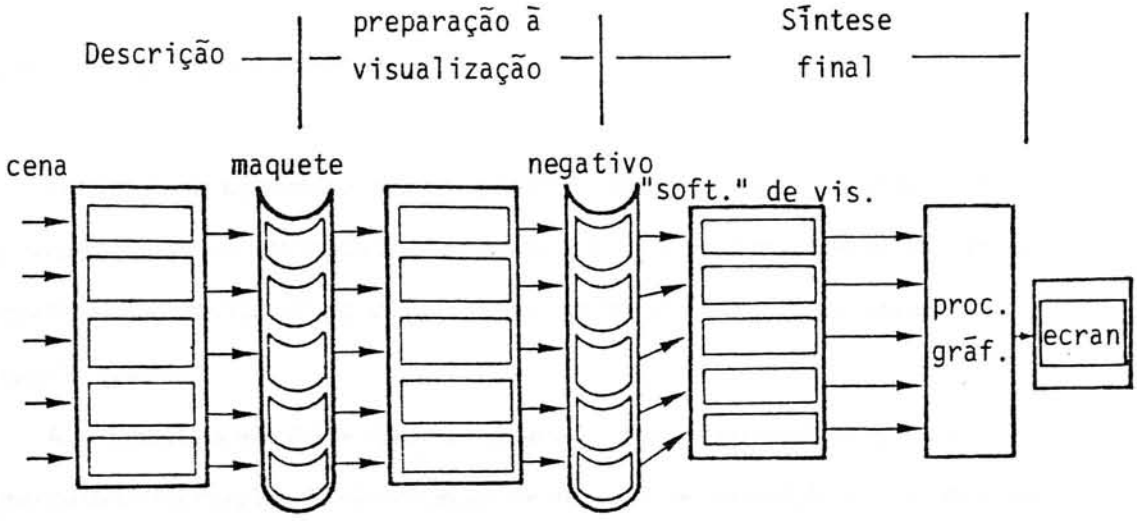


Fig. 1.10 – A arquitectura ideal.

I. 3.2.2 — **A arquitectura ideal**

A ideia de base desta arquitectura consiste em conservar separado o tratamento dos diferentes tipos de informações até ao processador gráfico [fig. 1.10].

Neste caso, todos os problemas que caracterizavam a arquitectura clássica são eliminados, pois:

— As possibilidades do processador gráfico são agora aproveitadas de uma maneira satisfatória, limitando-se o "software" de Visualização a activar as funções daquele.

— A modificação de um qualquer tipo de informação é realizada independentemente dos outros e, além disso, ao ritmo característico do "hardware".

Obviamente, o grande inconveniente que podemos apontar a esta solução reside na impossibilidade da sua realização integral.

I. 3.2.3 — **A arquitectura escolhida**

Entre os dois extremos indicados podemos individualizar várias soluções de compromisso, tendo por objectivo comum "empurrar" a síntese dos diferentes tipos de informação até ao "software" de Visualização e, na medida do possível, até ao processador gráfico.

A arquitectura escolhida conserva separado, até ao processador gráfico, o tratamento das informações de identificação, de aspecto, de iluminação e uma parte das informações geométricas [fig. 1.11].

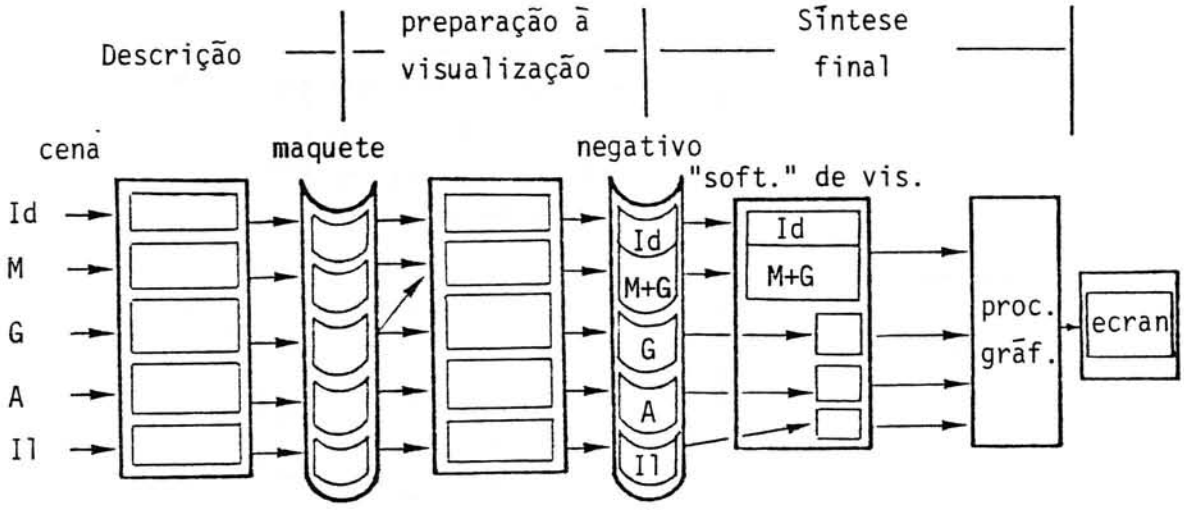


Fig. 1.11 – A arquitectura escolhida.

A escolha desta solução de compromisso é justificada pela:

— sua originalidade

Não conhecemos nenhum sistema cujo processador gráfico considere, simultaneamente, as informações de identificação, de aspecto (texturas coloridas), de iluminação (fonte luminosa colorida e luz ambiente) e uma parte das informações geométricas (projecção das texturas) [cf. II. 5].

— sua modularidade

Desenvolvimentos futuros são facilmente integrados por uma simples modificação ao "software" de Visualização [fig. 1.12].

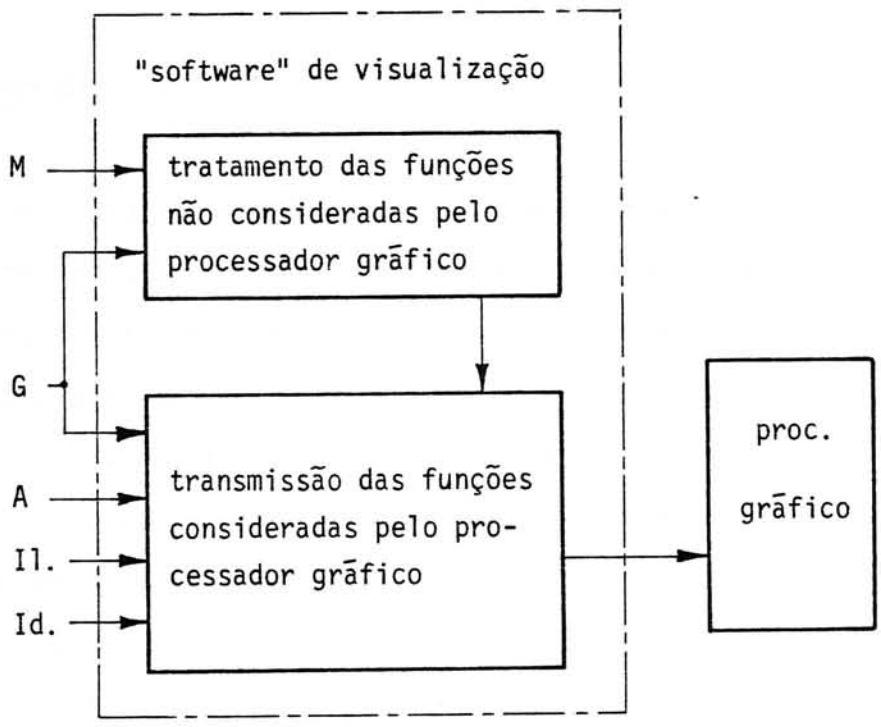


Fig. 1.12 – A integração de uma nova função do processador gráfico resume-se à modificação ou à supressão de uma parte do software de visualização.

I. 4. AS OPERAÇÕES ELEMENTARES DA SÍNTESE DE IMAGENS REALISTAS

Como já tivemos oportunidade de referir [cf. I. 3.2], a *síntese de imagens realistas* envolve várias operações elementares relativas à:

— Descrição

— Preparação à Visualização

— Síntese final

Analisaremos seguidamente estas operações, indicando para cada uma delas as incidências possíveis de uma realização de natureza "hardware".

I. 4.1 — **A descrição**

A existência desta operação é devida ao facto de grande parte das imagens manipuladas por computador terem a sua origem numa descrição produzida pelo utilizador. Raríssimas vezes, salvo em aplicações de Processamento de Imagem, a informação tem origem numa câmara video que percorre uma cena real.

Esta operação deve produzir, a partir de informações e comandos fornecidos pelo utilizador, um *modelo de Descrição* de uma cena, o qual tomará em linha de conta:

— os objectos que a compõem

— as condições de iluminação

— o observador e as condições de observação.

Trata-se, portanto, de uma operação longa e extremamente penosa, se o módulo de Descrição não fornecer utensílios tendentes a minimizar a quantidade de informação que circula entre o utilizador e a máquina.

Infelizmente, os trabalhos que tratam este problema (ser bibliografia em [NeS 79]), [Gre 77] consideram essencialmente as informações morfológicas, omitindo, em geral, todas as outras.

As diferentes técnicas utilizadas na criação e descrição das informações morfológicas baseiam-se em:

— volumes elementares [Bra 75]

— superfícies não planas [Luc 79]

— faces planas [The 72].

Atendendo à natureza desta operação, parece-nos que a possível influência do "hardware" sobre ela se limitará a fornecer os utensílios necessários à entrada de informação (mesa de digitalização, caneta luminosa, ...).

I. 4.2 — **A visualização das formas**

Nas fases Preparação à visualização e Síntese final, considera-se todo o processamento a que foi sujeita a informação contida no modelo de descrição até à obtenção da imagem correspondente no ecran. Surge-nos assim uma questão:

— Como produzir imagens realistas de cenas 3D sobre ecrans 2D?

Várias técnicas são conhecidas, que diferem entre elas, fundamentalmente, pelo nível de realismo que oferecem e pelo tempo de resposta que exibem. Segue-se uma breve descrição destas técnicas [NeS 79], que são apresentadas segundo a complexidade crescente das respectivas realizações.

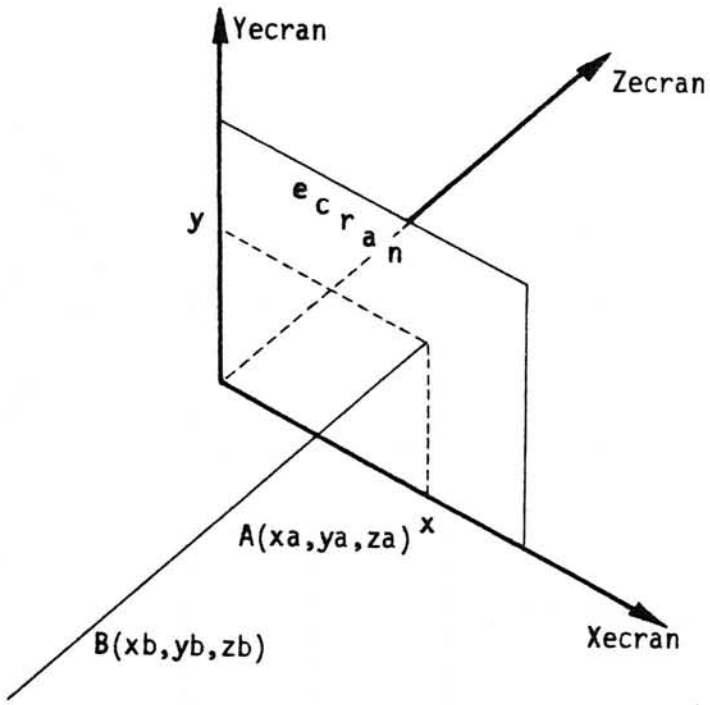


Fig. 1.13 – O princípio da projecção paralela.

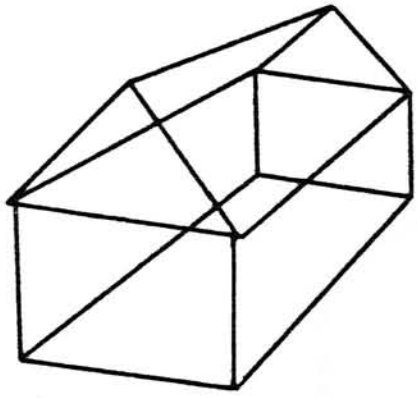
I. 4.2.1 — **Projecção paralela aos eixos**

Quando se utiliza uma projecção paralela aos eixos, o observador considera-se colocado a uma distância infinita do ecran.

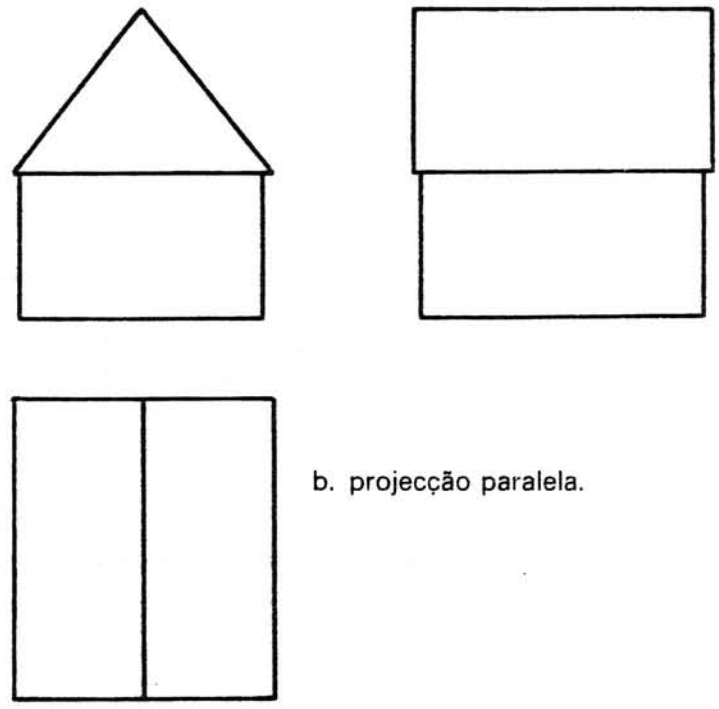
No exemplo apontado na figura 1.13, a visualização do ponto A apresenta como coordenadas (x, y), o que significa que a posição da imagem de um ponto situar-se-á sobre a linha perpendicular ao ecran e que passa por aquele ponto. Quando se encontram vários pontos sobre esta linha, será visualizado o que se encontra mais próximo do observador.

Ainda na figura 1.13, o ponto B esconde o ponto A, uma vez que za é superior a zb.

Esta técnica, muitas vezes utilizada por arquitectos na representação de moradias, está na origem de resultados cuja interpretação não é imediata e que se distanciam exageradamente das imagens com um aspecto realista [fig. 1.14 b].



a. projecção perspectiva



b. projecção paralela.

Fig. 1.14 – Visualização de uma moradia

I. 4.2.2 — **Projecção perspectiva**

Com esta técnica dá-se.a impressão de profundidade, diminuindo o tamanho dos objectos à medida que estes se afastam do observador [fig. 1.14 a].

Os resultados apresentam-se com um aspecto relativamente familiar, pois o processo descrito é idêntico ao utilizado pelo olho humano.

I. 4.2.3 — **Profundidade modulada em intensidade**

De acordo com esta técnica, as linhas apresentam-se tanto mais grossas (ou mais brilhantes) quanto mais perto se encontram do observador [fig. 1.15].

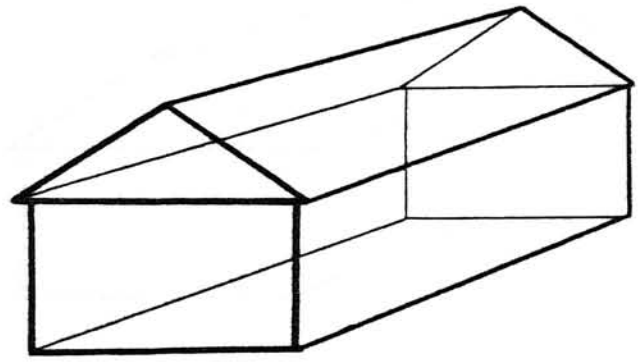


Fig. 1.15 – Profundidade modulada em intensidade.

Neste caso, a dificuldade de interpretação dos resultados visualizados acompanha a complexidade da cena em questão.

I. 4.2.4 — **Imagens estereoscópicas**

A impressão de profundidade pode igualmente ser obtida através da criação de duas imagens estereoscópicas, uma para cada olho.

Raras pessoas conseguem sintetizar estas duas imagens, a maior parte exige um par de óculos especiais.

I. 4.2.5 — **Eliminação das linhas escondidas**

Esta técnica considera opacos todos os objectos que entram na composição de uma cena. Assim, as linhas, ou partes das linhas, escondidas por qualquer face são eliminadas [fig. 1.16].

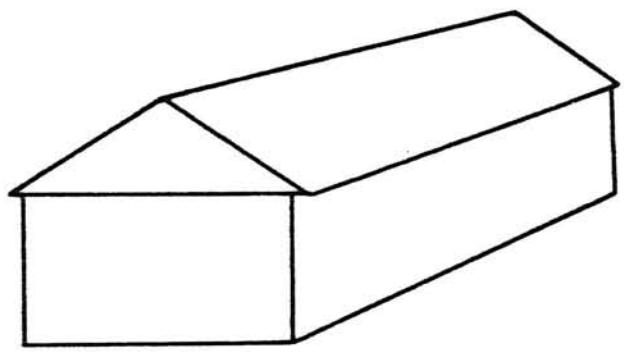


Fig. 1.16 – Impressão de profundidade originada pela eliminação das linhas escondidas.

Desta maneira, é possível obter uma facilidade notável na interpretação dos resultados visualizados, paga em tempo de CPU, por um custo que está longe, a maior parte das vezes, de ser insignificante.

I. 4.2.6 — **Eliminação das superfícies escondidas**

Com esta técnica pretende-se associar a cada objecto um ou vários materiais, caracterizados por cores, texturas, transparência, ... Assim, ao cálculo das partes visíveis de uma cena é preciso acrescentar a possibilidade de as podermos "pintar", de acordo com os materiais que lhes foram atribuídos.

Esta técnica, que na ausência de um processador gráfico poderoso exige tempos de cálculo extremamente longos, pode tomar em consideração um ou mais fontes luminosas e simular o aspecto de vários tipos de materiais; só com uma certa dificuldade se distinguem as imagens assim sintetizadas de imagens do mundo real [fig. 1.3].

I. 4.3 — **A determinação das superfícies visíveis**

Esta técnica é, entre as indicadas, aquela que maior interesse apresenta, devido ao elevado nível de realismo das imagens que dela resultam.

Nesta fase, pretende-se determinar os objectos (ou partes, de objectos) visíveis de uma cena, representada pelo seu *Modelo de Descrição,* constituindo-se o *Modelo de Visibilidade* respectivo [fig. 1.17].



Fig. 1.17 – O estudo de visibilidade.

Esta operação, indispensável na síntese de imagens realistas, tem uma importância fundamental em investigação gráfica, facto que se justifica pelos numerosos trabalhos já publicados.

A diversidade dos algoritmos publicados é devida principalmente aos diferentes modelos de descrição e de visibilidade que manipulam e ao espaço de cálculo que utilizam.

Assim, por exemplo, distinguem-se os algoritmos que trabalham no *espaço-objecto* e que calculam as partes visíveis com a precisão da vírgula flutuante disponível no computador em causa; outros trabalham no *espaço-imagem,* com uma precisão que toma em linha de conta a resolução dos dispositivos de visualização.

Sem pretendermos pormenorizar este tema, cuja análise sistemática encontra-se disponível em vários trabalhos publicados [SSS 74], [Luc 79], [Bou 80], indicamos o princípio dos quatro algoritmos mais divulgados. Tentamos, desta maneira, analisar para cada caso a viabilidade de uma realização de natureza "hardware".

Os quatro algoritmos escolhidos são:

— Algoritmo que utiliza uma memória de profundidade (Z-buffer) [Cat 75]

— Algoritmo de Atherton e Weiler [AtW 78]

— Algoritmo de Newell, Newell e Sancha [NNS 72]

— Algoritmo de Watkins [Wat 70]

Acrescente-se que os problemas relacionados com a *transparência* e *sombras,* considerados geralmente nestes algoritmos, são analisados posteriormente, numa altura julgada mais adequada.

I. 4.3.1 — **Algoritmo que utiliza uma memória de profundidade (Z-buffer)**

Este algoritmo [Cat 75] realiza os seus cálculos no *espaço-imagem* e retém, para cada ponto (x, y) do ecran, a profundidade (x, y) e o código da cor (x, y). O seu princípio de funcionamento é relativamente simples:

— a profundidade (x, y) e a cor (x, y) são inicializadas, respectivamente, com uma profundidade máxima e a cor de um plano de fundo.

— Determina-se, para cada polígono que entra na composição dos objectos da cena, todos os pontos (x_p, y_p) no interior da sua projecção sobre o plano do ecran.

Para todos estes pontos:

- calcula-se a profundidade do polígono em (x_p, y_p)
- se a profundidade calculada é inferior a profundidade (x_p, y_p), este ponto está mais próximo do observador que os precedentes; neste caso, o valor profundidade (x_p, y_p) é substituído pelo valor da profundidade calculada e cor (x_p, y_p) toma o código da nova cor, de acordo com a iluminação do polígono em estudo [fig. 1.18].

  Mas se, pelo contrário, a profundidade calculada é superior a profundidade (x_p, y_p), conservam-se os valores anteriores, pois algum(s) dos pontos precedentes está mais próximo do observador que o actual.

Logo que todos os polígonos tenham sido processados, cor (x, y) conterá o modelo de visibilidade desejado, neste caso, a imagem pronta a visualizar.

Apesar das dificuldades que resultam da utilização de duas matrizes de dimensões enormes (para uma resolução 500 × 500 cada matriz envolve 250 000 posições de memória), a simplicidade deste algoritmo inspirou vários trabalhos publicados, baseados em realizações programadas [Cat 75] e realizações micro-programadas [CHP 79].

A realização "hardware" deste algoritmo não envolve problemas de maior, se considerarmos o desenvolvimento tecnológico contínuo das memórias MOS.

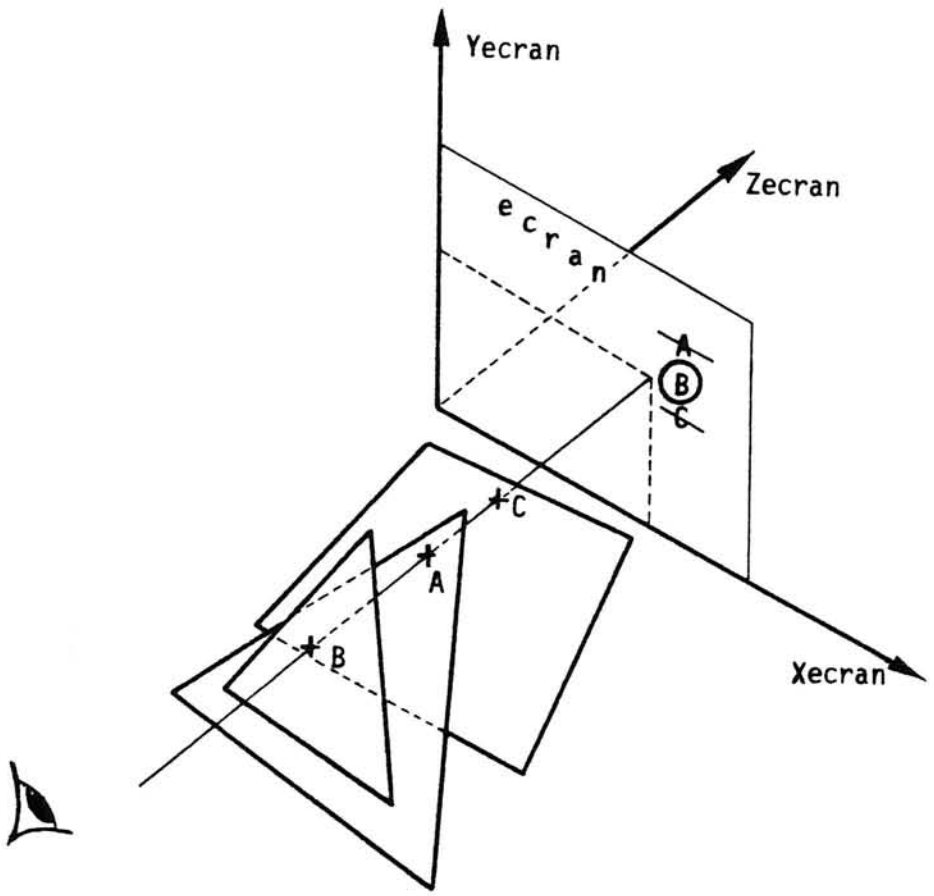


Fig. 1.18 – O princípio do algoritmo "Z – buffer".

I. 4.3.2 — **Algoritmo de Atherton e Weiler**

Este algoritmo [AtW 78], que trabalha no *espaço-objecto,* produz, como modelo de visibilidade, um conjunto de faces poligonais visíveis [fig. 1.19].

Quando se pormenoriza o seu princípio de funcionamento, encontramos como operações principais [Bou 80] duas ordenações (sort) de faces e um corte (corte de uma face pelo plano de uma outra), o que torna pouco viável uma realização totalmente "hardware", no estado actual da micro-electrónica. Contudo, uma solução híbrida ("hardware" e "bit slices") parece-nos uma pista com algumas hipóteses de realização.

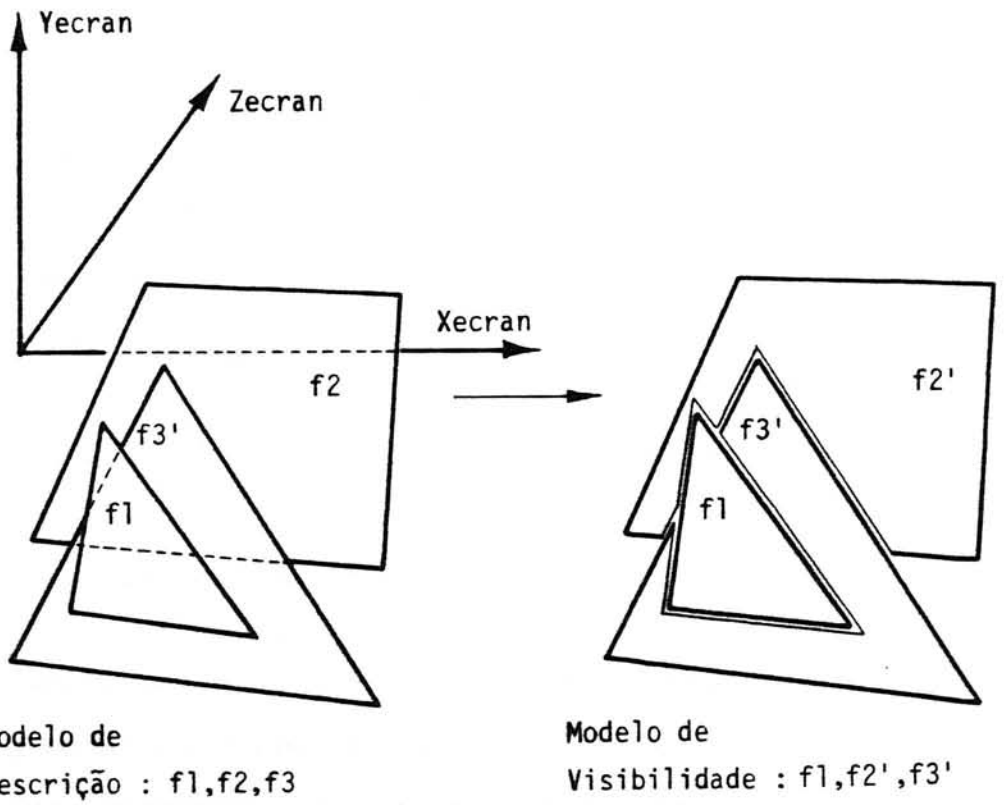


Fig. 1.19 – O princípio do algoritmo de Atherton e Weiler.

I. 4.3.3 — **Algoritmo de Newell, Newell e Sancha**

Este algoritmo [NNS 72], que trabalha igualmente no *espaço-objecto,* produz, como modelo de visibilidade, uma lista de polígonos ordenada segundo o valor decrescente da profundidade máxima (em Z) de cada um deles [fig. 1.20].

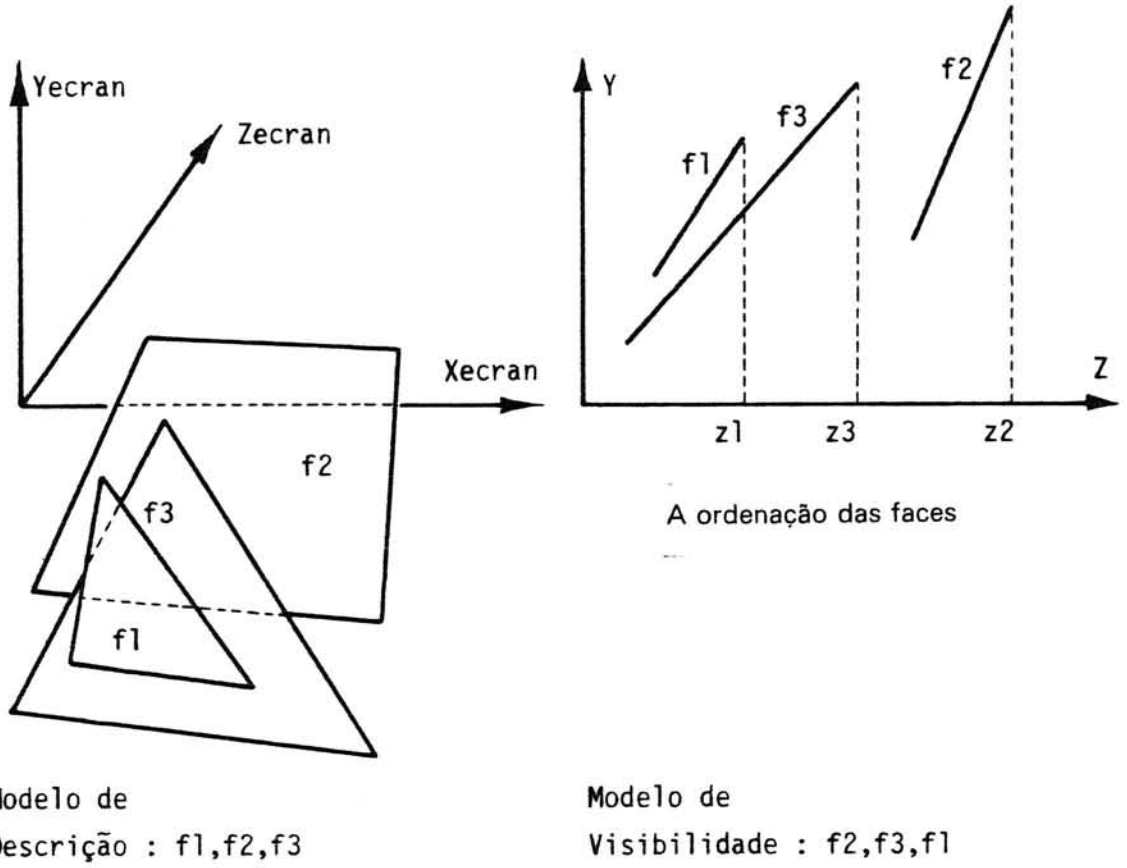


Fig. 1.20 – O princípio do algoritmo de Newell, Newell e Sancha.

Durante a ordenação das faces, a existência de duas delas, que se escondam mutuamente, faz com que uma seja cortada em função do plano da outra.

O grande inconveniente que apresenta o modelo de visibilidade gerado reside na sua dimensão; com efeito, se houve cortes durante o processo, a lista à saída é ainda mais longa que a lista à entrada.

Algumas vantagens importantes, que serão postas em realce mais tarde, são justificadas pelo facto de ser ainda disponível no modelo de visibilidade, toda a informação existente ao nível do modelo de descrição.

As considerações sobrè uma possível realização de natureza "hardware" deste algoritmo são muito idênticas às indicadas para o caso anterior.

I. 4.3.4 — **Algoritmo de Watkins**

Este algoritmo trabalha no *espaço-imagem* e gera um conjunto de segmentos horizontais, como modelo de visibilidade.

A cena a estudar é "varrida" linha por linha, de alto a baixo e, em cada linha, da esquerda para a direita.

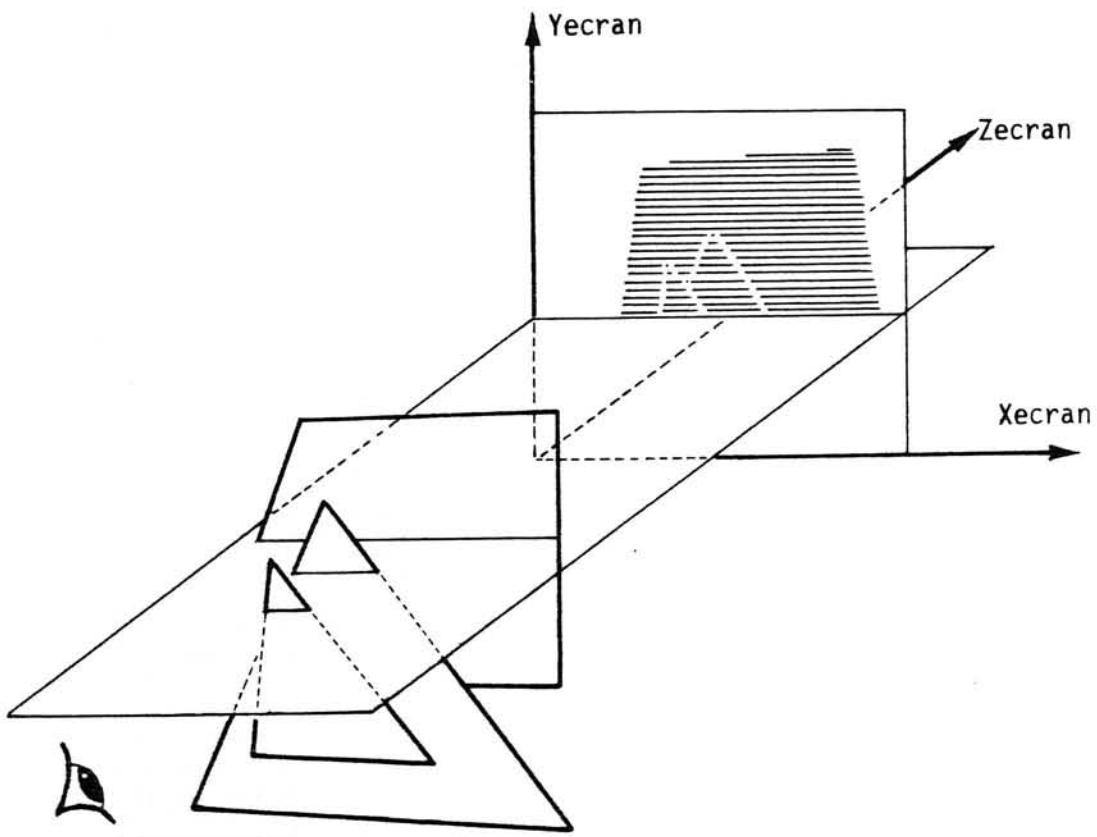


Fig. 1.21 – O princípio do algoritmo de Watkins.

Ao nível de cada linha, efectua-se o estudo de visibilidade entre os segmentos que resultam da intersecção de um plano perpendicular ao eixo OY [fig. 1.21] com todos os objectos presentes na cena. Justifica-se, desta maneira, a composição do modelo de visibilidade, conjunto de segmentos horizontais, aos quais se pode associar uma cor ou um número que identifique um objecto ou uma face.

Aqueles segmentos são ordenados de acordo com o varrimento do tipo TV, característica que tem inspirado muitas outras versões que propõem melhoramentos à versão original [Bou 80].

Uma realização de natureza "hardware" deste algoritmo foi anunciado em 1973 [NeS 73], da qual conhecemos algumas características através de um artigo de 1975 [EaS 75]: — Protótipo desenvolvido por Evans and Sutherland Computer Corp., que produz imagens a preto e branco com uma resolução 256 × 256; resoluções mais elevadas são previstas, mas o tempo de resposta vem extremamente mais longo. A versão comercializada exibe um custo extremamente elevado.

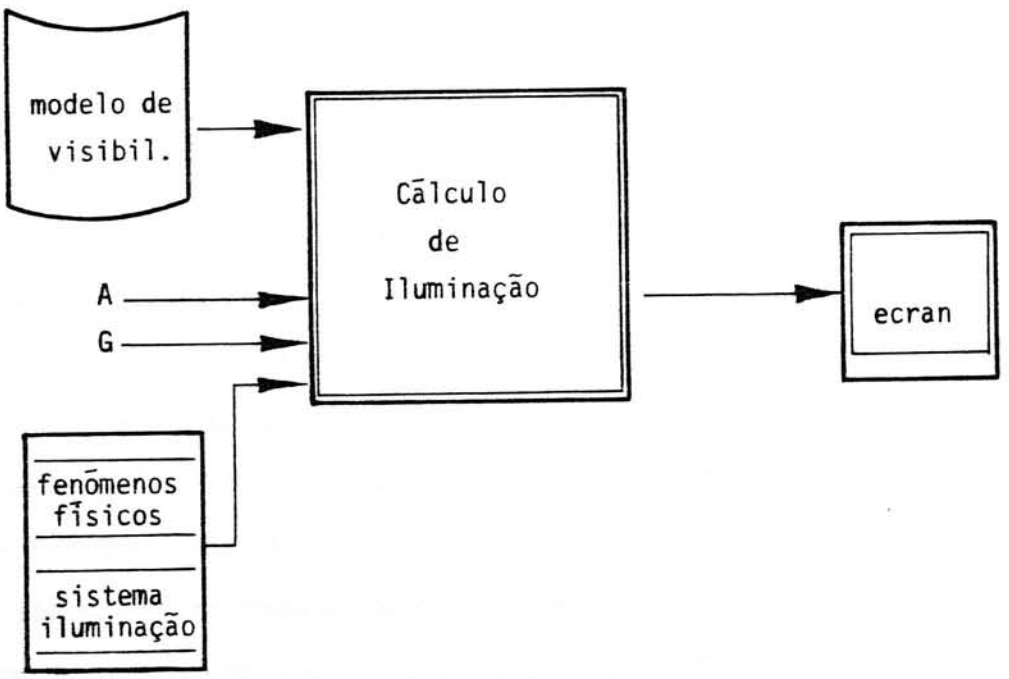


Fig. 1.22 – O cálculo de iluminação.

I. 4.4 — **O cálculo de iluminação**

Ao pretender-se uma apresentação visual realista das informações retidas pelo Modelo de Visibilidade, deve-se entrar em linha de conta com os efeitos produzidos pela iluminação da cena. Os resultados assim obtidos, que simulam imagens do mundo real, são possíveis graças à manipulação de modelos do sistema de iluminação da cena e dos principais fenómenos físicos postos em evidência pelos objectos iluminados [fig. 1.22].

I. 4.4.1 — **A modelização dos efeitos da iluminação**

Em geral, os raios luminosos chegam aos nossos olhos por intermédio dos objectos sujeitos a uma iluminação e é o comportamento destes objectos, em relação à luz que recebem, que determina o seu aspecto. Os modelos que descrevem este comportamento são relativamente complexos, devido ao grande número de fenómenos que integram.

O principal fenómeno a considerar é sem dúvida a *Reflexão,* que exprime a intensidade luminosa reflectida por uma superfície. A *Transparência* é também um fenómeno importante, traduzindo a intensidade luminosa que transita no interior de certos materiais.

No modelo que descreve o sistema de iluminação de uma cena podem ser considerados os dois componentes seguintes:

— a *intensidade luminosa difusa ou ambiente* que exprime uma iluminação uniforme em todas as direcções (efeito da atmosfera)

— as *fontes luminosas direccionais* que estão na origem das reflexões difusas, reflexos especulares e sombras, estas últimas traduzindo a ocultação mútua dos objectos.

Nos parágrafos seguintes consideram-se os modelos da iluminação de um ponto e alguns métodos para os aplicarem ao cálculo da iluminação de superfícies.

I. 4.4.2 — **O modelo de iluminação de um ponto**

O modelo da iluminação de um ponto P deve considerar a energia luminosa devida à *luz difusa* (Epd), à *luz direccional* de uma ou várias fontes luminosas (Eps) e à intensidade luminosa que lhe chega através de materiais transparentes (Ept) [NeS 79]; a iluminação total vem:

$$Ep = Epd + Eps + Ept$$

Pode-se modelizar a contribuição da luz difusa, cuja intensidade é Id, por:

$$Epd = Rp \times Id$$

Rp indica o coeficiente de reflexão no ponto em questão e varia entre 0 e 1.

Verifiquemos que um modelo que considerasse unicamente Epd originaria imagens pouco realistas; uma face que se deslocasse no espaço conservaria sempre a mesma iluminação. Esta constatação impõe que seja tomada em linha de conta a influência de fontes luminosas direccionais.

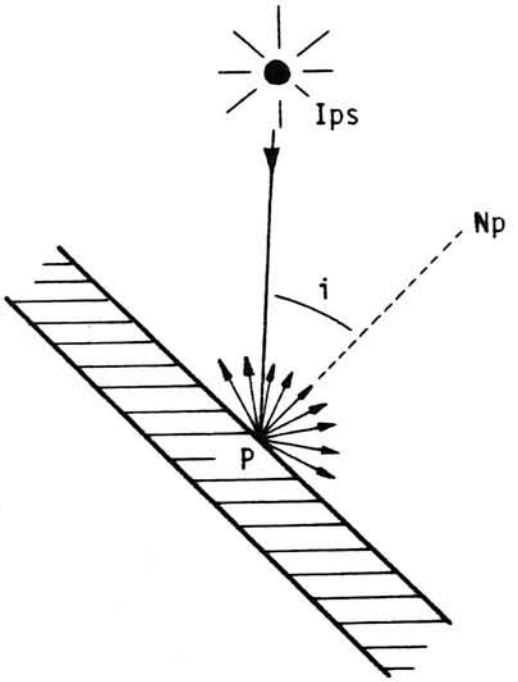


Fig. 1.23 – A reflexão difusa.

Analisemos três casos:

1 — Se o raio luminoso encontra uma superfície mate, reflete-se em todas as direcções sem excepção: trata-se de uma *reflexão difusa* [fig. 1.23].

Graças à lei de Lambert, que estabelece que a energia luminosa ao encontrar uma superfície reflete-se proporcionalmente ao cosseno do ângulo de incidência, i, pode-se modelizar a *reflexão difusa* através de

$$(Rp \times \cos i) \times Ips$$

sendo Ips a intensidade da fonte luminosa [fig. 1.23].

Acentua-se que, para $i > 90°$, a superfície pode considerar-se "escondida" em relação à fonte luminosa e Eps toma então o valor zero.

2 — Se o raio luminoso encontra uma superfície perfeitamente lisa, reflete-se segundo uma direcção privilegiada, definida por um ângulo de reflexão, r, igual ao ângulo de incidência; estamos agora em presença da *reflexão especular* [fig. 1.24].

Nesta situação a posição do observador é um parâmetro importante. Por exemplo, a reflexão especular é máxima se j = 0 e decresce rapidamente quando j aumentar [fig. 1.24].

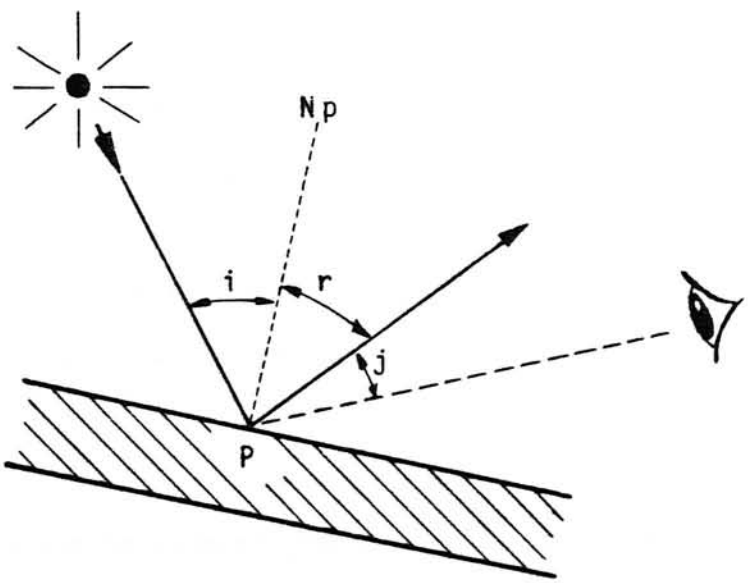


Fig. 1.24 – A reflexão especular.

A reflexão especular é ainda função de um *coeficiente de reflexão especular* W(i) [fig. 1.25] e pode ser modelizada por:

$$W(i) \times ((\cos j) \times\times n) \times Ips$$

onde n determina o *brilho* característico do material em causa. Por exemplo, uma superfície metálica, que produz reflexos muito pequenos em superfície, terá um valor de n elevado (n varia normalmente entre 1 e 10).

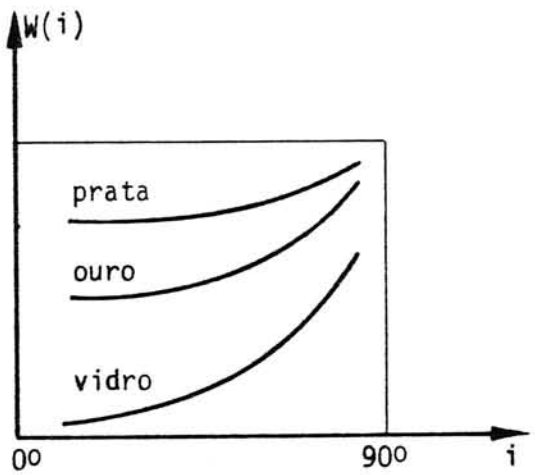


Fig. 1.25 – O coeficiente de reflexão especular.

3 — No caso mais frequente, em que uma superfície não é perfeitamente lisa nem perfeitamente mate, ter-se-á ao mesmo tempo uma reflexão difusa e uma reflexão especular. O modelo vem, então:

$$Eps = (Rp \times \cos i + W(i) \times ((\cos j) \times\times n)) \times Ips$$

Finalmente, a contribuição da energia que chega ao ponto P por *transparência,* pode ser descrita por:

$$Ept = Tp \times Epb$$

sendo Tp o *coeficiente de transmissão* ou *transmitância* que varia entre 0 e 1 e Epb a energia luminosa que chegaria a P pelo interior do material em causa. A energia Epb é devida, em geral, a reflexões originadas em superfícies mais afastadas do observador que o ponto P [fig. 1.26].

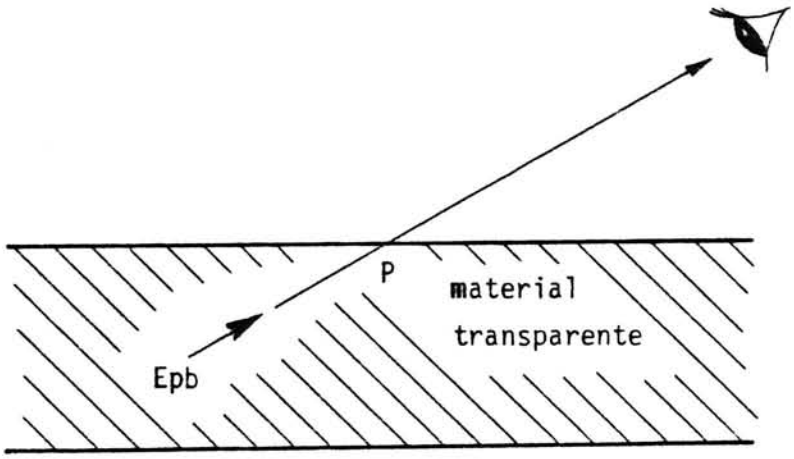


Fig. 1.26 – Um material transparente

O modelo descrito, traduzido por Ep, está na base de resultados de muito boa qualidade, uma vez que toma em consideração os principais fenómenos físicos postos em evidência por materiais expostos a uma iluminação.

Até aos nossos dias, a incidência do "hardware" neste domínio reduz-se a um número muito pequeno de realizações conhecidas.

Um destes casos [Sta 78], utilizado na visualização de moléculas orgânicas, limita a composição das cenas a conjuntos de esferas. O cálculo da iluminação, sempre do mesmo tipo, é assim guardado numa tabela [cf. II. 4.3.1], permitindo a obtenção de imagens em tempo real.

Um outro caso [Ler 80], mais geral que o anterior, executa os cálculos graças a uma realização híbrida (microprogramação e "hardware"). Neste caso, a limitação imposta reside no número de faces visualizáveis em tempo real [cf. II. 4.3.3].

Este número restrito de realizações conhecidas é sem dúvida o reflexo das dificuldades em jogo:

— tempo disponível e complexidade dos cálculos.

O grande problema consiste em encontrar modelos que exibam parâmetros suficientemente independentes, permitindo uma decomposição em problemas mais "simples"; desta maneira, torna-se viável a utilização de tabelas pré-calculadas...

Esta técnica foi considerada na concepção de HELIOS, como teremos oportunidade de mostrar no decorrer dos capítulos III e IV.

I. 4.4.3 — **Modelos de iluminação de superfícies**

A quantidade de factores a considerar no cálculo da iluminação torna praticamente inviável esta manipulação para todos os pontos de cada um dos objectos presentes numa cena. Os dois métodos mais usuais para atenuar este inconveniente são:

— considerar um sub-conjunto entre os factores mais importantes. Este método reduz drasticamente o nível de realismo das imagens,

— considerar o cálculo de alguns pontos "chave" para, a partir deles, deduzir a iluminação em cada ponto.

Vejamos algumas técnicas que se agrupam neste último caso:

1 — *Aproximação Uniforme* [Bok 70], que afecta a iluminação do ponto central da face a todos os outros pontos.

2 — *Interpolação linear da intensidade* [Gou 71], onde se calcula apenas a iluminação nos vértices de um quadrilátero (I_1, I_2, I_3 e I_4); nos outros pontos a iluminação é obtida por uma dupla interpolação linear [fig. 1.27].

A iluminação sobre o segmento AB é calculada por uma interpolação linear entre Ia e Ib, estes últimos calculados por uma primeira interpolação linear entre I_1, I_2 e I_2, I_3 [fig. 1.27].

Esta técnica suprime as descontinuidades luminosas características da "aproximação linear"; todavia, torna praticamente impossível o cálculo de reflexos.

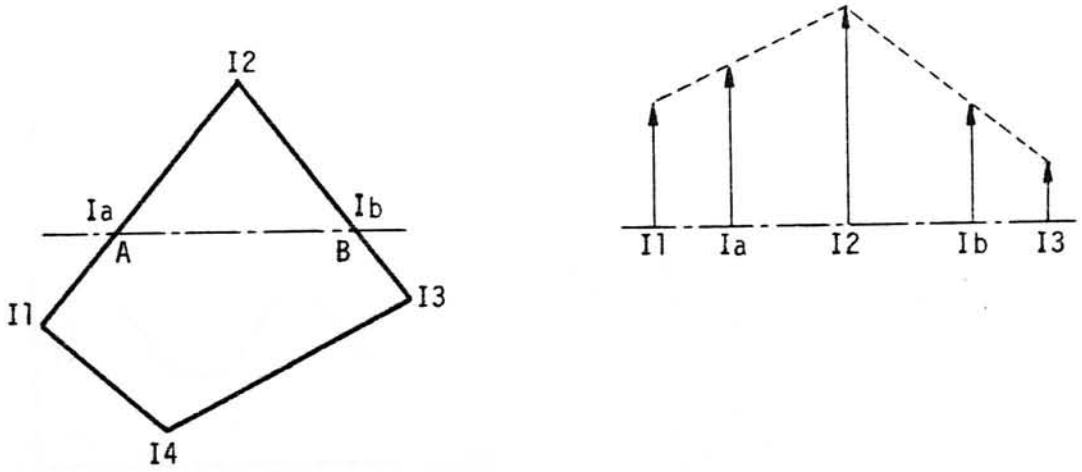


Fig. 1.27 – A interpolação linear da iluminação.

3 — *Interpolação linear da normal* [Pho 75].

Neste caso a interpolação não se faz directamente sobre a iluminação, mas sobre a normal à superfície, isto é, sobre os seus três componentes.

O tempo exigido é pois manifestamente superior aos anteriores, agravado pela necessidade de um cálculo da iluminação em cada ponto da superfície. Esta técnica permite a apresentação de reflexos muito realistas.

Podemos ainda analisar a possibilidade de modelizar uma "textura de iluminação".

A percepção de relevo de uma superfície realiza-se em geral graças às variações de iluminação que provoca. Para obter este efeito seria suficiente "perturbar" convenientemente a normal da superfície em causa. Assim, uma "perturbação aleatória" permitiria a visualização de superfícies rugosas (por exemplo, como a casca de laranja), enquanto que uma "perturbação regular" visualizaria uma superfície com motivos geométricos (por exemplo, um muro de tijolos, quadrados de chocolate, ...).

A literatura não se apresenta suficientemente pródiga neste domínio, pois conhecemos um único trabalho publicado [Bli 78]. O modelo que J. BLINN utiliza representa uma função que exprime a "altitude" do relevo a acrescentar a uma superfície regular inicial [fig. 1.28]. A partir do relevo resultante calcular-se-ia a nova normal em cada ponto.

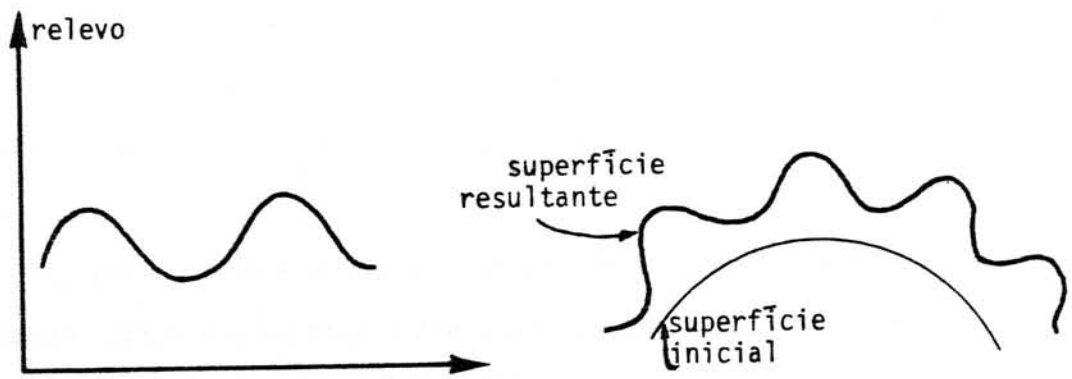


Fig. 1.28 – A modelização do relevo.

O principal inconveniente deste método reside no facto da "perturbação" criada não ser independente do tamanho dos objectos. BLINN propõe igualmente uma solução que permite suprimir este problema; a função modelo deixa de ser interpretada como a "altitude" do relevo, mas como a expressão de um ângulo de rotação a aplicar ao vector normal.

I. 4.5 — **Outros factores a considerar na síntese de imagens realistas**

A utilização de imagens que exibam um elevado nível de realismo impõe que sejam tomados em linha de conta diversos factores, nomeadamente a cor, as texturas coloridas, a transparência e as sombras, os quais são considerados nos parágrafos seguintes.

I. 4.5.1 — **A cor**

Um material que "reage" em função do comprimento de onda dos raios luminosos que o atingem, provoca a sensação de cor num eventual observador. Além disso, se este fenómeno obedece a uma certa distribuição, aquele material exibe uma *textura colorida.*

Quando se aprofunda este tema, rapidamente se verifica que um grande número de questões continuam ainda sem uma resposta satisfatória. Citemos alguns exemplos:

1 — Os monitores a cores produzem a sensação de cor, através da síntese aditiva das três cores fundamentais, vermelho, verde e azul; trata-se do modelo RGB (red, green, blue) [cf. II. 2.2].

De acordo com este modelo uma cor qualquer, C, pode ser representada por três coeficientes, r, g e b, que exprimem a intensidade das três cores fundamentais, fazendo:

$$m(C) = r(R) + g(G) + b(B)$$

onde m indica a intensidade da cor C.

Infelizmente, esta maneira de descrever uma cor é muito pouco natural para o utilizador, o que justifica uma certa atenção da investigação actual na procura de modelos mais intuitivos [GiJ 87], [Smi 78].

Contudo, a criação de cores não é a única característica importante de um modelo; a facilidade de cálculo numérico é igualmente fundamental.

Com efeito, a simulação de fenómenos, como por exemplo o fumo ou o nevoeiro, é mais ou menos difícil de realizar conforme o modelo de cor utilizado; isto justificaria a co-existência, num sistema de síntese de imagens realistas, de vários modelos de cores e dos correspondentes algoritmos de conversão para o modelo RGB, característico dos monitores.

2 — O aspecto pouco realista das imagens constituídas unicamente por cores uniformes limita consideravelmente o seu campo de aplicação. De facto, a visualização realista de objectos exige quase sempre a manipulação de texturas coloridas que caracterizem os materiais em jogo.

Alguns trabalhos publicam técnicas para a resolução desta questão (ver bibliografia em [Mar 79]), as quais apresentam em geral duas fases:

— síntese de um modelo de textura, obtido a partir de uma descrição numérica ou algorítmica.

— "preenchimento" de uma superfície com uma textura, fase que põe em evidência vários tipos de problemas. Estes podem aparecer durante o "preenchimento" da superfície (por exemplo, a não concordância das fronteiras) ou durante as transformações a que o modelo é sujeito com o intuito de satisfazer a uma perspectiva ou à forma dos objectos.

Actualmente não conhecemos nenhum sistema que permita, por via "hardware", a memorização e o tratamento de tais modelos de texturas, conjuntamente com um modelo de iluminação. O terminal HELIOS é uma primeira abordagem importante a este problema.

I. 4.5.2 — **A modelização da transparência**

A modelização da transparência encontra-se muitas vezes associada aos modelos de visibilidade, mas os algoritmos de eliminação das partes escondidas que manipulam objectos transparentes são pouco numerosos.

Entre os algoritmos estudados, o de Newell, Newell e Sancha [NNS 72] é de longe o que melhor se harmoniza com este problema. De facto, ao nível do seu modelo de visibilidade, as faces encontram-se já ordenadas, da mais distante à mais próxima.

A intensidade luminosa correspondente a uma face opaca vem pura e simplesmente substituir as intensidades anteriormente calculadas.

Se a face encontrada é transparente ou semi-transparente, não haverá uma substituição pura e simples, mas uma combinação de intensidades.

Para contemplarem a transparência, os restantes algoritmos exigem alguns tratamentos especiais. Assim, por exemplo:

— o *algoritmo "Z-buffer"* [Cat 75] requer uma ordenação das faces, o que lhe retira o seu principal interesse — a simplicidade —.

— o *algoritmo de Atherton e Weiler* [AtW 78] requer um segundo estudo de visibilidade que englobe os elementos "escondidos" pela face transparente.

— o *algoritmo de Watkins* [Wat 70] requer, por seu turno, modificações relativamente complexas; algumas soluções foram já publicadas [Cro 77], [Mye 75].

A *transmitância* [cf. I. 4.4.2] é tomada em linha de conta por certos algoritmos, nomeadamente Newell [NNS 72] e Myers [Mye 75], que associam a cada face um coeficiente que exprime o *factor de reflexão,* r ($0 <= r <= 1$); nestes casos a transmitância é suposta igual a $1 - r$, uma vez que a energia absorvida pelo material não é considerada.

O modelo proposto por Crow [Cro 77] não despreza o fenómeno da absorção, fazendo variar de maneira adequada o ângulo de incidência no cálculo da transmitância.

Neste momento, impõe-se duas conclusões:

— o número de respostas aos problemas suscitados pela transparência é extremamente limitado.

— a literatura especializada não aponta nenhuma pista que contemple os fenómenos da refracção e da transparência difusa, sob uma forma simples e, muito menos, sob a forma de texturas.

I. 4.5.3 — **As sombras**

As sombras são também um elemento importante na visualização de imagens realistas. Este fenómeno, posto em evidência pela introdução de fontes luminosas direccionais, é muitas vezes desprezado, considerando, por motivos de simplificação, o observador e a fonte luminosa confundidos num mesmo ponto.

As diferentes técnicas conhecidas que consideram este problema, associadas geralmente aos algoritmos de visibilidade, podem ser agrupadas em três categorias [Cro 77]:

— integração da sombra no cálculo de visibilidade [Bok 70].

Neste caso as superfícies na sombra são detectadas durante o varrimento horizontal da cena.

— sombras determinadas por um duplo estudo de visibilidade.

A ideia de base resume-se a utilizar um algoritmo de visibilidade para determinar as partes "visíveis" a partir da fonte luminosa, isto é, aquelas que não se encontram à sombra [fig. 1.29]. O modelo de "visibilidade" assim obtido constituirá o modelo de descrição do cálculo de visibilidade efectivo.

— sombras determinadas pela introdução de "volumes de sombras" [Cro 77].

Associa-se a cada face o seu "volume de sombra" gerado no espaço da cena. As "faces" destes volumes são consideradas de maneira adequada pelo algoritmo de visibilidade.

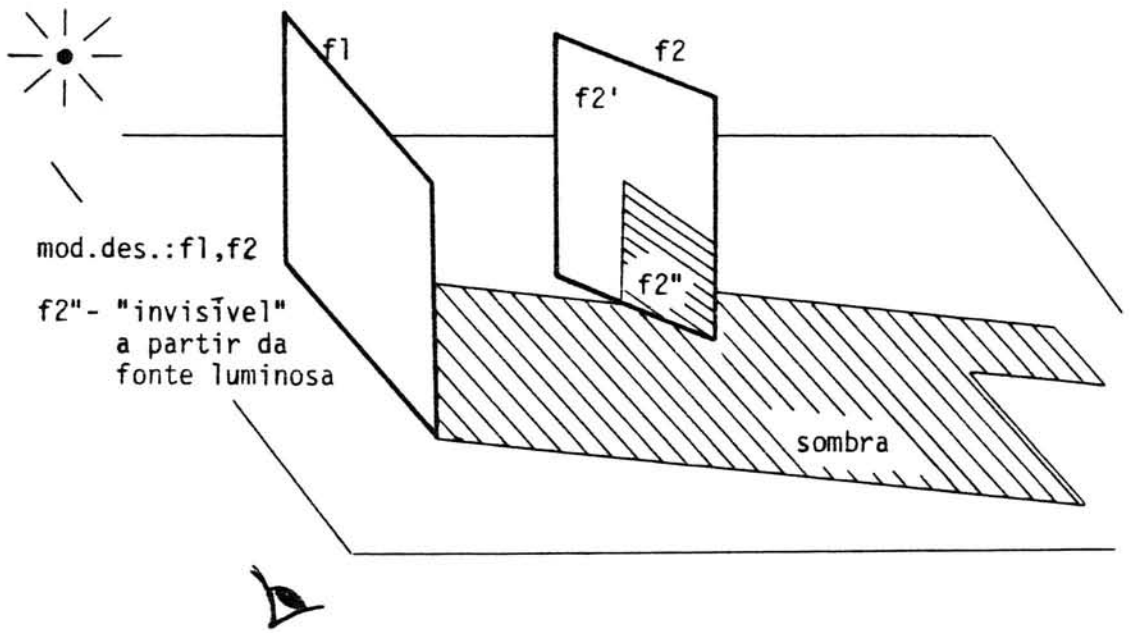


Fig. 1.29 – As sombras.

Das três categorias de técnicas apresentadas, a segunda engloba o maior número de trabalhos publicados (ver bibliografia em [Mar 79]).

Acentua-se que esta maneira de tratar as sombras exige que os algoritmos de visibilidade forneçam um modelo do mesmo tipo do modelo de descrição; entre os algoritmos estudados, somente o de Atherton [AtW 78] e o de Newell [NNS 72] contemplam esta condição, exigindo este último cuidados especiais para que o problema das sombras possa ser resolvido.

Uma última nota refere-se às sombras projectadas sobre superfícies não planas; o tratamento desta informação torna-se agora muito complexo, devido ao cálculo de volumes no espaço.

Alguns métodos são conhecidos para a determinação de sombras tanto em faces planas como em faces não planas [Wil 78], [Cat 74].

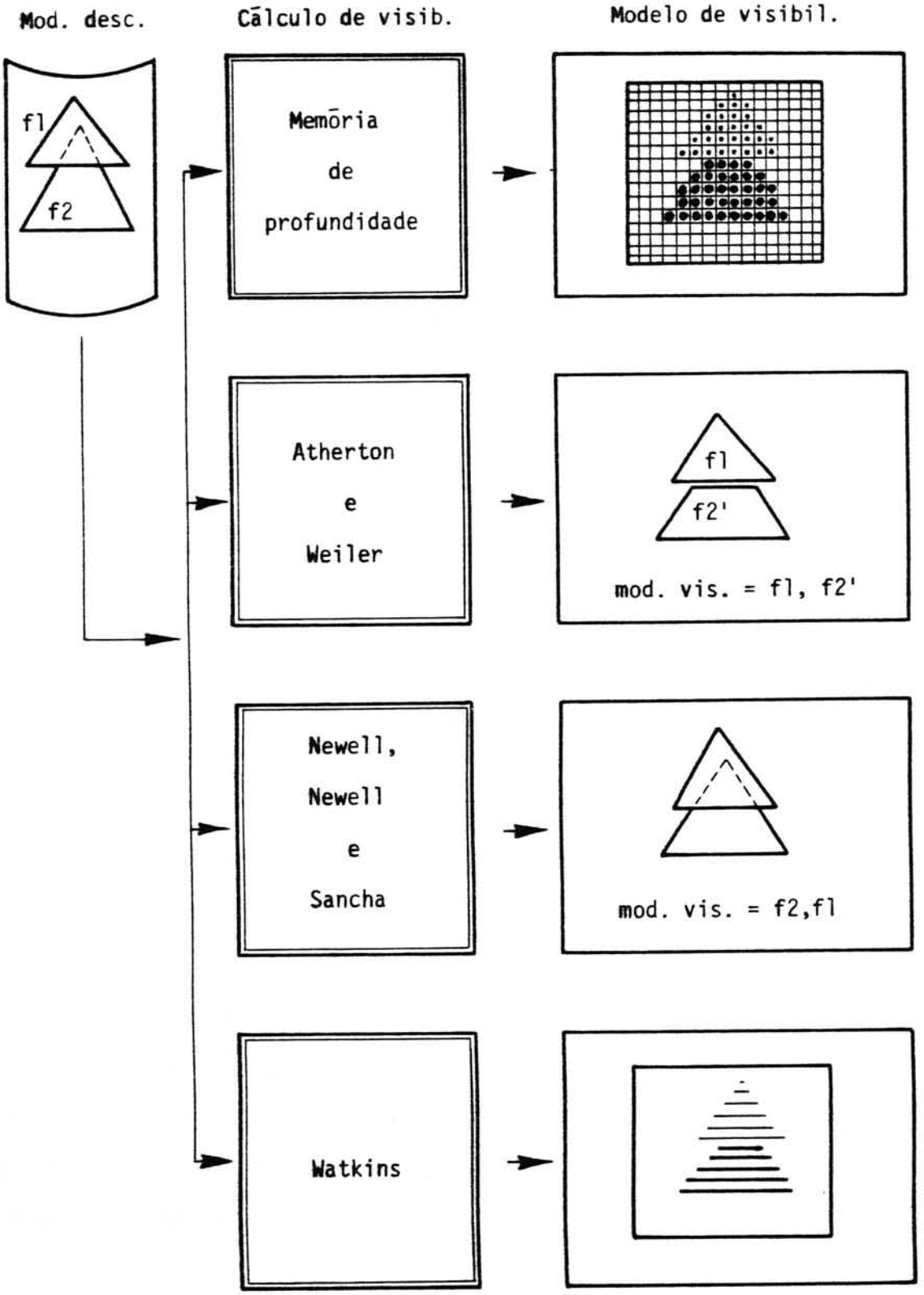


Fig. 1.30 – Os modelos de visibilidade dos algoritmos estudados.

I. 4.6 — **Síntese final**

Em geral esta operação encontra-se mais ou menos distribuída pelas fases precedentes, em particular, no cálculo de visibilidade.

A figura 1.30 resume os modelos de visibilidade dos algoritmos estudados, aplicados a um modelo de descrição comum — faces planas —.

A missão da síntese final está estreitamente ligada à natureza do modelo de visibilidade produzido:

— para o "Z-buffer" a imagem encontra-se totalmente calculada.

Esta solução caracteriza-se por uma rigidez evidente, uma vez que o utilizador não tem nenhuma facilidade de acesso aos parâmetros de iluminação, aos ficheiros dos materiais, etc., qualquer modificação passa obrigatoriamente por uma reconstrução total da imagem.

Estamos perante um exemplo da chamada "solução clássica" [cf. I. 3.2.1].

— o modelo de visibilidade que corresponde ao algoritmo de Watkins permite associar um número de face a cada segmento horizontal e, a este número, podemos ligar, por exemplo, uma cor; a afectação de uma textura cria algumas dificuldades, pois um certo número de informações adicionais não estão presentes (por exemplo, um referencial 3D da face à qual pertence o segmento).

— no modelo de visibilidade de Atherton a noção de face é conservada, o que permite um certo número de tratamentos, entre os quais a textura e a iluminação.

Surge agora uma nova atribuição da síntese:

- o preenchimento das faces

Esta operação toma como dados os vértices que definen cada face e determina o conjunto de pontos que lhe são interiores.

Acentua-se que a transparência não pode ser considerada, salvo tratamentos especiais anteriores ao modelo de visibilidade. Pelo contrário, as sombras podem ser calculadas por um segundo estudo de "visibilidade", considerando agora o observador e a fonte luminosa situados sobre o mesmo ponto.

— finalmente, o modelo de visibilidade que tem origem no algoritmo de Newell fornece-nos toda a informação disponível ao nível do modelo de descrição, ordenada de maneira a facilitar a eliminação das partes escondidas.

Esta solução é, entre todas as estudadas, aquela que mais se adapta à ideia que preconizámos:

- Conservar a separação de todos os tipos de informação até ao "software" de Visualização e, na medida do possível, até ao processador gráfico.

Como no caso precedente, a síntese final deve efectuar o "preenchimento" das faces, operação onde uma realização do tipo "hardware" nos parece simultaneamente viável e importante.

I. 5. CONCLUSÃO

No decorrer deste capítulo tivemos a oportunidade de indicar os problemas mais importantes integrados na síntese interactiva de imagens realistas.

Uma primeira análise conduziu-nos aos problemas que se põem quando se pretende escolher uma arquitectura para um sistema gráfico interactivo, neste caso orientado para a síntese de imagens realistas.

As duas primeiras soluções estudadas apresentavam inconvenientes bem notáveis:

— a primeira, a solução dita "clássica", não aproveitava correctamente as facilidades de um processador gráfico evoluído.

— a segunda, classificada de "ideal", não é viável no estado tecnológico actual.

Assim, a solução escolhida, colocando-se entre as duas precedentes, é apresentada como uma das soluções de compromisso realizável.

Uma segunda análise pôs em evidência os problemas característicos das diferentes operações elementares da síntese de imagens realistas e as incidências possíveis do "hardware" sobre cada uma delas.

CAPÍTULO II

EVOLUÇÃO E TENDÊNCIAS DO HARDWARE GRÁFICO APLICADO À SÍNTESE DE IMAGENS REALISTAS

CAPÍTULO II: EVOLUÇÃO E TENDÊNCIAS DO HARDWARE GRÁFICO APLICADO À SÍNTESE DE IMAGENS REALISTAS

II. 1. INTRODUÇÃO

As primeiras tentativas para utilizar a imagem como meio de comunicação homem-máquina datam do início dos anos 50 e, desde então, a evolução do "hardware" associado aos sistemas gráficos seguiu de muito perto o progresso tecnológico dos dispositivos de visualização. Com efeito, as características destes têm tido uma influência preponderante na estrutura interna do processador gráfico.

A tendência actual, cujo objectivo principal se situa na produção de imagens com aspecto realista, aponta na direcção dos monitores a cores de varrimento do tipo TV. As características importantíssimas destes dispositivos de visualização não justificam, isoladamente, esta situação; de facto, o progresso incessante da micro-electrónica permitiu uma modificação importante na estrutura clássica dos sistemas gráficos, ao dar resposta a um certo número de problemas ligados a esta tecnologia particular.

Na concepção de HELIOS enveredou-se também por este caminho e o protótipo realizado apresenta um conjunto de originalidades, que o justificam em relação aos sistemas actualmente existentes.

Evocaremos sucessivamente neste capítulo:

— A evolução do "hardware" associado aos sistemas gráficos, com base num estudo dos dispositivos de visualização mais vulgarizados.

— As tendências actuais, que apostam na utilização do monitor a cores de varrimento do tipo TV.

— A análise de sistemas existentes, através de um estudo de alguns sistemas característicos.

— Uma apresentação geral de HELIOS.

II. 2. EVOLUÇÃO TECNOLÓGICA DOS DISPOSITIVOS DE VISUALIZAÇÃO

Desde muito cedo se reconheceu o valor da imagem como meio de comunicação homem-máquina; progressos importantes são realizados continuamente, suscitados principalmente pela evolução tecnológica da micro-electrónica e dos dispositivos de visualização.

Entre os dispositivos de visualização disponíveis actualmente, os monitores a cores de varrimento do tipo TV são de longe os mais indicados para aplicações que envolvam a visualização de imagens realistas, apesar das dificuldades que ainda levantam no que diz respeito à interactividade.

Assim, dois critérios podem dirigir o estudo dos dispositivos existentes:

- a possibilidade de interacção que oferecem
- a qualidade de imagens que exibem.

O critério da interactividade é suficiente para eliminar deste estudo todos os dispositivos que utilizam o papel como suporte, por exemplo, os traçadores de gráficos; a nossa atenção concentra-se pois nos dispositivos equipados de um ecran.

É nesta classe que vamos encontrar os *tubos catódicos de varrimento incremental* e de *varrimento do tipo TV*, os *tubos de memória*, os *ecrans a plasma* e *a laser*.

II. 2.1 — Os tubos catódicos de varrimento incremental

A primeira utilização de um tubo catódico como meio de comunicação com um computador situa-se por volta de 1950 [Eve 2]. Nesta época, o principal problema a resolver não residia na baixa capacidade de interacção que tais sistemas ofereciam nem no nível de qualidade extremamente pobre dos resultados que exibiam, mas simplesmente na dificuldade de manter estável a imagem sobre o ecran.

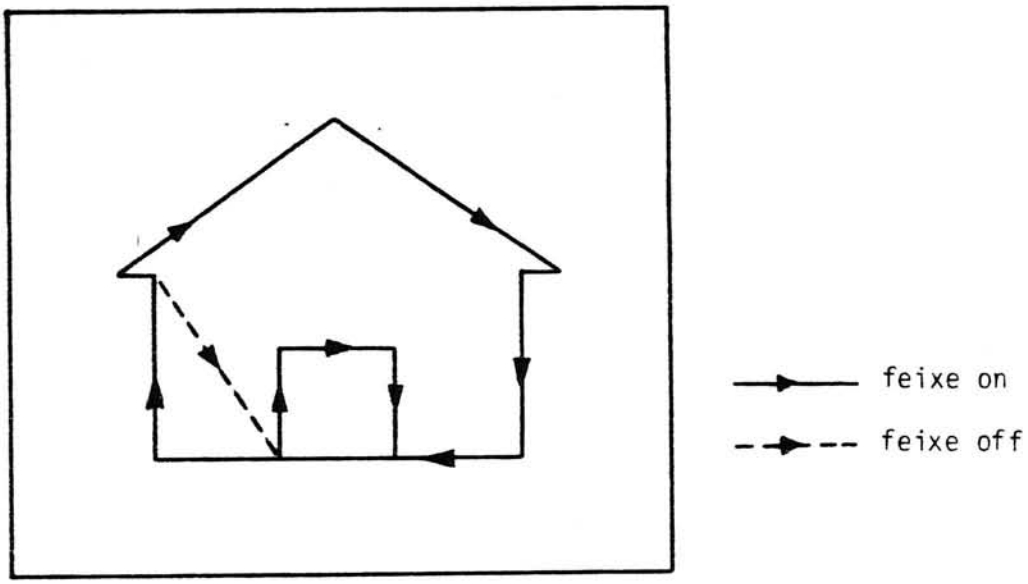


Fig. 2.1 – O varrimento do tipo incremental.

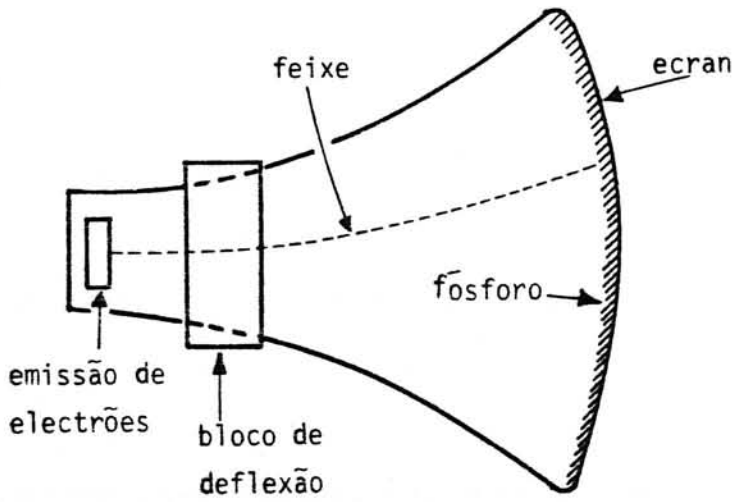


Fig. 2.2 – O princípio de um tubo catódico.

Com efeito, aquela imagem é devida ao fenómeno de *luminiscência* provocado pelo bombardeamento de um feixe de electrões, deslocado de maneira adequada [fig. 2.1] sobre a camada de fósforo que compõe o ecran [fig. 2.2]. Como a duração da luminiscência não é ilimitada, mas função da *persistência* do fósforo utilizado, o refrescamento da imagem torna-se indispensável [fig. 2.3].

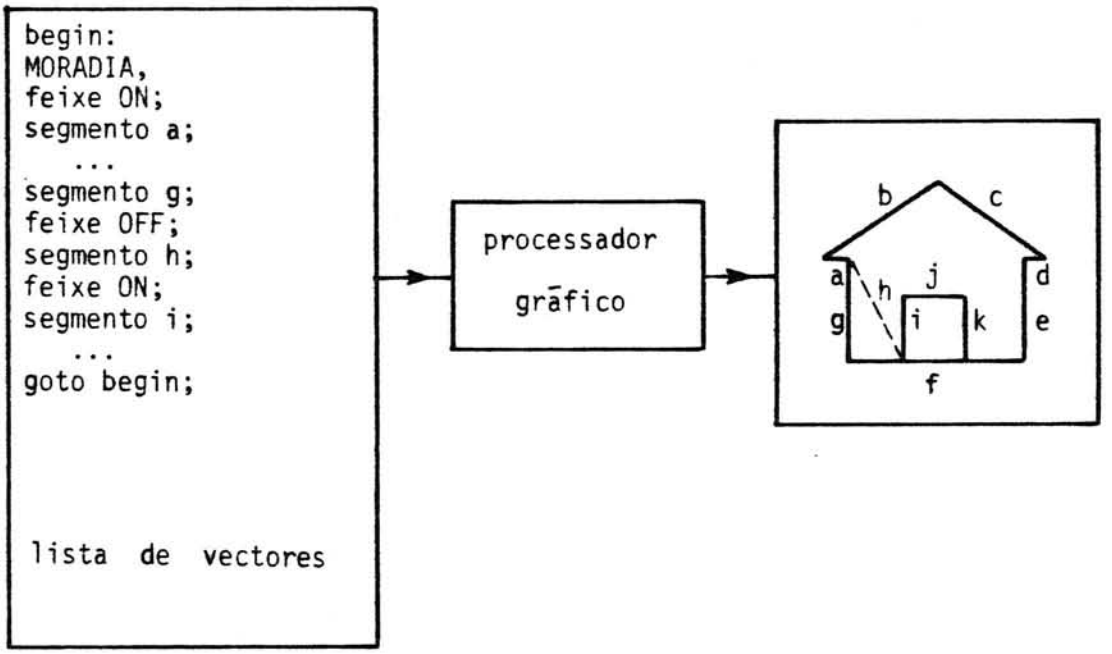


Fig. 2.3 – O refrescamento de um monitor de varrimento incremental.

Este tipo de varrimento, onde o feixe de electrões descreve um percurso idêntico à imagem, é designado por *varrimento incremental.*

Os monitores de varrimento incremental adaptam-se relativamente bem às aplicações que exijam apenas representações do tipo "desenho". Oferecem uma resolução elevada (4096 × 4096) e, além disso, o conteúdo da memória "lista de vectores" apresenta uma estruturação razoável [fig. 2.3], o que permite uma *designação directa* sobre o ecran e, logicamente, um nível de interacção importante.

Pelo contrário, o tempo necessário ao refrescamento da imagem impõe uma limitação ao número de segmentos visualizáveis. De facto, quanto maior for o número daqueles segmentos mais longo será o período de refrescamento e, consequentemente, mais nítida será a cintilação (flicker) da imagem sobre o ecran.

O número de cores disponíveis é também bastante reduzido, pois nos *tubos de penetração,* tecnologia muitas vezes associada com este tipo de varrimento, a cor é obtida controlando a profundidade de penetração do feixe de electrões num conjunto de camadas (em geral duas) de fósforos diferentes [fig. 2.4].

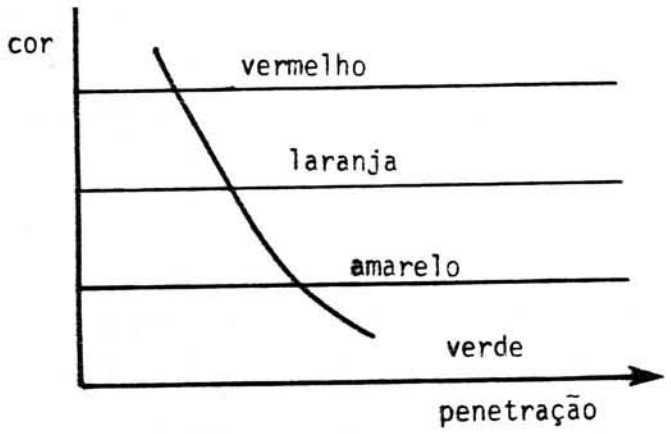


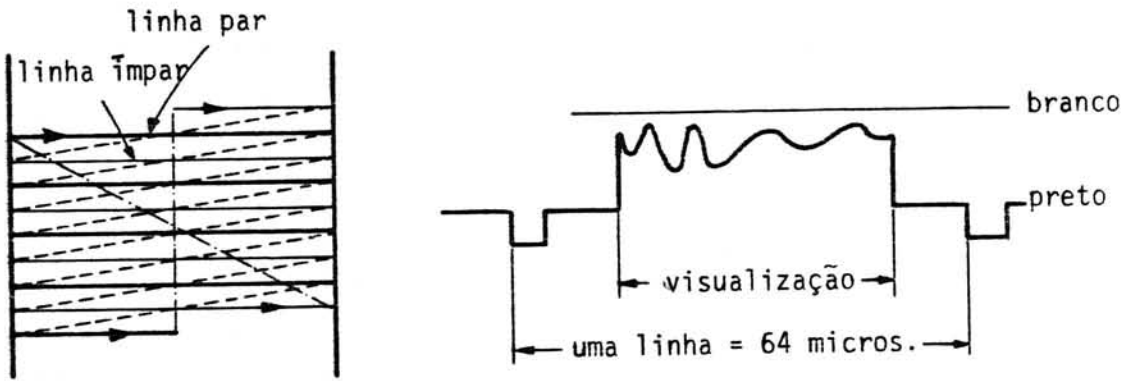
Fig. 2.4 – A obtenção de cores num tubo de penetração.

II. 2.2 — Os tubos catódicos de varrimento do tipo TV

Este tipo de varrimento é, actualmente, o mais divulgado, pois tem beneficiado de todo o progresso de que tem sido alvo o televisor doméstico.

Contrariamente ao que acontecia com o varrimento incremental, o percurso descrito pelo feixe de electrões não depende da imagem visualizada [fig. 2.5]; o deslocamento daquele feixe faz-se agora linha por linha, de alto a baixo e da esquerda para a direita. Esta operação repete-se regularmente um certo número de vezes por segundo (standards de 25 e 30 imagens/segundo), garantindo desta maneira o refrescamento.

Num varrimento deste tipo, uma imagem é em geral composta pelo conjunto das linhas pares, *quadro par,* e o conjunto das linhas ímpares, *quadro ímpar,* que são visualizados alternadamente [fig. 2.5].



standard europeu:

quadro ímpar = 312,5 linhas = 20 ms
\+ quadro par = 312,5 linhas = 20 ms
imagem = 625 linhas = 40 ms

Fig. 2.5 – O varrimento do tipo TV; algumas características do standard europeu.

A lista das principais vantagens deste varrimento é relativamente longa; entre outras, podemos salientar:

— imagem estável, independentemente da complexidade da imagem, uma vez que o percurso de varrimento é sempre o mesmo qualquer que seja a informação a visualizar.

— possibilidade de colorir os elementos da imagem (faces), pois que todos os pontos do ecran são "visitados" pelo feixe de electrões num período de 40 ms (no standard europeu de 50 quadros/s).

— baixo custo do televisor doméstico.

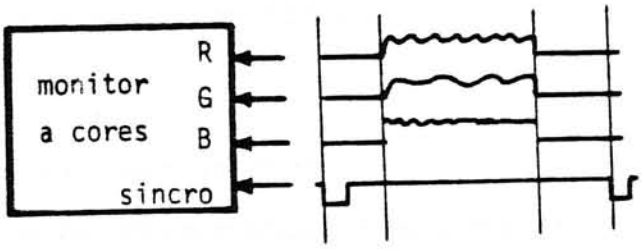
— facilidades para transmitir e guardar imagens (aparelhagem video).

— possibilidade de misturar uma imagem sintetizada com uma imagem real, esta última obtida a partir de uma câmara TV.

— várias cores ou níveis de cinzento podem ser utilizados.

Cores fundamentais			Cores obtidas
R	G	B	
0	0	0	preto
0	0	1	azul
0	1	0	verde
0	1	1	turquesa
1	0	0	vermelho
1	0	1	violeta
1	1	0	amarelo
1	1	1	branco

a – Formação de cores a partir do Vermelho, Verde e Azul.



b – monitor com as entradas R (red), G (green), B (blue) e SINCR independentes.

Fig. 2.6 – A formação das cores num monitor de varrimento do tipo TV.

O princípio da formação de cores é indicado na figura 2.6. Com efeito, quando se misturam as cores fundamentais (vermelho, verde e azul) em proporções adequadas, é possível obter uma cor qualquer.

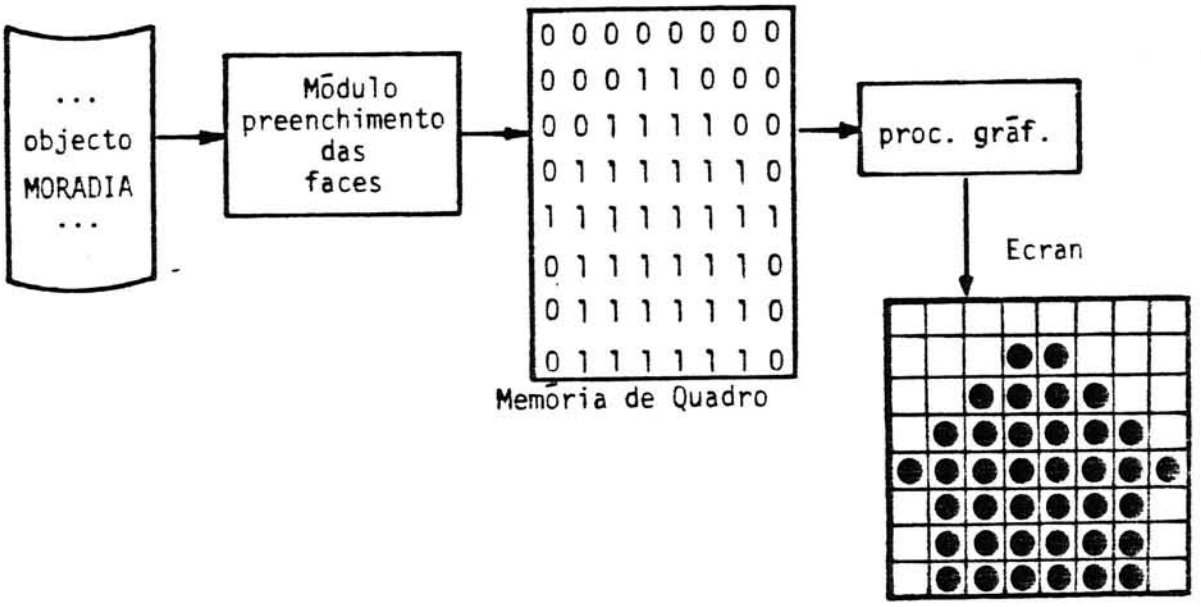


Fig. 2.7 – Estrutura clássica de um sistema que utiliza um monitor de varrimento TV.

Vejamos agora os principais inconvenientes deste varrimento:

— o refrescamento do tubo utiliza uma memória especializada, *memória de quadro,* onde a estruturação inicial da informação deixou de estar presente [fig. 2.7]. Assim, um módulo *"preenchimento de faces"* deverá existir, com o intuito de adaptar a informação a visualizar às características particulares deste tipo de varrimento.

— para uma resolução 512 × 512 a memória de quadro apresenta já uma capacidade importante.

— o nível de interacção é agora mais baixo, pois a designação de um elemento visualizado fornecerá apenas uma cor; trata-se de uma informação "anónima" que não evita uma pesquisa num ficheiro onde as informações de identificação estejam presentes.

— a capacidade da memória de quadro cria igualmente problemas quando se pretende visualizar uma nova imagem ou modificar uma grande parte da imagem presente. Em qualquer um destes casos, a quantidade de informação a considerar, passando pelo módulo "preenchimento de faces", torna-se importantíssima.

— baixa resolução em relação ao varrimento incremental.

Uma resolução 1024 × 1024 é possível, mas torna-se relativamente cara. Além disso, quando se abandona o standard do varrimento TV, perde-se igualmente algumas das duas vantagens.

II. 2.3 — **Os tubos de memória**

Do ponto de vista da escrita, o tubo de memória utiliza um princípio idêntico ao do varrimento incremental. A grande diferença, na qual reside o seu principal interesse e, paradoxalmente, o seu principal inconveniente, é o *refrescamento automático* da imagem, o que elimina a necessidade de uma memória especializada.

Estamos em presença de uma tecnologia cujas vantagens principais são:

— ausência de refrescamento externo, o que origina imagens estáveis, independentemente da quantidade de informação visualizada.

— custo relativamente baixo.

— resolução excelente (1024 × 1024 a 4096 × 4096).

Como inconvenientes podemos apontar:

— impossibilidade de modificar parcialmente a imagem, característica que torna difícil a sua utilização em aplicações interactivas.

— ausência de cores ou níveis de cinzento, o que limita consideravelmente o nível de realismo das imagens produzidas.

O conjunto das características indicadas restringe o campo de utilização dos tubos de memória às aplicações onde os níveis de interacção e de realismo não são extremamente importantes.

II. 2.4 — **Os ecrans a plasma**

Os ecrans a plasma também não exigem refrescamento e, além disso, a informação visualizada pode ser modificada selectivamente.

Este tipo de tecnologia utiliza uma pequena célula cheia de gaz como elemento de visualização. Esta célula, que pode ser escrita ou apagada individualmente pela activação de um par de electrodos X e Y [fig. 2.8], resolveria, à primeira vista, os problemas inerentes ao tubo de memória.

Enumeremos as principais características deste ecran:

— pode ser escrito e apagado selectivamente,

— não exige refrescamento,

— apresenta uma espessura da ordem de uma dezena de milímetros.

Quanto aos inconvenientes:

— as operações que permitem escrever e apagar são relativamente longas (20 microsegundos).

— a obtenção do efeito de cores é extremamente difícil.

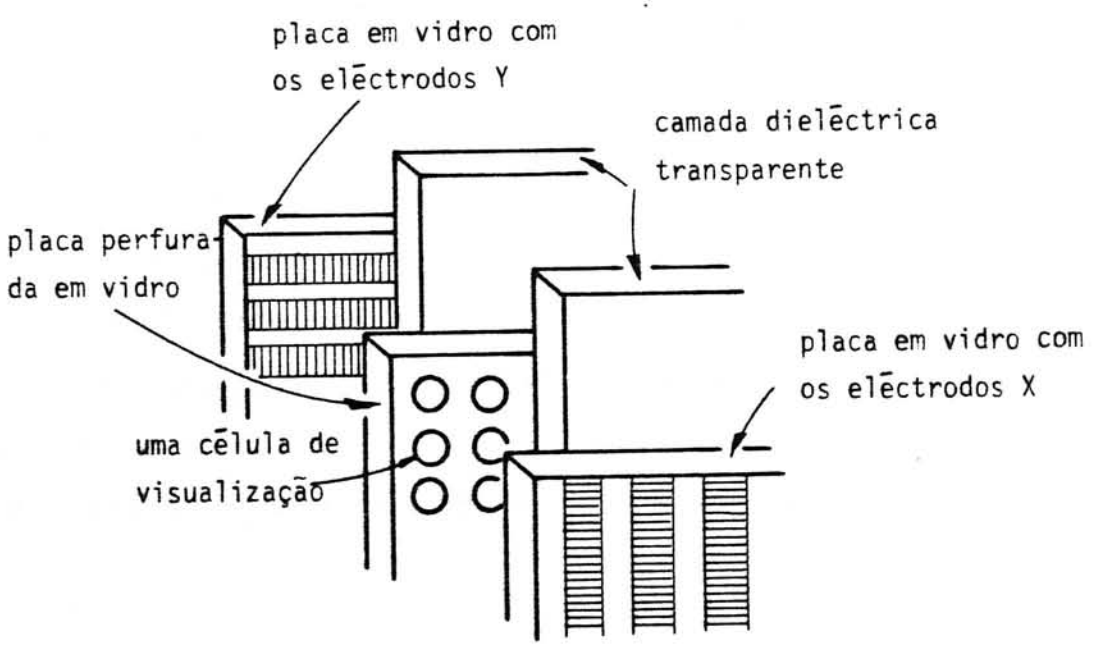


Fig. 2.8 – O princípio do ecran a plasma.

Este dispositivo de visualização, como acontecia com os tubos de memória, vê o seu campo de acção limitado às aplicações com níveis de interacção e de realismo relativamente pobres.

II. 2.5 — **Os ecrans a laser**

Este dispositivo de visualização baseia-se na utilização de um "laser" para produzir um feixe de luz de diâmetro extremamente pequeno. Este feixe de luz, ao descrever um percurso idêntico à imagem, impressiona uma película fotográfica, a qual, por sua vez, é projectada sobre uma parede.

Apesar das características que apresenta:

— alta precisão do traçado,

— 100 000 × 70 000 pontos podem ser endereçados,

— tamanho importante do ecran (100 cm × 70 cm),

— rapidez de escrita (5 m/s),

— possibilidade de vários níveis de cinzento,

estamos em presença de um dispositivo muito pouco divulgado, por ser extremamente caro e não muito adequado a aplicações interactivas.

Com efeito, o sistema electro-óptico que controla o deslocamento e a intensidade do raio emitido é bastante complexo.

Além disso, estes sistemas não permitem, em geral, a utilização de cores.

II. 2.6 — **Conclusão**

Do estudo precedente podemos deduzir que os dispositivos de visualização mais divulgados são:

— os tubos de memória

— os tubos de varrimento incremental

— os tubos de varrimento TV

Os primeiros devem a respectiva popularidade à simplicidade e ao baixo custo que os caracterizam, uma vez que a possibilidade de interacção e o nível de realismo que exibem são de pequena expressão.

Os segundos permitem uma interacção razoável, podendo mesmo dizer-se que são os mais indicados para aplicações de animação em tempo real. Todavia, apresentam grandes limitações no que respeita à qualidade das imagens que oferecem; número de cores reduzido e limitação do número de segmentos visualizáveis em boas condições (ausência de "flicker").

Os últimos, apesar dos problemas que lhes são criados pela memória de quadro são, quanto a nós, os mais prometedores. Recordemos as suas principais características:

— baixo custo

— gama enorme de cores

— imagens complexas e isentas de cintilação (flicker)

— resolução, a curto prazo, dos problemas suscitados pela memória de quadro.

Saliente-se, uma vez mais, a influência importante da micro-electrónica neste campo e o esforço de investigação actual que visa, simultaneamente, uma estruturação adequada do conteúdo da memória de quadro e uma arquitectura que se adapte razoavelmente aos requisitos necessários à síntese interactiva de imagens realistas.

II. 3. TENDÊNCIAS ACTUAIS DA ARQUITECTURA DOS SISTEMAS GRÁFICOS

Actualmente, um dos objectivos fundamentais da Computação Gráfica é a síntese de imagens realistas em tempo real; daqui resulta um certo número de tendências, mesmo ao nível do "hardware" associado aos sistemas gráficos.

Do ponto de vista do dispositivo de visualização, a tendência orienta-se para a utilização dos monitores a cores de varrimento TV, apesar de alguns problemas que continuam ainda sem uma resposta satisfatória [cf. II. 2.6].

A disponibilidade destes monitores data dos anos 60, mas a respectiva utilização efectiva só se torna evidente a partir dos anos 70. Este facto está relacionado com o aparecimento das memórias MOS, que tornam viável a realização de módulos de memória com uma capacidade idêntica à memória de quadro exigida pela arquitectura de um sistema deste tipo [fig. 2.7].

O progresso da micro-electrónica repercute-se igualmente no aumento da inteligência local, através da utilização de microprocessadores ao nível do processador gráfico. Este facto permite minimizar as trocas de informação entre o terminal e o computador principal o qual, em certos casos, pode mesmo ser eliminado [fig. 2.9].

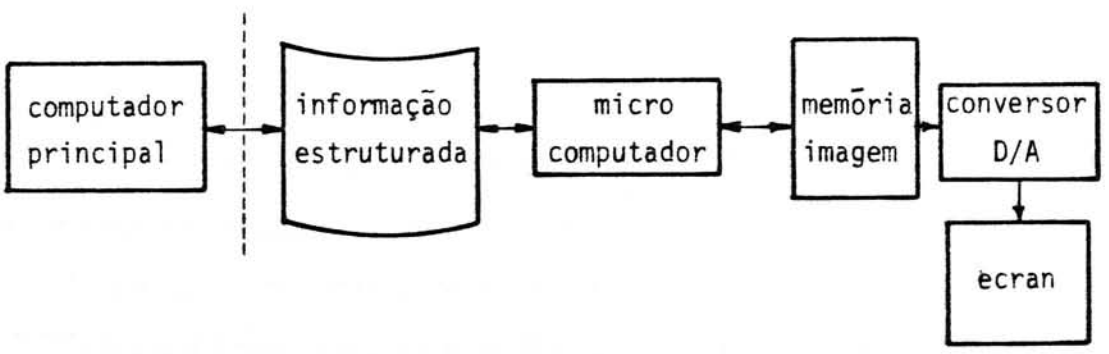
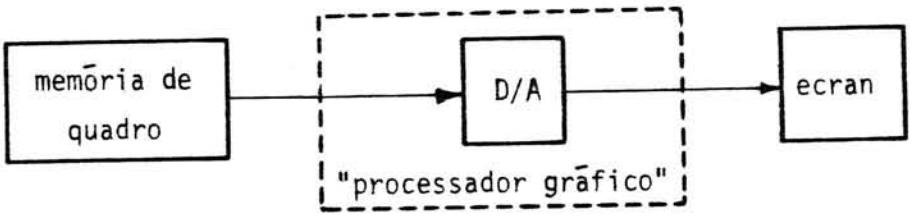
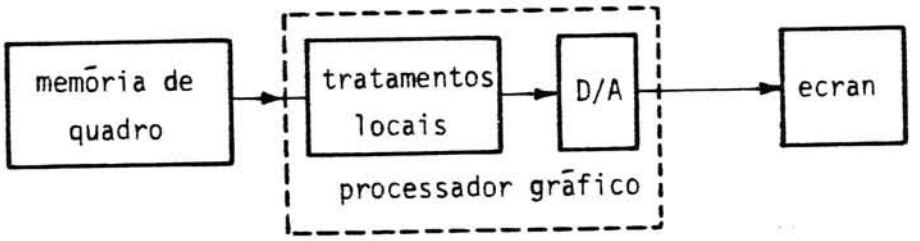


Fig. 2.9 – A influência da inteligência local na arquitectura de um sistema gráfico.

Todavia, o ponto mais interessante, alvo da investigação actual, diz respeito à estruturação da informação retida pela memória de quadro, a qual vai permitir toda uma série de tratamentos locais, em tempo real, ao nível do processador.



a. solução clássica



b. tendência actual

Fig. 2.10 – A evolução do processador gráfico.

Numa estrutura clássica [fig. 2.10 a], a memória de quadro contém a cor associada a cada ponto do ecran e é designada, por este facto, por *memória imagem.*

Os principais inconvenientes neste caso são:

— impossibilidade de identificar localmente os elementos visualizados,

— o conteúdo da memória imagem deve ser totalmente refeito, desde que ocorra uma modificação de um tipo qualquer de informação.

Pelo contrário, se a informação retida na memória de quadro apresenta uma certa estruturação, alguns tratamentos serão da responsabilidade do processador gráfico e os inconvenientes anteriormente citados são minimizados. Neste caso, a memória de quadro é designada por *memória de identificação* [fig. 2.10 b].

II. 4. ANÁLISE DE ALGUNS SISTEMAS EXISTENTES

Os sistemas utilizados na síntese de imagens são geralmente caracterizados pelos três factores seguintes:

— realismo das imagens que produzem,

— possibilidade de animação ou de dinamismo em tempo real,

— capacidade de interacção.

O realismo das imagens está intimamente ligado aos dispositivos de visualização utilizados, aos modelos de descrição das cenas e aos fenómenos reais simulados.

A animação ou dinamismo em tempo real significa que se pode criar sobre o ecran uma visualização com movimento. Esta característica exige a renovação completa da imagem visualizada num período de 40 ms (isto para um standard de 25 imagens/segundo).

A capacidade de interacção define as potencialidades de diálogo homem-máquina.

O *sistema ideal* seria, consequentemente, aquele que produzisse de uma forma interactiva imagens com um elevado nível de realismo, em tempo real. Contudo, os requisitos devidos ao tempo real e ao realismo, sendo de natureza antagónica, tornam impossível o sistema ideal, o que obriga a dosear os dois factores em causa. Pretende-se, desta maneira, encontrar soluções de compromisso que respondam satisfatoriamente a cada tipo de aplicação.

A diversidade relativamente importante de sistemas utilizados na síntese de imagens impõe a definição de uma metodologia adequada ao respectivo estudo comparativo.

A base da metodologia que propomos reside na análise, sistema por sistema, dos diferentes tipos de informação que manipulam e da correspondente capacidade de resposta.

A análise das diferentes informações manipuladas dará uma ideia do nível de realismo das imagens produzidas. Quanto à capacidade de resposta, consideramos que está estreitamente ligada com o tipo de realização utilizada, "hardware" ou "software"; em geral, os tempos de resposta mais longos estão associados ao último tipo de realização indicado.

Entre os sistemas existentes consideraremos os que julgamos mais significativos, agrupados em duas categorias:

— sistemas comercializados (justificados pelo número de unidades em serviço)

— sistemas que são alvo da investigação actual.

II. 4.1 — **Metodologia de análise de sistemas gráficos**

Uma metodologia de análise de sistemas gráficos deve pôr em evidência os critérios mais interessantes que serão utilizados numa análise crítica comparativa dos sistemas em causa.

R. Baecker propõe uma solução [Bae 79] que considera que um sistema deste tipo pode sempre ser decomposto em módulos de memória e de processamento, estruturados de maneira adequada. Pretende assim estudar os sistemas existentes e até mesmo, prever a eficácia de sistemas ainda em projecto.

Os resultados a que chega são, contudo, um pouco vagos.

O método que propomos apoia-se nas respostas às três questões seguintes:

— Quais os tipos de informações que chegam ao sistema de síntese de imagens?

— A que nível se realiza a síntese?

— Qual o tempo de resposta do sistema às modificações daquelas informações?

Enquanto que a resposta à primeira questão permite avaliar o nível de realismo oferecido pelo sistema, a resposta às duas últimas dá uma ideia das "performances" das suas reacções.

Consideremos, por exemplo, os sistemas que produzem imagens com um elevado nível de realismo, mas onde se verifica uma ausência nítida dos factores interacção e tempo real.

Neste caso, o processamento relativamente complexo dos modelos relativos a vários fenómenos reais (reflexão, transparência, ...) implica, em geral, uma realização "software" que absorve algumas dezenas de minutos de cálculo por cada imagem [fig. 2.11].

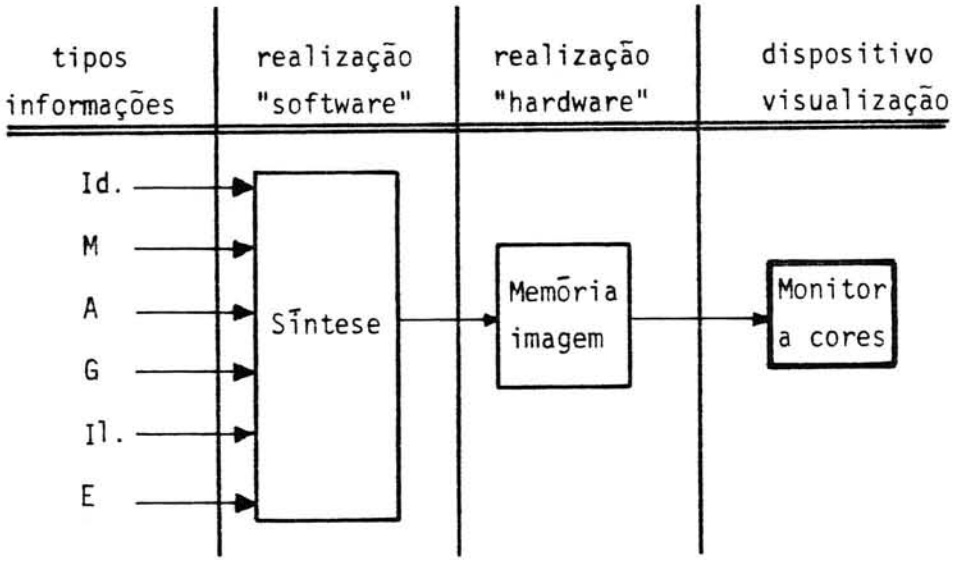


Fig. 2.11 – Estrutura de um sistema que produz imagens com elevado nível de realismo.

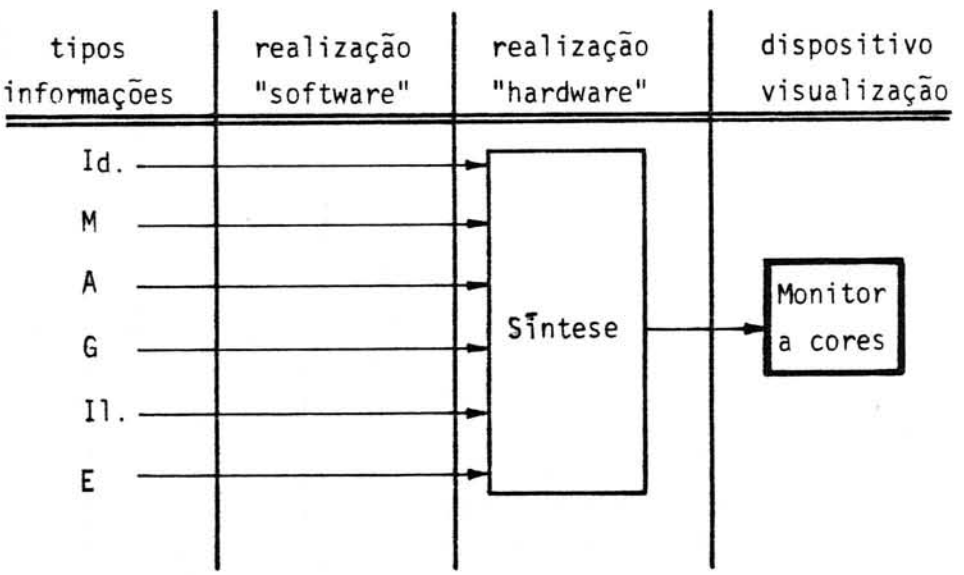


Fig. 2.12 – A estrutura do sistema ideal.

Pelo contrário, parece-nos evidente que, para o mesmo nível de realismo, o factor tempo real exige uma realização "hardware", como se indica na Figura 2.12, a que corresponderia o "sistema ideal".

Entre os dois casos apontados, os sistemais existentes adoptam soluções de compromisso e os seus esquemas estruturais permitirão avaliar o nível de realismo e a capacidade de resposta que os caracterizam.

II. 4.2 — **Alguns sistemas comercializados**

Os sistemas que se seguem, classificados por ordem de complexidade crescente, dão-nos uma ideia dos produtos actualmente disponíveis no mercado.

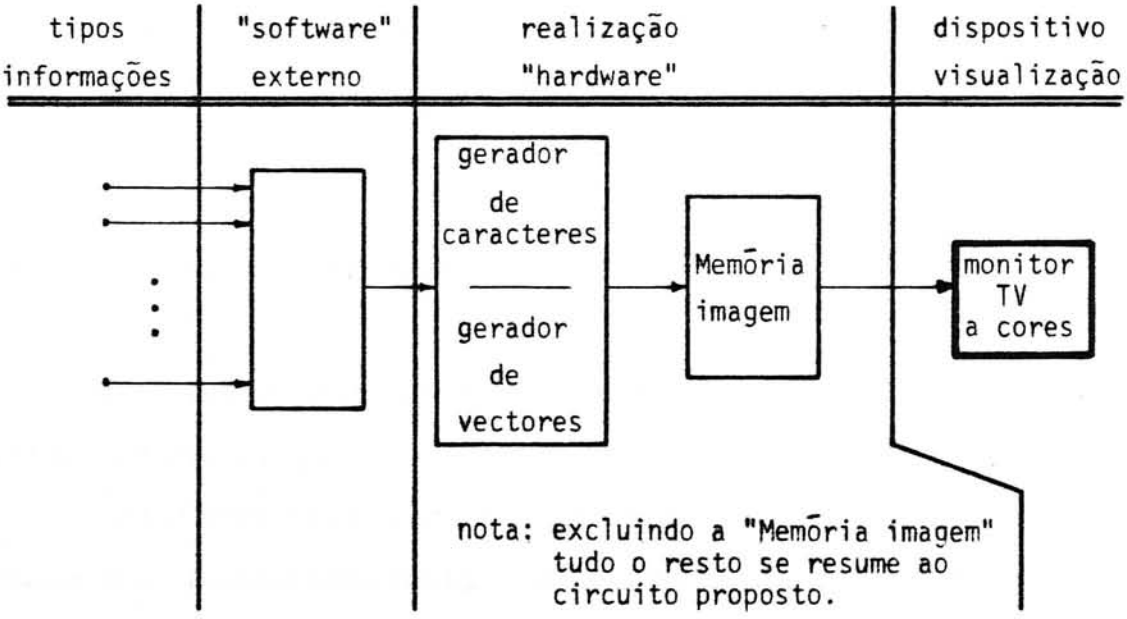


Fig. 2.13 – Estrutura de um sistema baseado no circuito proposto.

II. 4.2.1 — **Um sistema simples**

Este sistema, apresentado por P. Mathérat [Mat 78], é muito simples, mas bastante flexível.

O circuito integrado LSI (large scale integration) proposto executa as funções fundamentais de controlo de uma memória de quadro, neste caso, uma memória imagem [fig. 2.13].

Outras características igualmente interessantes deste circuito são:

— diversas resoluções da imagem (entre 64 × 64 a 512 × 512),

— número de cores apenas função do comprimento da palavra da memória de quadro,

— utilização possível de uma caneta luminosa (light pen) ou de um cursor,

— sinais de sincronização de um monitor de varrimento TV disponíveis.

O custo pouco elevado e a ligação simples com um sistema de tratamento qualquer (normalmente um microcomputador de 8 bits) constituem as grandes vantagens dos sistemas gráficos realizados com base neste circuito.

Contudo, o nível de realismo das imagens produzidas dependerá sempre de um programa externo e, consequentemente, a sua capacidade de resposta não pode ser muito importante.

II. 4.2.2 — **Sistemas com software interno**

Nesta classe vamos considerar dois sistemas, o TEKTRONIX 4027 e o HEWLETT-PACKARD 9845C.

— o TEKTRONIX 4027 é, em relação ao exemplo anterior, um sistema pronto a funcionar. O seu esquema estrutural [fig. 2.14] põe em evidência um "software" interno que controla os acessos à memória de quadro, cujo conteúdo apresenta uma certa compressão. Este facto permite a visualização de imagens com uma resolução 640 × 480, para uma capacidade de

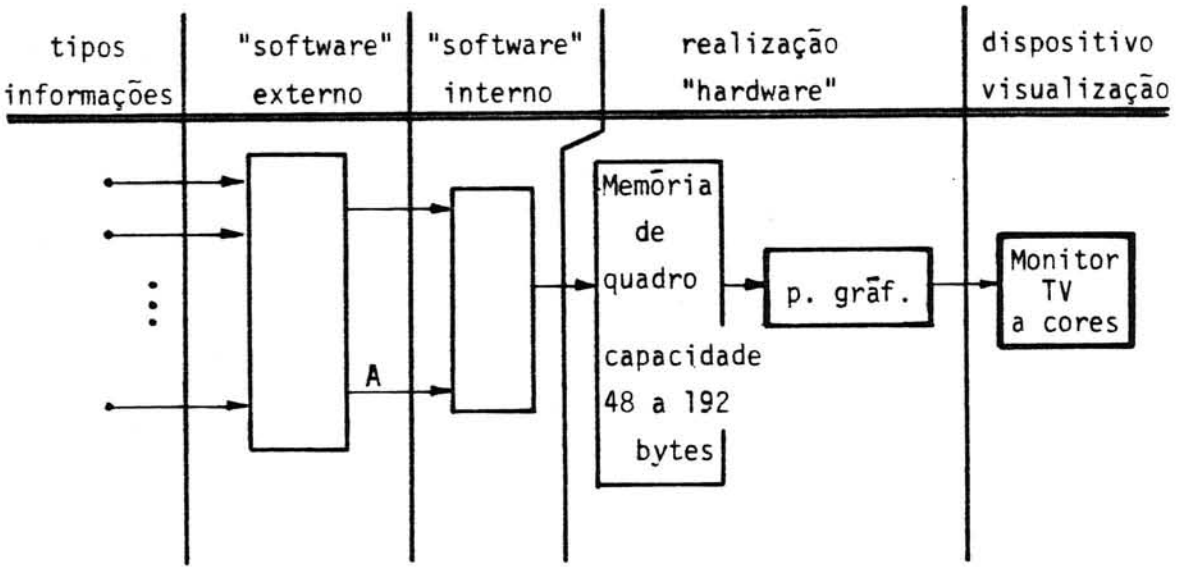


Fig. 2.14 – O esquema estrutural do TEKTRONIX 4027.

memória de quadro que se pode situar entre 48 e 192 Kbytes. O número de cores simultaneamente visualizáveis é de 8, mas podem ser escolhidas entre 64 disponíveis.

As "texturas" consideradas pelo "software" interno (informação do tipo A (aspecto) na figura 2.14) são muito pouco funcionais; numa matriz 8 × 14 podemos especificar unicamente uma cor de fundo e a cor de pontos interiores (a mesma para todos os pontos escolhidos).

Podemos assim dizer que os diferentes tipos de informação são manipuladas por um "software" externo, que condicionará obrigatoriamente o grau de realismo das imagens e a capacidade de resposta do sistema.

— O esquema e a maior parte das observações indicadas a propósito do 4027 são ainda válidas para o terminal HEWLETT-PACKARD 9845C; as principais diferenças são:

- ausência de compressão ao nível da informação retida pela memória de quadro que, consequentemente, pode atingir uma capacidade entre 56 a 449 Kbytes,
- resolução da imagem 560 × 455,
- possibilidade de visualizar um pouco mais de 4000 cores,
- custo mais elevado.

II. 4.2.3 — **Visualização de sequências animadas pré-calculadas**

Nesta classe vamos igualmente considerar dois sistemas, o CVD/2 da Rivers Computer Corporation e o RAMTEK 9400.

— pelo esquema estrutural do terminal CVD/2 [fig. 2.15], podemos deduzir que neste sistema o realismo das imagens depende necessariamente de um programa externo, facto que impossibilita qualquer resposta interactiva em tempo real.

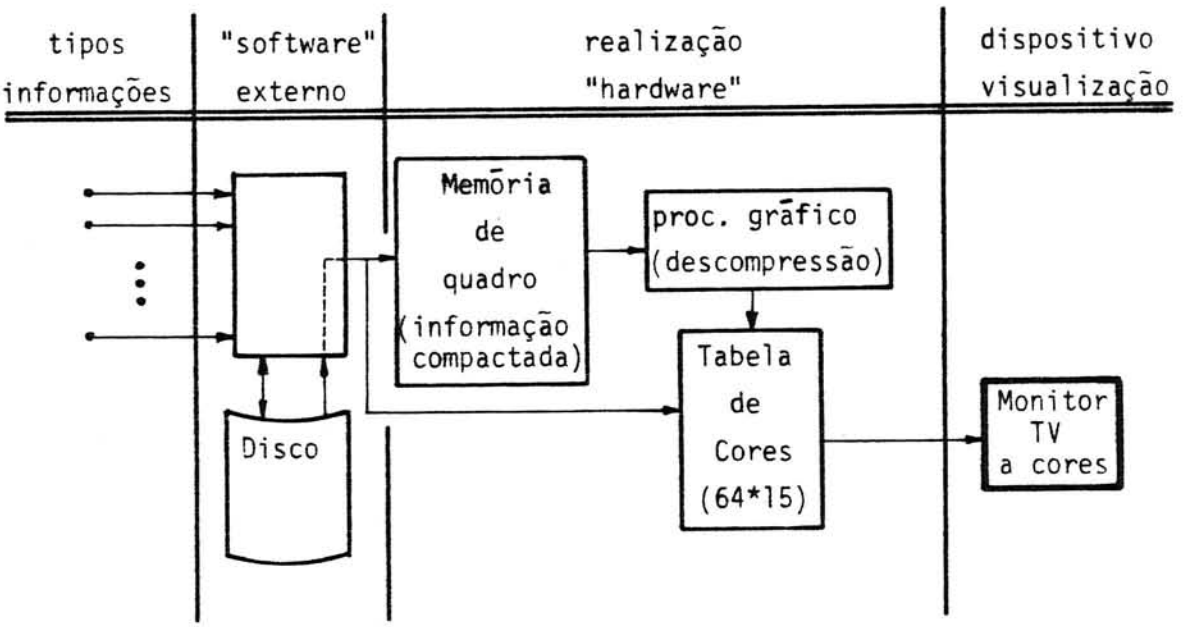


Fig. 2.15 – O esquema estrutural do CVD/2.

Todavia, é possível com este terminal visualizar imagens em tempo real (!...), se estas tiverem sido previamente calculadas, compactadas e guardadas em disco.

Saliente-se que a compressão da informação, após o cálculo respectivo, é extremamente útil quando a imagem apresente zonas com cores uniformes. Neste caso, a descrição de um

segmento horizontal unicolor resume-se ao código da cor e ao valor que representa o comprimento daquele segmento.

As vantagens da compressão traduzem-se pela economia de capacidade de memória secundária e de memória de quadro e, ao nível da transmissão, pela redução da respectiva largura de banda.

A compressão traz ao processador gráfico uma nova função, a "descompressão" da informação a visualizar.

Um sistema deste tipo permite a criação de sequências animadas em tempo real (isto é, criação de uma nova imagem todos os 40 ms) directamente sobre o ecran, portanto sem a intervenção de aparelhagem fotográfica.

Apesar da codificação das cores ao nível da memória de quadro se resumir a 8 bits, é possível visualizar simultaneamente 64 cores diferentes, à escolha entre 2 ×× 15. De facto, o processador gráfico dispõe de uma *Tabela de Cores* com uma organização 64 × 15, acessível pelo "software" externo [fig. 2.15]. O utilizador pode desta maneira programar as 64 cores de que vai necessitar.

Esta facilidade permite ainda uma certa "animação colorida", modificando regularmente o conteúdo daquela tabela, técnica já explorada por R. G. Shoup [Sho 79].

— A originalidade do terminal RAMTEK 9400 consiste em reunir as vantagens do varrimento incremental, em particular, a possibilidade de definir imagens a partir de vectores, e as vantagens do varrimento do tipo TV (gama de cores importante, imagem estável independentemente da sua complexidade, ...).

O utilizador tem acesso a uma memória integrada ao terminal, que retém a imagem definida por um conjunto de vectores [fig. 2.16]. Este acesso é possível graças a um "software" interno, executado por um microcomputador (Z 80), com um espaço virtual de cálculo de 32 K × 32 K; uma imagem obtém-se definindo uma "janela" naquele espaço virtual.

Pode-se criar uma certa animação quando se desloca aquela "janela".

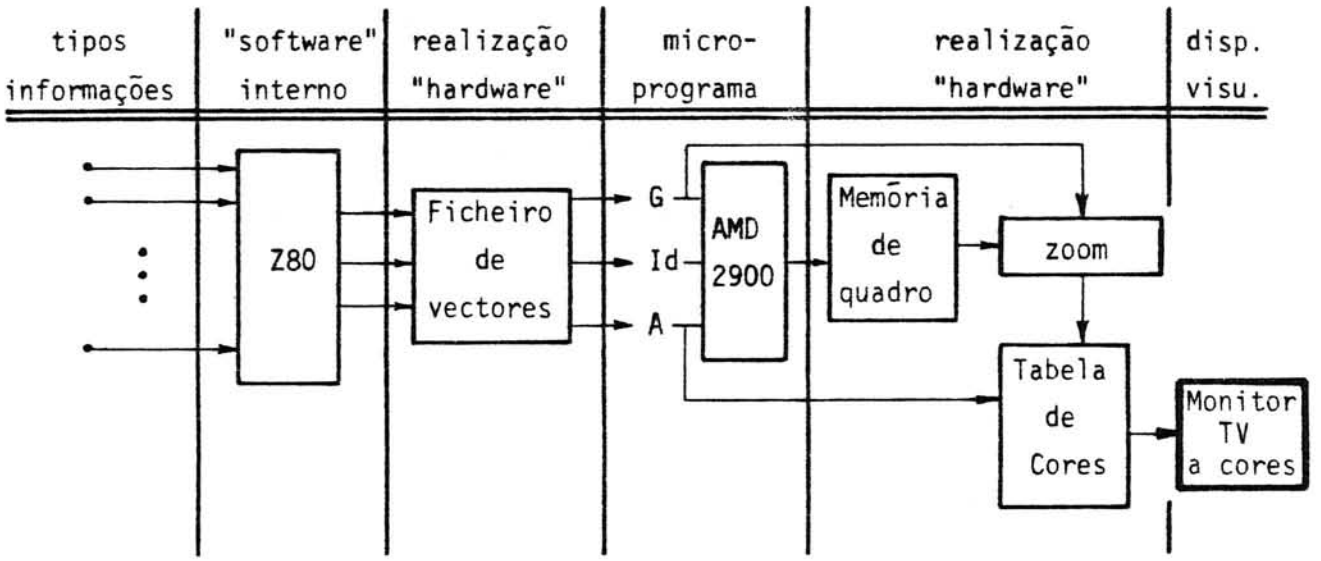


Fig. 2.16 – O terminal RAMTEK 9400.

O processamento das informações geométricas (zoom, translação e rotação) e de aspecto (preenchimento de polígonos) é também considerada pelo terminal, graças a um microcomputador bastante rápido, baseado num microprocessador bipolar (bit slice) da família AMD 2900, com um ciclo de instrução da ordem dos 280 ns. É o resultado deste processamento que vai constituir o conteúdo da memória de quadro, a qual permanecerá constantemente "invisível" para o utilizador.

Um "zoom" realizado por "hardware" também equipa este terminal, que dispõe ainda de uma tabela de cores [fig. 2.16].

O fabricante oferece diferentes resoluções de imagem, sendo a mais original 1024 × 1280. Esta afasta-se do standard video e implica, consequentemente, alguns inconvenientes:

- impossibilita a utilização de uma aparelhagem video standard,
- gera 25 quadros por segundo em vez de 50, o que origina uma cintilação inadmissível,
- exige monitores especiais cujo custo é bastante elevado.

Resumindo, estamos na presença de um terminal que processa de uma forma eficaz as informações geométricas (translação, rotação e zoom), as informações de aspecto (preenchimento de polígonos, afectação de "texturas" a vectores) e as informações de identificação (identificação de um vector pela simples designação, sobre o ecran, de um dos seus elementos).

Em tempo real, dispõe de um "zoom" a 16 níveis e a geração de pseudo-cores graças a uma tabela de cores.

II. 4.2.4 — **Simulação de Condução**

Agrupámos nesta categoria os sistemas utilizados em simulação de condução, fabricados em geral pelas firmas norte-americanas McDonnell Douglas, General Electric e Evans and Sutherland.

As publicações que consideram este tipo de sistemas não são suficientemente pormenorizadas para nos permitir um estudo como desejaríamos. Algumas fontes que tocam este tema [Wei 78], põem em evidência as potencialidades de cálculo que sistemas deste tipo utilizam; alguns computadores em paralelo, circuitos especialmente concebidos para resolverem certos problemas característicos (cálculo de visibilidade, por exemplo) e ainda certos meios, mais ou menos sofisticados, como, por exemplo, a produção de resultados visualizados pela mistura de imagens sintetizadas com imagens obtidas graças a uma câmara TV em circuito fechado, que se desloca, comandada pelo sistema, sobre uma maquete de cena em causa.

Estas características explicam o custo elevadíssimo destes sistemas, o que limita a respectiva utilização às grandes companhias de aviação, ao exército ou à NASA.

Alguns documentos publicitários difundidos por McDonnell Douglas Corp. e uma publicação de G. Handberg [Han 77] da mesma empresa, permitiu-nos a apresentação do exemplo que se segue.

VITAL IV é um sistema utilizado no treino de pessoas ligadas à aviação civil e militar e comporta fundamentalmente [fig. 2.17]:

— um computador de potência média, equipado de uma memória secundária importante (discos),

— vários circuitos "pipe-line" orientados para esta aplicação,

— um sistema de visualização composto de vários monitores a cores e ainda um sistema óptico adequado,

— painéis de controlo (painel-aluno, painel-instructor).

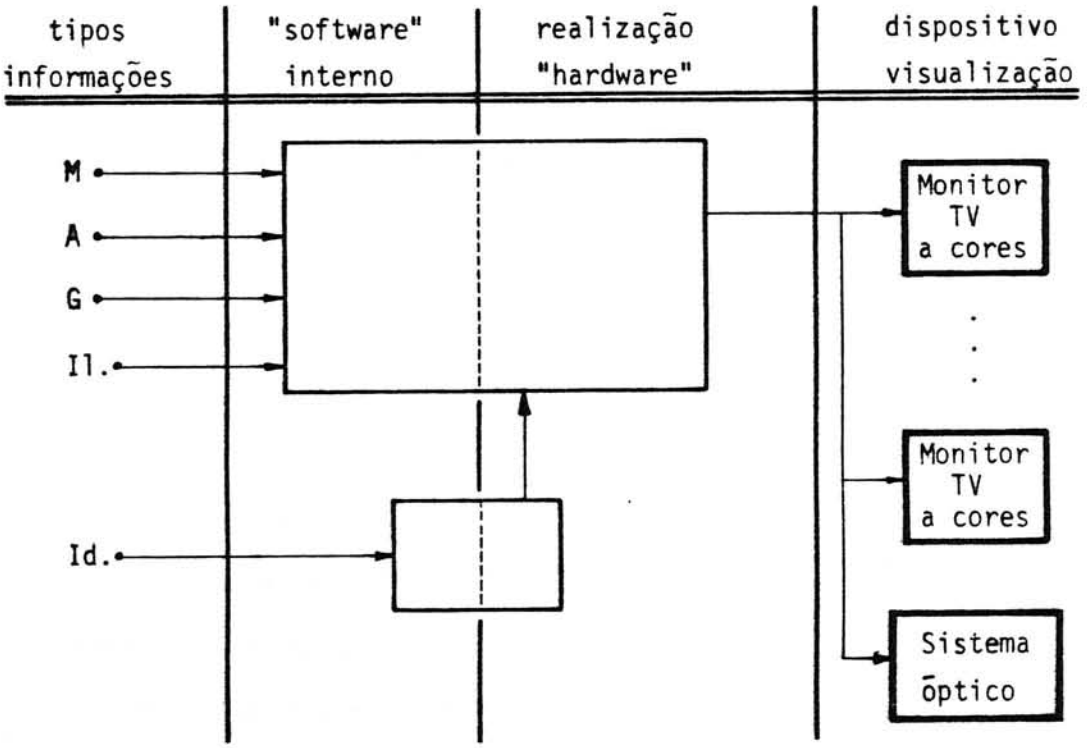


Fig. 2.17 – Um sistema utilizado na simulação de condução.

Graças ao "software" interno e a circuitos "pipe-line" (não dispomos de dados suficientes que nos permitam definir a fronteira entre as tarefas da responsabilidade do "hardware" e do "software") vários tipos de informações são manipuladas em tempo real:

— o sistema considera unicamente uma fonte luminosa difusa, produzindo imagens sem reflexos. Contudo a descrição de uma cena noturna pode comportar até 400 000 pontos de luz, dos quais 8 000 podem ser visualizados simultaneamente,

— as faces apresentam cores uniformes (10 cores são possíveis),

— os objectos deslocam-se em função dos comandos com origem nos painéis já citados; a eliminação das partes escondidas é também efectuada,

— a forma dos objectos é descrita a partir de pequenas faces planas definidas por quatro vértices; o sistema processa até 15 000 faces, das quais 300 podem ser visualizadas simultaneamente,

— a identificação de um objecto é implícita, pois, por exemplo, a um veículo estará associado um certo conjunto de botões dos painéis.

Resumindo:

— nível de realismo das imagens bastante baixo,

- faces planas de cor uniforme
- 10 cores disponíveis
- ausência de reflexos

— capacidade de resposta,

- cálculo de 50 quadros por segundo

— gama de aplicações reduzida,

- simulação de condução

— custo extremamente elevado.

II. 4.3 — A investigação actual: alguns sistemas característicos

O estudo da síntese de imagens realistas realizado no capítulo I pôs em evidência numerosos trabalhos, cujas preocupações principais diziam respeito:

— ao aumento do nível de realismo das imagens sintetizadas

Os objectos são descritos por superfícies não planas e fenómenos como a transparência e a reflexão são tomados em linha de conta; são também propostos alguns algoritmos para o cálculo de visibilidade de tais objectos e técnicas que integram a utilização de texturas e de vários pontos luminosos direccionais. Tudo isto tornou possível a síntese de imagens coloridas magníficas que exibem, nomeadamente, reflexos e sombras.

Exceptuando um pequeno número de trabalhos (como, por exemplo, [Sta 78]), as realizações em questão envolvem sempre soluções de natureza "software", que requerem largas dezenas de minutos de cálculo por imagem.

— ao desenvolvimento de sistemas que utilizam alguma forma de compressão da informação manipulada, permitindo assim a criação de sequências animadas em tempo real, isentas de interacção, mas sem a intervenção de aparelhagem fotográfica [StT 79].

Nestes casos, é possível um certo nível de realismo graças ao grande número de faces planas que os sistemas podem tratar [CHP 79].

— à investigação no domínio da animação interactiva em tempo real, para aplicações do tipo simulação de condução.

Em relação aos grandes sistemas que já mencionámos [cf. II. 4.2.4], os trabalhos englobados nesta categoria pretendem [Ler 80]:

- encontrar soluções que apresentem um custo inferior ao daqueles sistemas,
- aumentar o nível de realismo das imagens,
- aumentar a gama de aplicações.

Pormenorizemos cada um dos casos citados e tentemos por em evidência as respectivas originalidades do ponto de vista de realização "hardware".

II. 4.3.1 — **Visualização de moléculas em tempo real**

Vamos tomar como exemplo um sistema utilizado na visualização de moléculas orgânicas complexas [Sta 78].

O modelo que descreve cada molécula é formado por um conjunto de esferas coloridas (átomos), cujo número pode atingir 1000 unidades visíveis.

Saliente-se que a iluminação de uma superfície esférica é relativamente simples, pois apresenta, durante o respectivo varrimento, um comportamento idêntico de linha em linha. Assim, a originalidade desta realização [fig. 2.18] reside na utilização de uma tabela que simula aquele comportamento, implementada com uma memória extremamente rápida, permitindo o cálculo em tempo real da iluminação de cenas deste tipo.

O autor prevê ainda alguns desenvolvimentos futuros, com o objectivo de alargar este princípio à visualização de outras superfícies não planas, frequentemente presentes em cenas reais (superfícies cónicas, cilíndricas, ...).

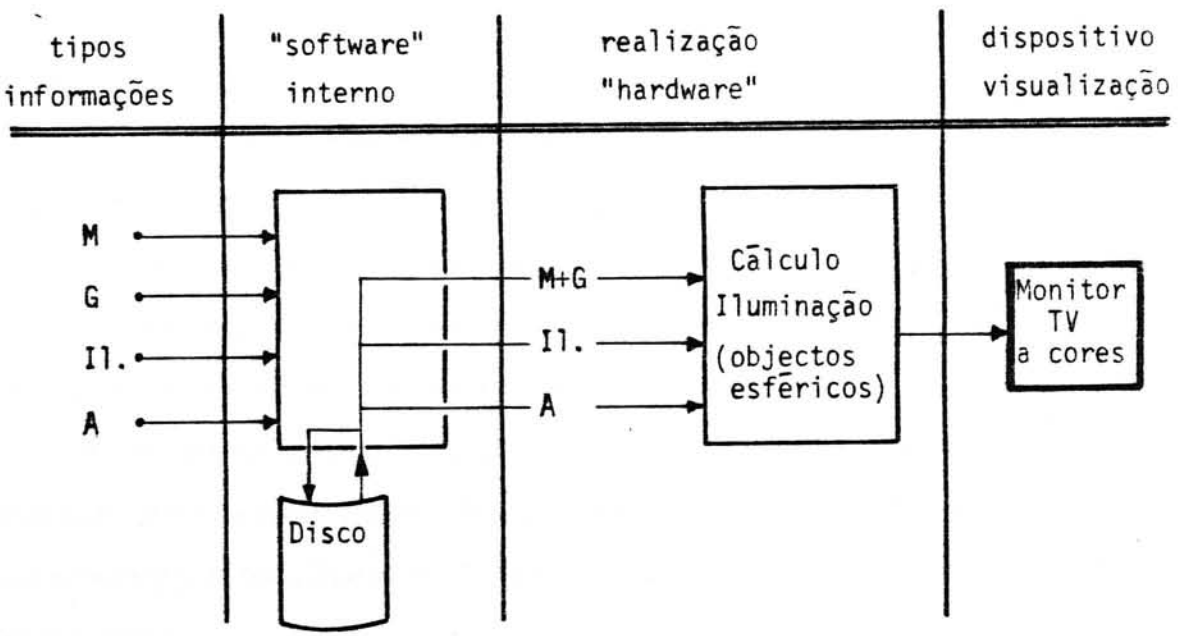


Fig. 2.18 – Exemplo de um sistema que calcula, em tempo real, a iluminação de cenas compostas de objectos esféricos.

É igualmente possível a produção de sequências animadas em tempo real (sem interacção) desde que uma certa compressão da informação possa estar presente. A existência de tal compressão dependerá da complexidade da cena, esta, de alguma maneira, associada com o número de "átomos" em jogo; a partir de uma certa complexidade o nível de compressão torna-se demasiado baixo, originando problemas durante a transferência de informação entre a memória secundária e o sistema de visualização.

Por exemplo, para uma molécula que comporta 1000 átomos aproximadamente, a largura de banda do canal envolvido naquela transferência admite unicamente 12 imagens por segundo.

II. 4.3.2 — **Visualização de sequências animadas em tempo real com um elevado nível de realismo**

O sistema que vai ser considerado como exemplo [StT 79] é um desenvolvimento de trabalhos anteriores [OgS 75], [Eas 75], [Eng 75], que explora o aproveitamento da compressão da informação a visualizar.

As cenas descritas exclusivamente por faces planas permitem, em geral, uma forte compressão da informação, quando se exprime cada segmento horizontal unicolor pelo seu comprimento e pelo código da cor correspondente. A produção de sequências animadas em tempo real, desprovidas de interacção, torna-se então possível, a partir da informação pré-calculada, compactada e guardada em disco.

Evidentemente, para que tudo isto se torne possível, é essencial que o nível médio de compressão presente numa imagem não seja inferior a um certo limite imposto, por exemplo, pelas características tecnológicas da memória secundária e do canal de transmissão onde se encontra inserida.

No sistema em questão, o limite imposto ao nível médio da compressão foi substancialmente diminuído graças à utilização de um duplo "tampão" de memória [fig. 2.19].

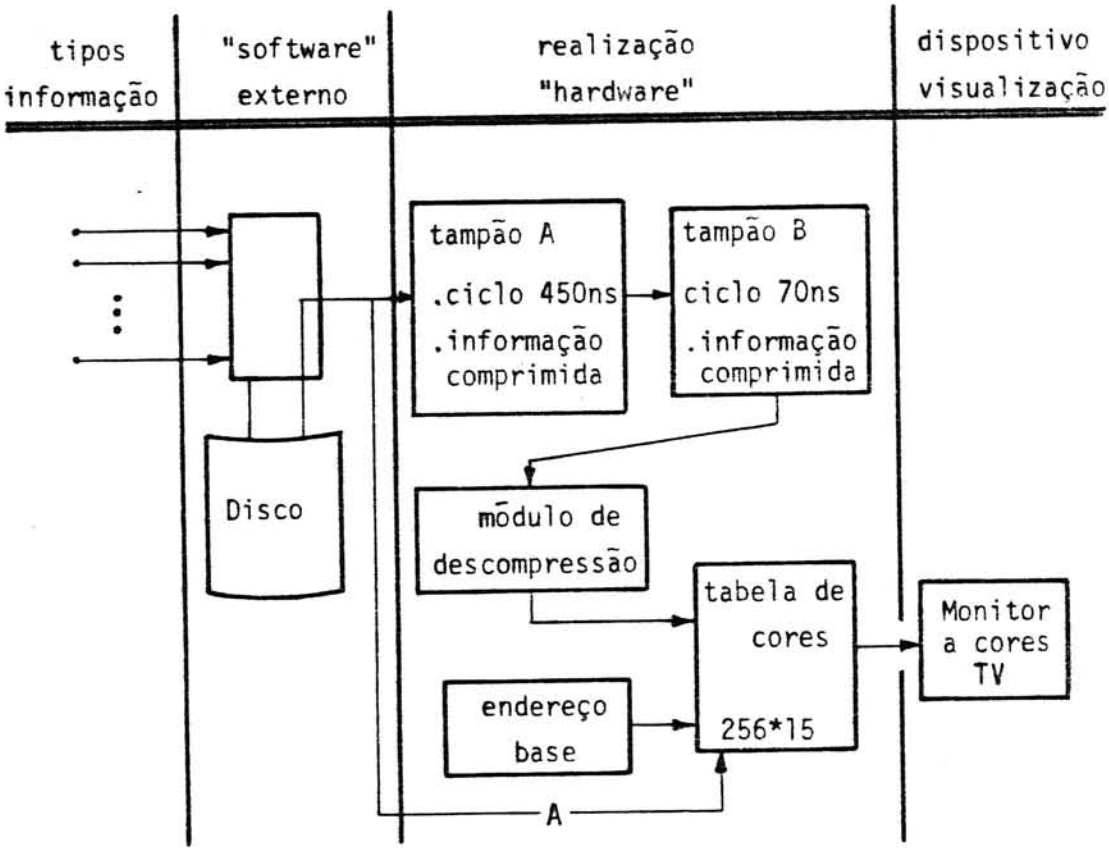


Fig. 2.19 – Um sistema utilizado na produção de sequências animadas pré-calculadas.

Um primeiro "tampão", realizado com memórias MOS de 450 ns de ciclo, apresenta uma capacidade muito mais importante que um segmento "tampão", este utilizando memórias muito mais rápidas (ciclo de 70 ns), mas também mais dispendiosas.

Desta maneira, durante os retornos de linha e de quadro e nas zonas da imagem onde as cores não mudam frequentemente, atinge-se em geral o preenchimento total do primeiro "tampão", o qual realiza com o segundo uma espécie de "memória virtual".

Uma tabela de cores com uma capacidade 256 × 15 (5 bits por cada cor), endereçada directamente pelo resultado da operação "descompressão" (8 bits), permite utilizar 256 cores entre 2 ×× 15 possíveis; saliente-se que esta tabela é programável do exterior.

Se se considerar que algumas das posições desta tabela podem ser modificadas durante os retornos de linha (ou de quadro), o número de cores visualizáveis em cada imagem ultrapassa largamente 256.

O autor prevê ainda um registo "endereço de base" [fig. 2.19] para a tabela de cores, com o qual poderá modificar instantaneamente as 256 cores disponíveis.

Saliente-se ainda que esta tabela possibilita uma animação colorida em tempo real [Sho 79].

Este sistema inspirou pelo menos um trabalho de desenvolvimento interessante [CHP 79], no qual podem ser visualizadas, em tempo real, cenas compostas de alguns milhares de faces. Por exemplo, imagens idênticas ao que um observador veria quando se desloca em torno de uma árvore modelizada por 20 000 faces planas, aproximadamente.

II. 4.3.3 — Produção de animação interactiva em tempo real

Nesta categoria tomámos como exemplo o sistema GSI 1 apresentado por P. Leray [Ler 80].

O método utilizado para resolver o problema de eliminação das partes escondidas de uma cena (cálculo de visibilidade) é bastante interessante. Podemos nele distinguir duas fases:

- Um pré-tratamento de cena, em diferido, determina as prioridades de visualização das faces que entram na composição da cena, independentemente da posição do observador (cálculo das prioridades estáticas).
- Um cálculo, em tempo real, das prioridades dinâmicas, função da lista das prioridades estáticas e da posição do observador.

O resultado final [fig. 2.20] não é mais que uma lista de faces, ordenadas da mais afastada à mais próxima do observador; a imagem obtém-se sobrepondo sucessivamente aquelas faces de acordo com a lista das prioridades dinâmicas.

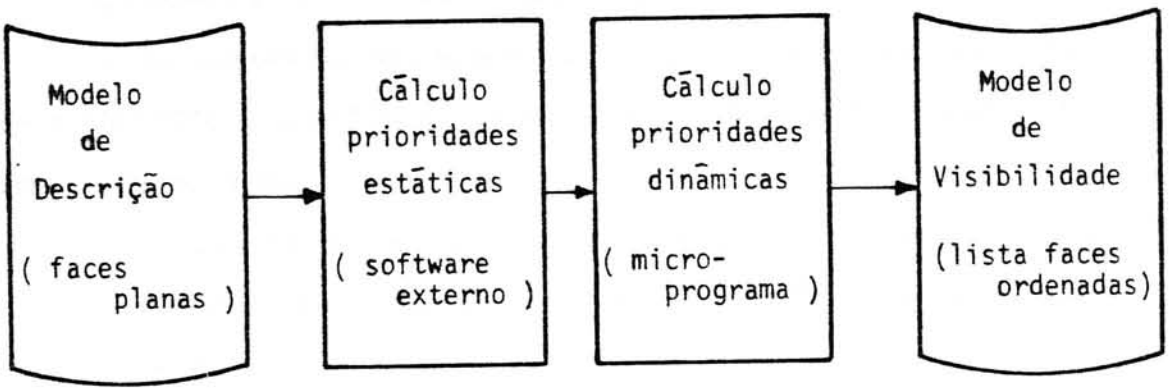


Fig. 2.20 – O cálculo de visibilidade no sistema GSI 1

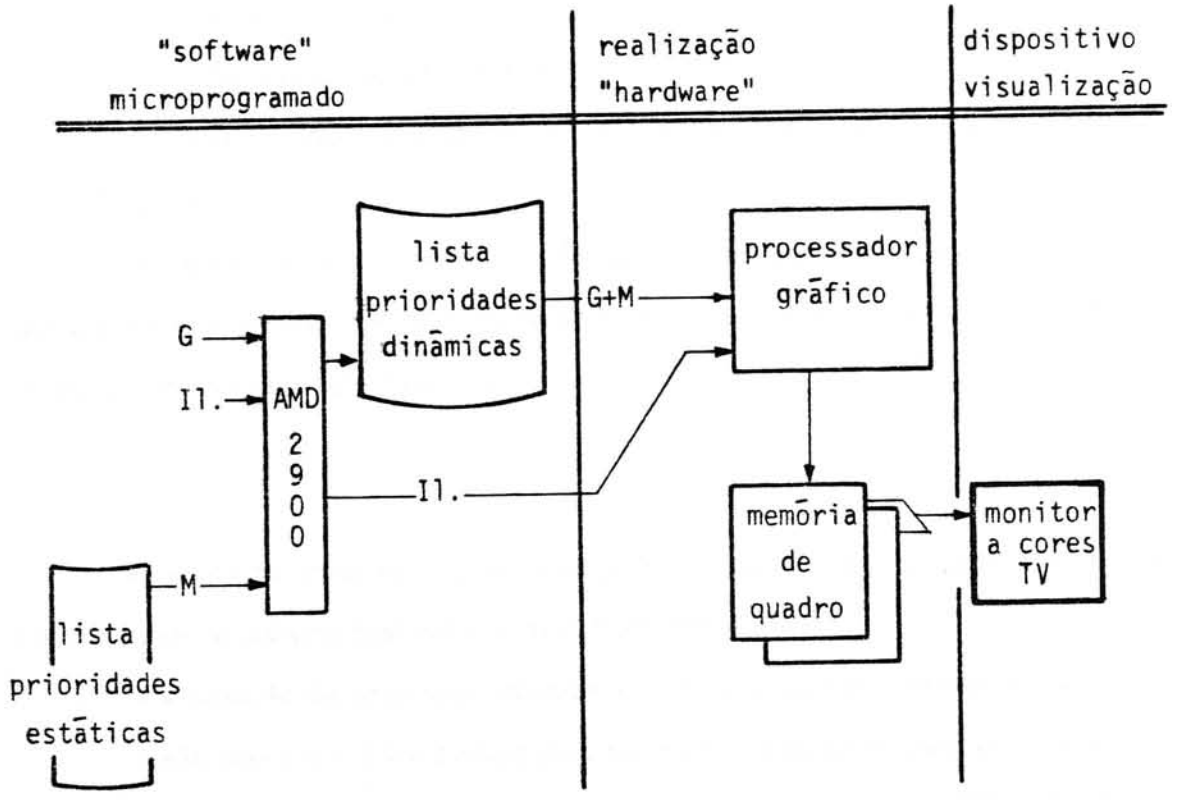


Fig. 2.21 – O esquema estrutural do sistema GSI 1.

O esquema estrutural do sistema GSI 1 [fig. 2.21] comporta:

— um *"software microprogramado"* executado por um microcomputador bipolar da família AMD 2900 (ciclo de 200 ns), que toma como dados a lista de prioridades estáticas e manipula, em tempo real:

- as informações geométricas (translação, rotação e lista das prioridades dinâmicas),
- as informações de iluminação (iluminação das faces tomando em linha de conta a direcção de uma fonte luminosa),
- as informações morfológicas (lista das prioridades estáticas).

— um *módulo "hardware"* que manipula as informações anteriormente consideradas e executa em tempo real:

- o preenchimento das faces,
- o cálculo da iluminação ponto a ponto.

— a *memória imagem* composta de 2 planos, cada um com uma capacidade $512 \times 512 \times 8$.

Um destes planos contém a imagem corrente enquanto que no outro vai sendo armazenada a informação correspondente à imagem seguinte; logo que esta operação termine, os planos alternam as respectivas funções.

Este sistema revela-se relativamente poderoso, pois oferece um dinamismo interactivo com um nível de realismo bastante razoável. Com efeito:

— A direcção de uma fonte luminosa unicolor é tomada em linha de conta,

— são oferecidos $2 \times\times 8$ níveis de cinzento, mas é proposta uma solução para implementar a cor neste terminal; uma tabela de cores com uma capacidade 256×24 (8 bits para cada cor fundamental) endereçada directamente pelos 8 bits actualmente calculados.

— são visualizáveis em tempo real entre 200 e 800 faces.

Por exemplo, um objecto do tipo avião de 132 faces exige 10 ms de tempo de cálculo, o que está ainda muito longe dos 40 ms admissíveis.

Comparando agora o sistema GSI 1 com os grandes sistemas utilizados actualmente na simulação de condução [cf. II. 4.2.4], consideramos que aquele oferece um certo número de vantagens, em particular:

— custo muito mais baixo,

— nível de realismo muito mais elevado,

— possibilidade de uma mais ampla gama de cores.

O grande inconveniente reside no pré-tratamento em diferido, cujo tempo de execução pode atingir algumas dezenas de minutos.

Além disto, subsistem ainda alguns problemas:

— impossibilidade de identificar um objecto visualizado (com efeito, o conteúdo da memória imagem não apresenta a estruturação necessária),

— impossibilidade de tomar em linha de conta a noção de textura, devido à organização interna que exibe,

— necessidade de recalcular periodicamente as prioridades estáticas se vários objectos se deslocam livremente no espaço da cena.

Este trabalho abre contudo uma via de investigação muito interessante, ao considerar a síntese das informações geométricas e de iluminação ao nível do processador gráfico.

As características mais importantes de uma versão futura, GSI 2, são já apontadas:

— 5 000 faces visualizadáveis em tempo real entre 10 ×× 4 a 10 ×× 6 consideradas na descrição,

— imagens coloridas com uma resolução 1024 × 1024,

— em relação ao sistema GSI 1, custo muito mais elevado.

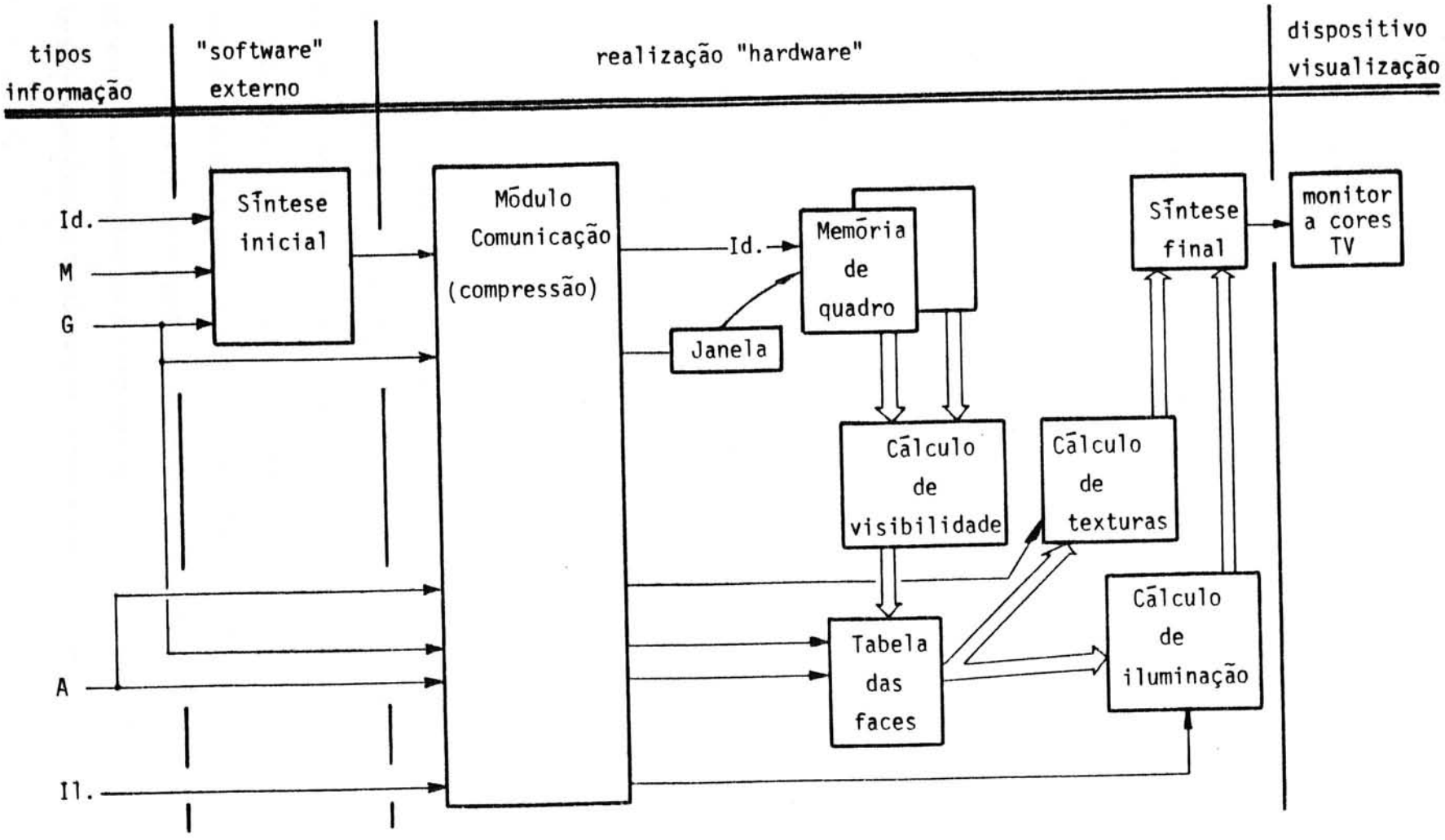


Fig. 2.22 – Esquema simplificado da estrutura de HELIOS.

II. 5. UMA APRESENTAÇÃO GERAL DE HELIOS

HELIOS é o passo lógico, ao nível do "hardware", da nossa ideia de base que consiste em manter separado o tratamento dos diferentes tipos de informação até ao "software" de visualização e, na medida do possível, até ao processador gráfico [cf. I. 3.2.3].

Sem desejarmos pormenorizar o seu princípio de funcionamento (para tal ver o Capítulo III), vamos indicar as respectivas características principais, com o intuito de o podermos comparar com os sistemas anteriormente analisados.

II. 5.1 — **Características principais**

No estado actual do protótipo, as informações morfológicas, geométricas e de identificação são manipuladas por "software", tendo como objectivo fundamental o preenchimento de uma memória de quadro multi-plano [fig. 2.22]. Devemos contudo salientar que, contrariamente à solução clássica, as informações de identificação, de aspecto, de iluminação e algumas informações geométricas são transmitidas ao processador gráfico.

A memória de quadro é composta de:

— vários *Planos de Identificação* (2 no protótipo actual) que memorizam, para cada ponto de uma imagem de resolução 512 × 512, o número de face à qual ele pertence. Esta informação é suficientemente estruturada para possibilitar a identificação de uma face, logo que se designe um dos seus pontos sobre o ecran, com a ajuda de um cursor; esta característica justifica a designação que se adoptou para estes planos — Plano de Identificação —.

— um *"Plano" de Fundo* que se distingue dos precedentes pelo facto da sua capacidade se reduzir à unidade. Este plano é representado por um único registo que contém o número de face correspondente ao fundo desejado, o qual afecta todo o ecran.

Saliente-se que os planos que constituem a memória de quadro são visualizados segundo uma prioridade pré-estabelecida (ao plano de fundo será sempre associada a prioridade mínima) como se verá brevemente [cf. III. 2.1.4].

As informações de aspecto e de iluminação e parte das informações geométricas são consideradas pelo processador gráfico, ao ritmo do sinal video (em tempo real).

A imagem final, que apresenta uma resolução 512 × 512, pode conter simultaneamente 2 ×× 15 cores.

II. 5.2 — **Outras características**

A entidade de base manipulada directamente pelo processador gráfico é a face no espaço a três dimensões (3D), suposta plana; podem ser sintetizadas simultaneamente 1024 faces diferentes.

A cada plano de identificação está associado uma *Tabela de Visibilidade* que contém, para cada face, um *indicador de visibilidade* no plano em causa. Assim, uma mesma face pode ser visível num plano e invisível num outro.

Uma *Tabela de Faces* associa a cada uma delas:

— um referencial 3D,

— uma textura que pode tomar diferentes dimensões,

— um brilho que caracterize o material a simular.

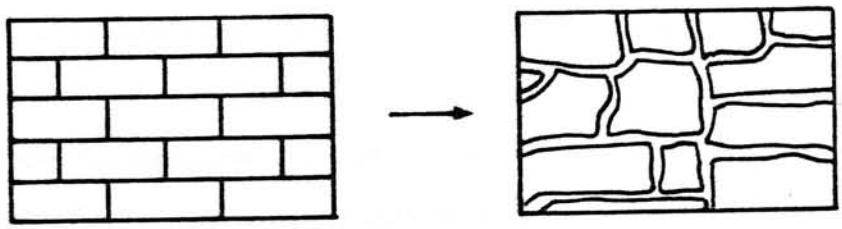


Fig. 2.23 – Um muro de tijolo "transforma-se" instantaneamente num muro de pedra.

A manipulação das informações de aspecto é possível graças:

— à memorização de texturas coloridas, que podem tomar diferentes dimensões, numa memória gerida pelo processador gráfico,

— à atribuição instantânea de uma textura memorizada a qualquer face [fig. 2.23],

— ao cálculo da projecção de texturas sobre as faces em função do referencial 3D associado [fig. 2.24].

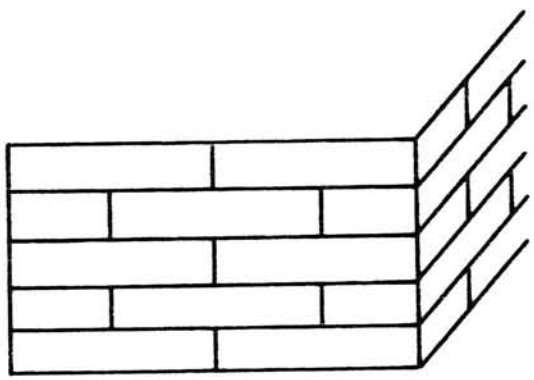


Fig. 2.24 – Uma única textura é necessária por cada material visualizado.

As informações de iluminação são consideradas a partir:

— de uma fonte luminosa suposta situada numa esfera de raio infinito, definida pela sua posição no espaço (2048 direcções possíveis) e pela cor que emite (2 ×× 12 cores possíveis),

— da intensidade de uma luz ambiente difusa (16 valores de intensidade),

— do nível de brilho (4 níveis) do material associado à face.

Todos os parâmetros indicados e relacionados com as informações de aspecto e de iluminação podem evoluir em tempo real.

O processador gráfico exibe ainda outras características igualmente interessantes, entre as quais distinguimos:

— definição de uma "janela" por plano de identificação, independentes umas das outras,

— modificação instantânea do indicador de visibilidade das faces,

— cálculo de visibilidade em função da profundidade dos planos e do indicador de visibilidade das faces [fig. 2.25].

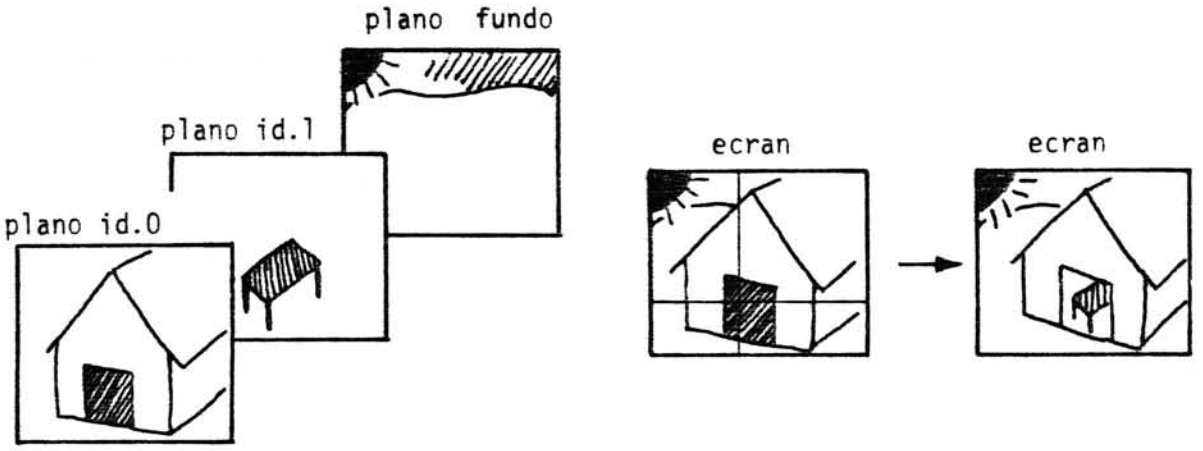


Fig. 2.25 – A face "porta" torna-se invisível quando é apontada sobre o ecran.

II. 5.3 — **As facilidades de animação**

A separação prolongada das informações gráficas possibilita a produção de animação em tempo real.

Uma animação qualquer pode-se decompor em quatro operações independentes:

— modificação das informações morfológicas,

— modificação das informações geométricas,

— modificação das informações de aspecto,

— modificação das informações de iluminação.

Considerando que HELIOS efectua a síntese, em tempo real, das informações de aspecto, de iluminação e de parte das informações geométricas, o único problema que nos resta relaciona-se com a modificação do conteúdo da memória de quadro. Ora, estas informações desprovidas de todas as indicações de ordem visual podem ser fortemente compactadas. Um *módulo de comunicação* "hardware", que realiza as trocas de informação com o exterior, autoriza uma compressão cujo nível pode atingir 255 e assegura desta maneira a modificação completa de um plano de identificação no intervalo de tempo a que corresponde uma imagem (40 ms). Evidentemente, estas facilidades só podem ser aproveitadas ao máximo, desde que exista um módulo "hardware" relacionado com o cálculo de visibilidade.

II. 5.4 — **O posto de trabalho**

O desenvolvimento do protótipo HELIOS efectuou-se no seio da configuração esquematizada na figura 2.26.

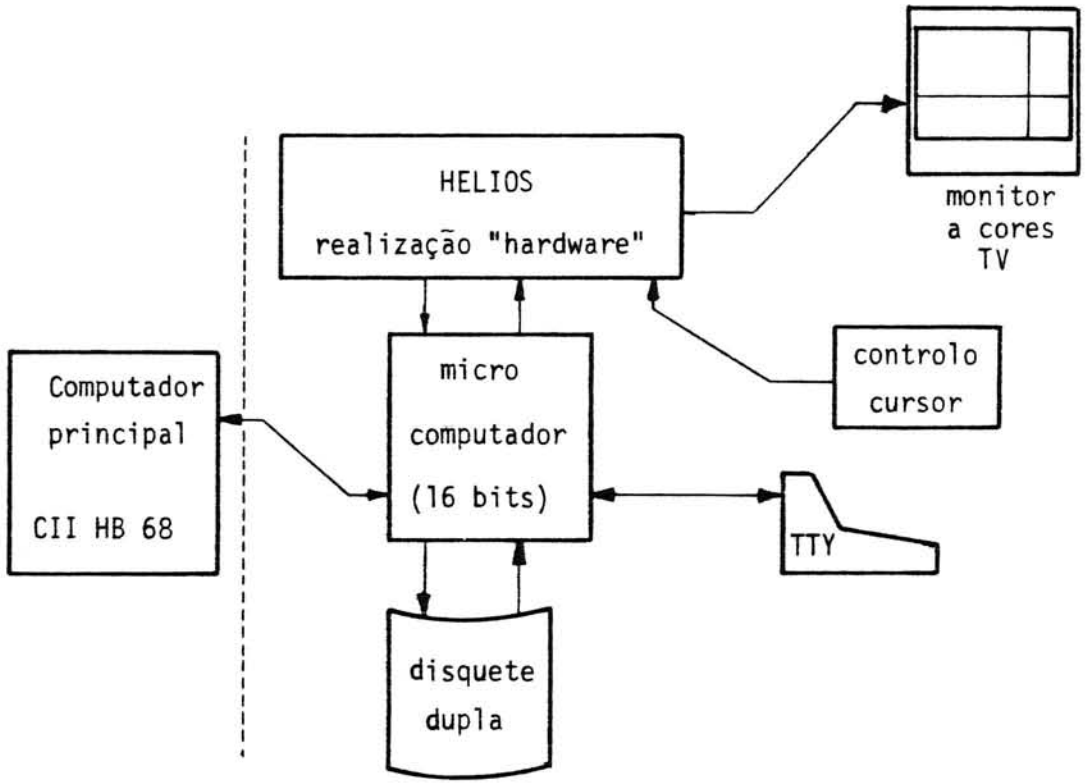


Fig. 2.26 – A configuração actual.

Este posto de trabalho integra como dispositivos de comunicação:

— Um monitor a cores de varrimento do tipo TV, que admite uma resolução 512 × 512 e apresenta como entradas independentes R, G, B e sincronização,

— um teclado alfanumérico para a recolha de dados e para os comandos locais do sistema,

— um cursor digital para a recolha de coordenadas e para a designação de objectos visualizados.

O deslocamento deste cursor, no espaço imagem 512 × 512, é normalmente controlado por um conjunto de quatro teclas (cima, baixo, esquerda e direita, com auto-repetição), mas outros dispositivos de comunicação são igualmente admissíveis como, por exemplo, uma mesa de digitalização,

— entradas analógicas podem também ser incorporadas para a modificação de parâmetros locais do sistema. Por exemplo, intensidade da luz ambiente, direcção da fonte luminosa, ...

II. 6 — **CONCLUSÃO**

Uma primeira nota relaciona-se com a metodologia utilizada na análise de alguns sistemas existentes. De facto, ela mostrou-se como um utensílio eficaz, pondo em evidência as características realmente importantes dos sistemas estudados.

Foi possível analisar vários sistemas existentes, os mais significativos em nossa opinião, e dar uma ideia das possibilidades e das originalidades da estrutura de HELIOS.

Da análise feita deduzimos que nenhum sistema, com excepção de HELIOS, oferece uma realização "hardware" que integre simultaneamente os conceitos de textura colorida e iluminação de uma cena. Esta característica permite, sem intervenção de uma potência de cálculo importante (utilizámos um microcomputador de 16 bits, mas com um de 8 bits seria igualmente viável), modificar em tempo real as informações de aspecto, de iluminação e geométricas e de efectuar a identificação de faces, directamente sobre o ecran.

No estado de desenvolvimento actual, a possibilidade de manipular a informação compactada ao nível do módulo de comunicação, permite minimizar o problema relacionado com a modificação do conteúdo dos planos de identificação. Ao mesmo tempo fica aberta a hipótese de integrar no sistema um módulo "hardware", considerando o cálculo de visibilidade.

CAPÍTULO III

HELIOS VISTO DO UTILIZADOR

CAPÍTULO III: HELIOS VISTO DO UTILIZADOR

III. 1. APRESENTAÇÃO E PRINCÍPIO DE FUNCIONAMENTO

III. 1.1 – Relações software / hardware

Num sistema informático podemos sempre distinguir uma parte "hardware" (a máquina propriamente dita), em geral, com uma capacidade para executar um conjunto de instruções de baixo nível e o "software" de exploração ou simplesmente sistema [fig. 3.1].



Fig. 3.1 – Esquema muito simplificado de um sistema informático.

Com o "software" de exploração pretende-se fornecer um meio eficaz e natural para utilizar as possibilidades da parte "hardware", muitas vezes de acordo com uma certa gama de aplicações. Por este motivo, os potenciais utilizadores terão normalmente uma ideia bastante deformada daquele conjunto de possibilidades.

Pelo contrário, o responsável pela concepção do "software" de exploração deverá conhecer perfeitamente o princípio de funcionamento da máquina e, em particular, os seus meios de comunicação com o exterior.

Se se considerar o terminal gráfico HELIOS como um caso particular de um sistema informático, distingue-se nele igualmente uma parte "hardware" e uma parte "software", cuja camada mais interna designámos por "software" de visualização [cf. I. 3.2] [fig. 3.2].

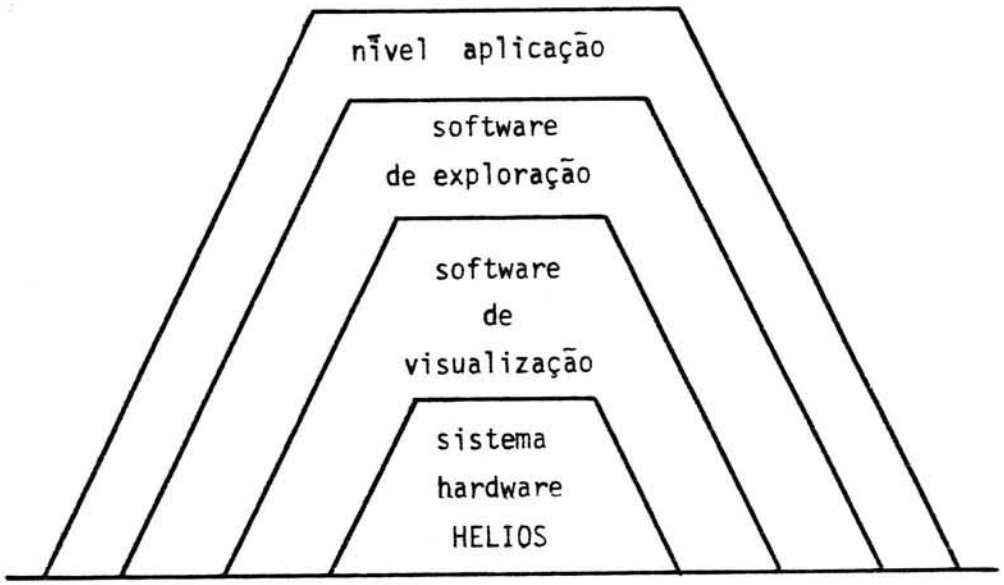


Fig. 3.2 – HELIOS visto como um sistema informático.

Este capítulo considera o conjunto de informações relacionadas com HELIOS necessárias ao responsável pela concepção do "software" de visualização. Integra, portanto, uma descrição pormenorizada do seu princípio de funcionamento e dos seus meios de comunicação com o exterior.

São igualmente evocados os princípios que devem inspirar o "software" de visualização, que permitam, simultaneamente, uma utilização fácil e eficaz das possibilidades oferecidas pelo processador gráfico.

III. 1.2 – **Princípio de funcionamento de HELIOS**

Vamos distinguir dois modos de funcionamento fundamentais de HELIOS:

— Modo refrescamento do ecran.

Neste caso, HELIOS calcula, ao ritmo do sinal video, a imagem necessária ao refrescamento de um monitor a cores de varrimento do tipo TV; os dados considerados neste cálculo são guardados em memórias e registos do terminal.

— Modo acesso exterior.

HELIOS admite acessos vindos do exterior, realizados com o intuito de modificarem ou analisarem o conteúdo das suas memórias e registos [fig. 3.3].

A existência destes dois processos, que comunicam através de memórias comuns, gera numerosas situações de conflito, em geral, resolvidas pelo *Módulo de controlo* [fig. 3.3]; este coordena os tempos de acesso a estas memórias.

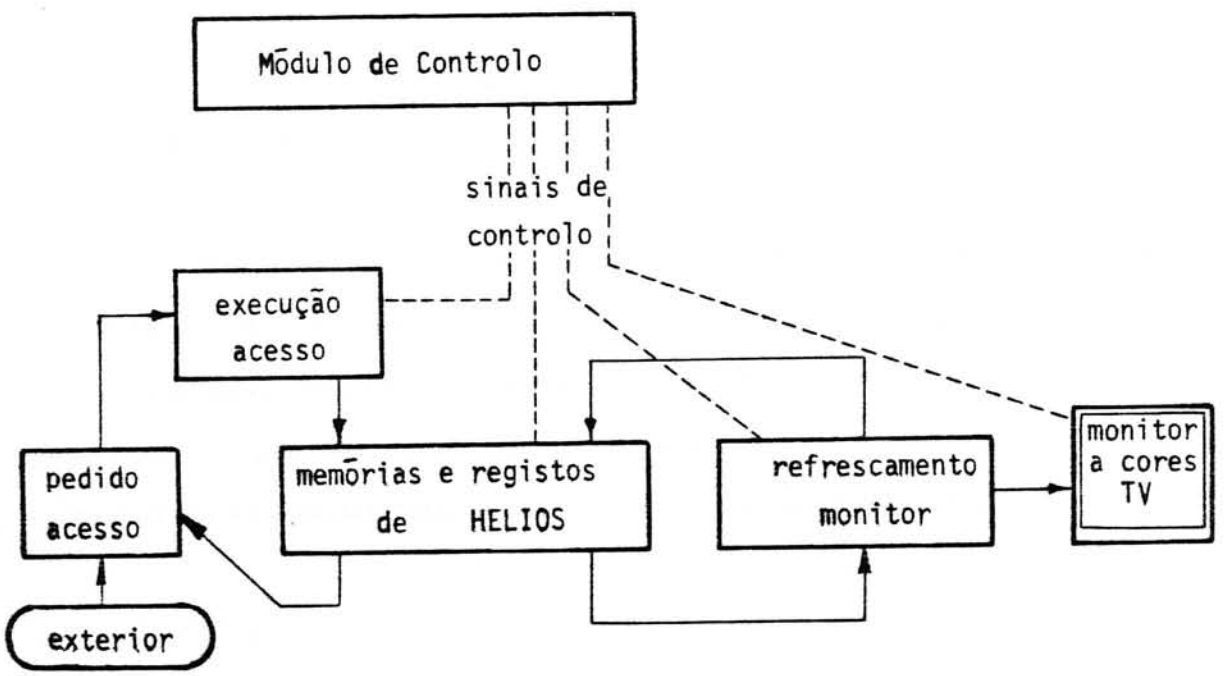


Fig. 3.3 – Os modos de funcionamento de HELIOS.

O modo acesso exterior é descrito em (III. 4). Far-se-á, seguidamente, a descrição do funcionamento em modo refrescamento do ecran, tomando como base a estrutura representada no esquema da figura 3.4.

Numa primeira fase, consideramos de forma resumida o princípio de funcionamento de HELIOS e, seguidamente, pormenorizaremos o funcionamento de cada um dos seus módulos:

— A memória de quadro [cf. III. 2]

— O tratamento exercido sobre a informação desde a saída da memória de quadro até ao monitor a cores [cf. III. 3]

- a tabela das faces,
- o módulo das texturas,
- o cálculo de iluminação,
- o módulo de visualização.

III. 1.3 – **Caminho percorrido pela informação em modo de refrescamento do ecran**

Uma *Memória de Quadro, multi-plano* fornece, depois de um cálculo de visibilidade, um número de face (NFACE) por cada ponto de uma imagem 512 × 512 [fig. 3.4].

Este NFACE, utilizado como endereço de uma memória muito rápida, a *Tabela das Faces,* permite obter a informação relativa à posição da face no espaço 3D (referencial 3D) e ao tipo de material que lhe está associado (apontador de uma textura colorida e um nível de brilho).

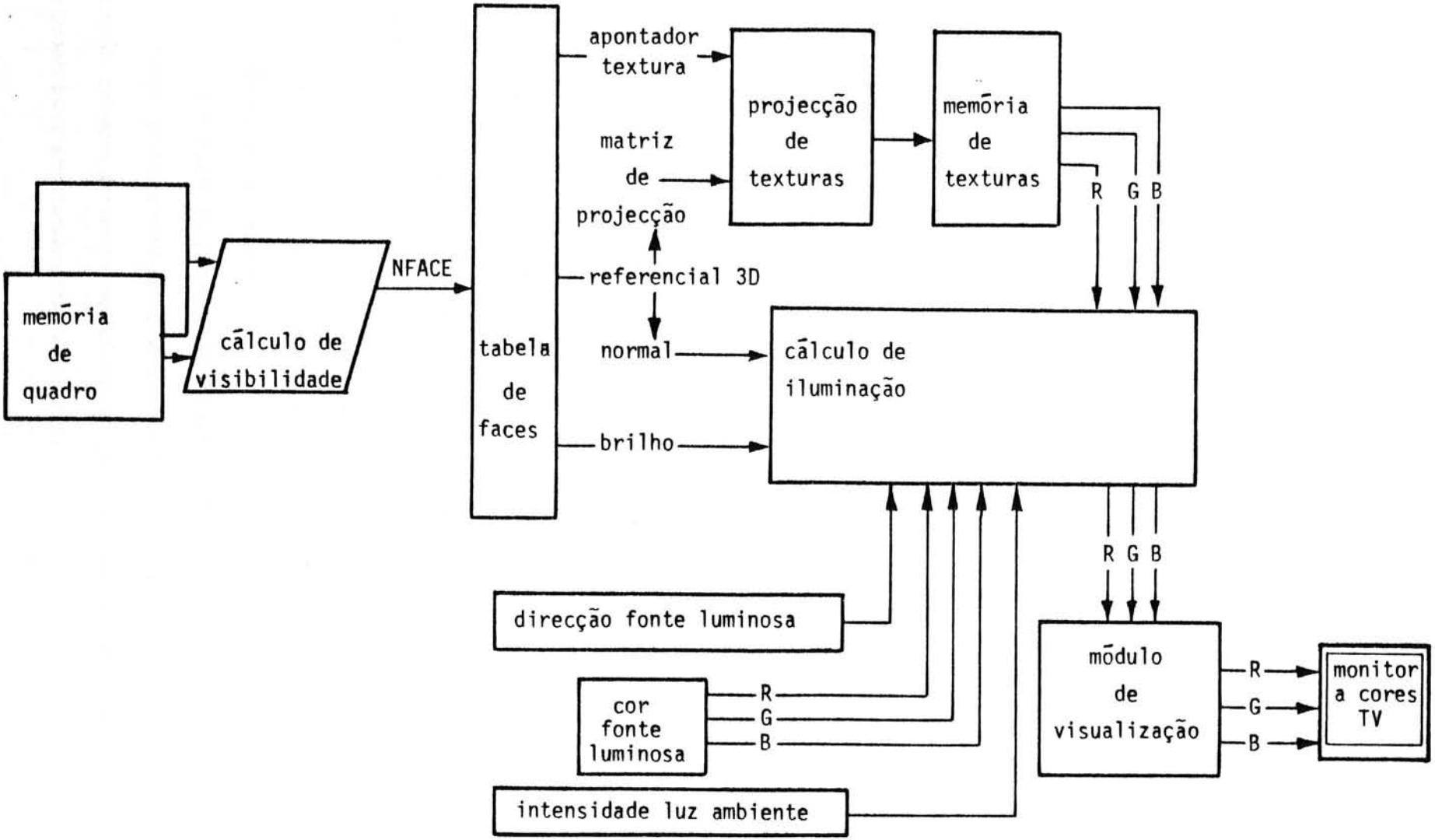


Fig. 3.4 – A estrutural geral de HELIOS.

A partir deste momento dois processos paralelos são postos em funcionamento:

— um primeiro calcula um endereço da Memória de texturas, tomando em linha de conta o apontador de uma textura e o referencial 3D da face (fig. 3.5); o resultado final, depois de uma leitura daquela memória, corresponde à cor associada ao ponto em questão, independentemente das condições de iluminação.

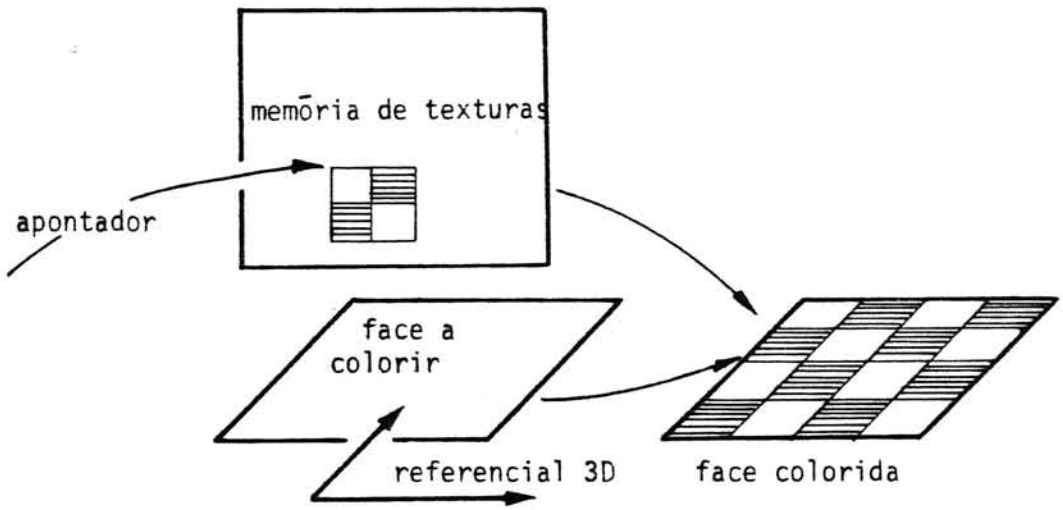


Fig. 3.5 – As informações consideradas no cálculo de texturas.

— um segundo considera a influência do sistema de iluminação da cena e toma em linha de conta:

- a normal à face e o brilho respectivo,
- uma fonte luminosa colorida, colocada no infinito, definida pela sua direcção no espaço 3D e a cor que emite,
- a intensidade da luz ambiente difusa [fig. 3.6].

O resultado deste cálculo afectará de maneira adequada o resultado do processo anterior, originando imagens, em tempo real (25 imagens por segundo), com uma resolução 512 × 512, que podem exibir simultaneamente 2××15 cores.

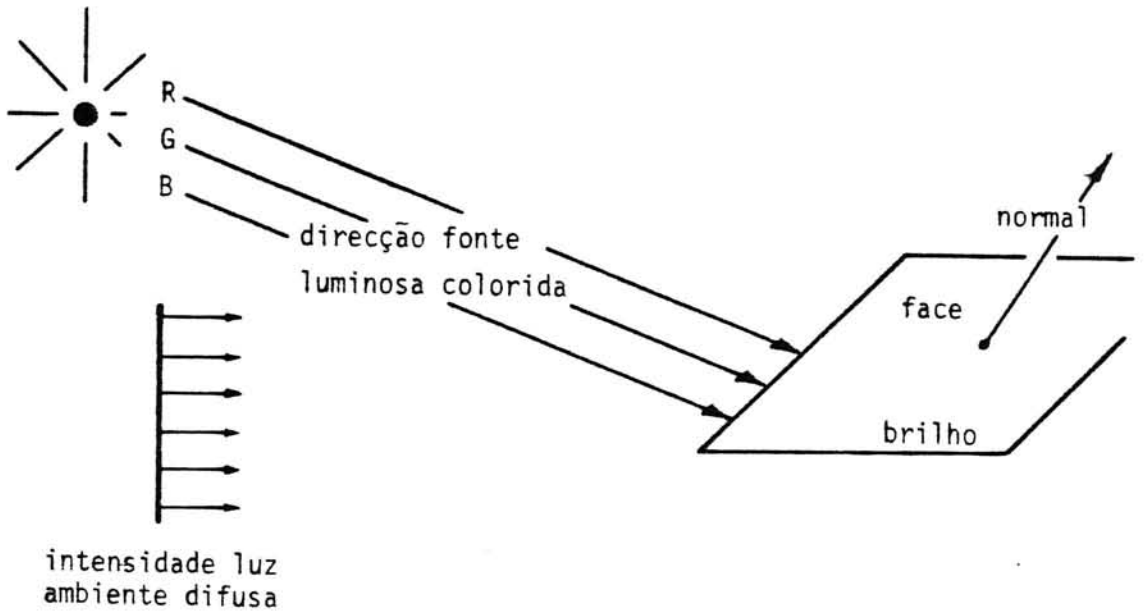


Fig. 3.6 – As informações consideradas no cálculo de iluminação.

III. 2. MODELIZAÇÃO DAS INFORMAÇÕES PRE-SINTETIZADAS

No estado actual do protótipo estas informações são obtidas por "software". Elas representam uma primeira síntese das informações morfológicas, de identificação e algumas informações geométricas e são guardadas na memória de quadro.

A memória de quadro é composta de vários *Planos de Identificação* (dois no protótipo actual) e um *Plano de Fundo.*

III. 2.1 – **Os planos de identificação**

III. 2.1.1 – **Natureza das informações memorizadas**

Estes planos contêm as informações de identificação que definem a imagem a visualizar [fig. 3.7].

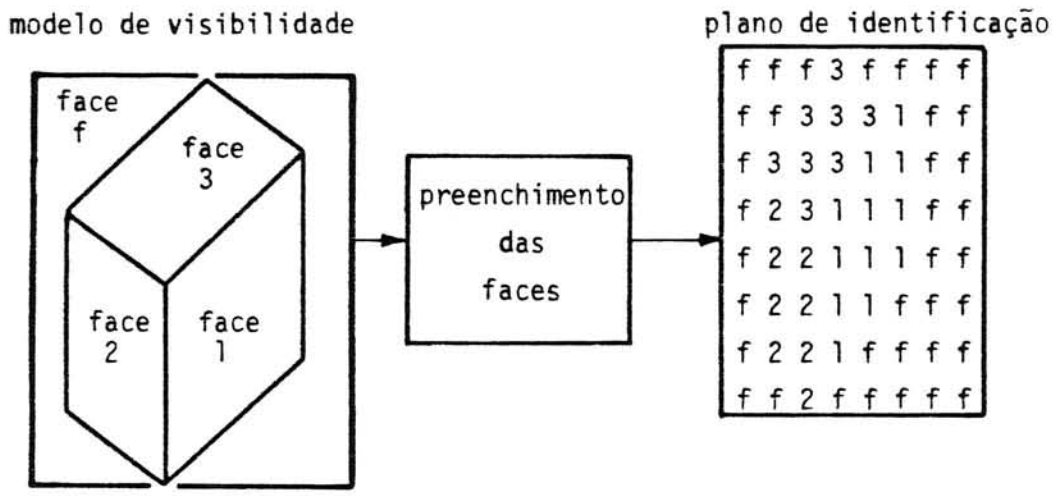


Fig. 3.7 – O conteúdo de um plano de identificação.

O nível de estruturação da informação neste plano é suficiente para permitir uma fácil identificação das entidades visualizadas, bastando aponta-las com a ajuda de um cursor; uma leitura do plano em questão (no endereço referenciado pelo cursor) fornece um número de identificação de face (NFACE) codificado em 10 bits. Desta maneira é sempre possível distinguir até 1024 faces diferentes.

III. 2.1.2 – **A capacidade dos planos**

O plano de identificação é constantemente "varrido", linha à linha, de alto a baixo e da esquerda para a direita, originando a informação que, submetida a um tratamento adequado, vai permitir o refrescamento do ecran, ponto a ponto [fig. 3.8]. Assim, a capacidade dos planos de identificação deve ser suficiente para armazenar um número de face (NFACE) por cada ponto da imagem, isto é, 512 × 512 × 10 bits.

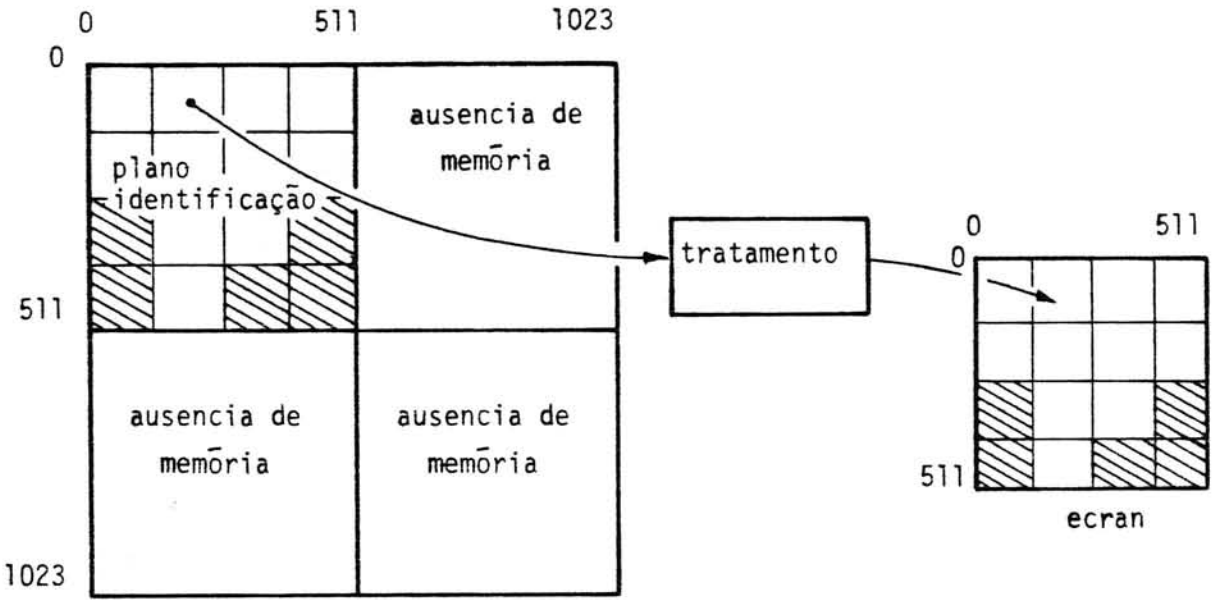


Fig. 3.8 – O varrimento do plano de identificação.

III. 2.1.3 – **A função JANELA**

O varrimento do plano de identificação faz-se no interior do espaço 1024 × 1024, em sincronismo com o varrimento de um monitor video e, em geral, a partir do ponto superior-esquerdo (X = 0 e Y = 0 na figura 3.8).

A modificação do ponto de partida, independentemente em cada plano, permite definir uma *JANELA* 512 × 512 de visualização, cujo canto superior-esquerdo coincide com aquele ponto de partida [fig. 3.9].

O espaço de varrimento exterior ao espaço 512 × 512 corresponde a uma zona de "ausência de memória" e é considerado totalmente preenchido pelo número de face 1023-i, onde i representa o número de ordem do plano de identificação respectivo.

Para o plano de identificação i, se a face 1023-i é feita invisível, a imagem deste plano relativa ao espaço de varrimento exterior a 512 × 512 é também invisível, não contribuindo para

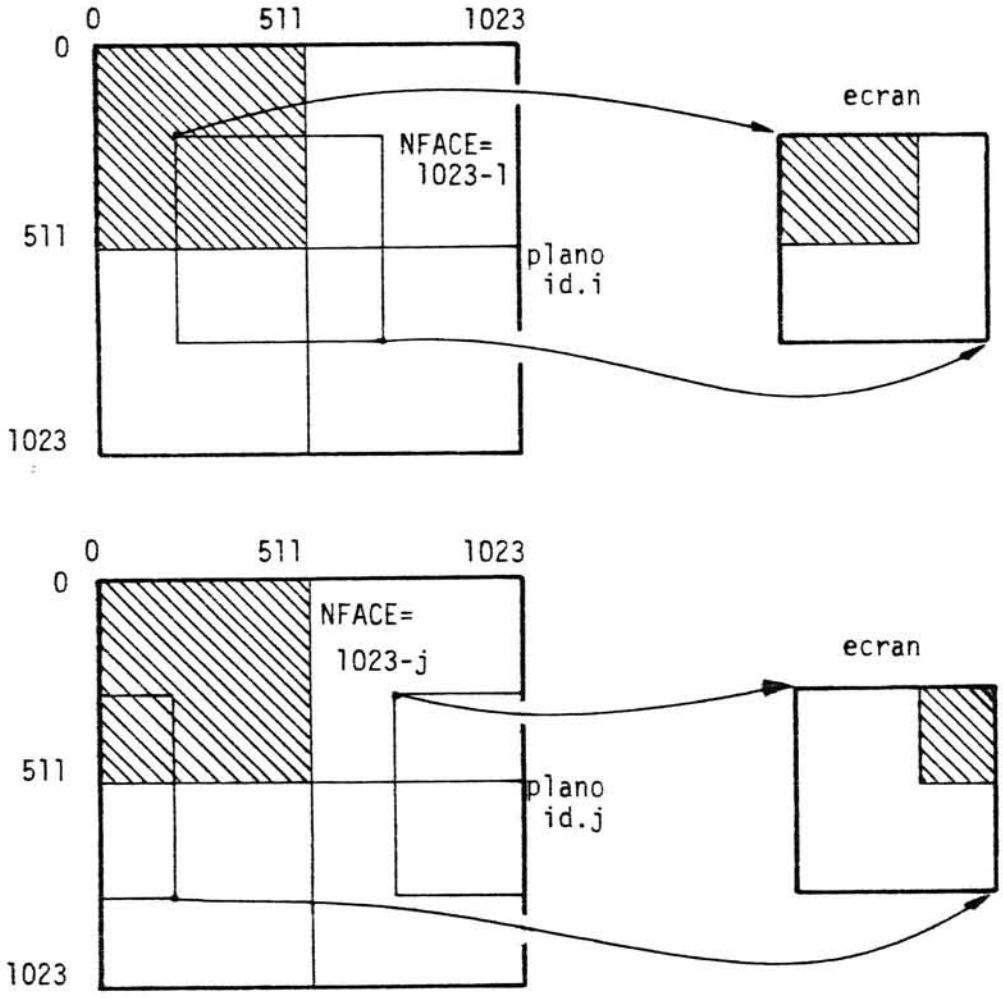


Fig. 3.9 – A definição de uma Janela de visualização no espaço 1024 × 1024.

a formação da imagem. Assim, por exemplo, para o plano de identificação zero, se a face 1023 é invisível, os pontos de partida A, B ou C [fig. 3.10] fazem com que este plano deixe aparecer os planos seguintes [cf. III. 2.1.4], através de uma única operação – mudança do ponto de partida do varrimento do plano de identificação –.

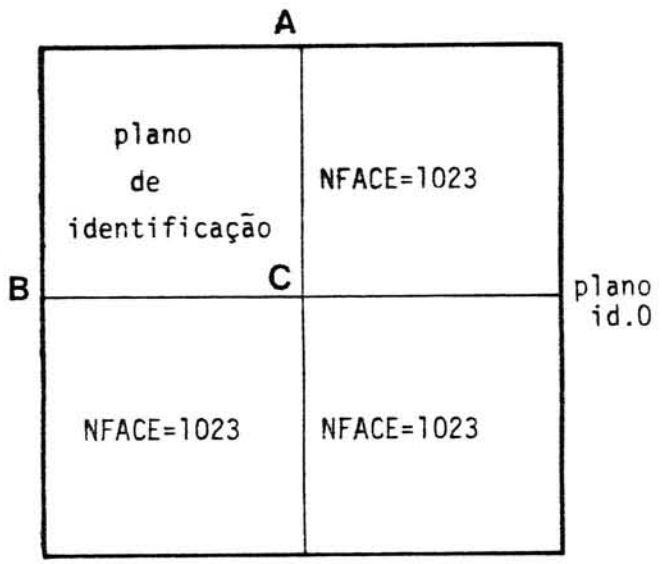


Fig. 3.10 – Os pontos de partida A, B ou C permitem uma invisibilidade total do plano de identificação em que estão inseridos.

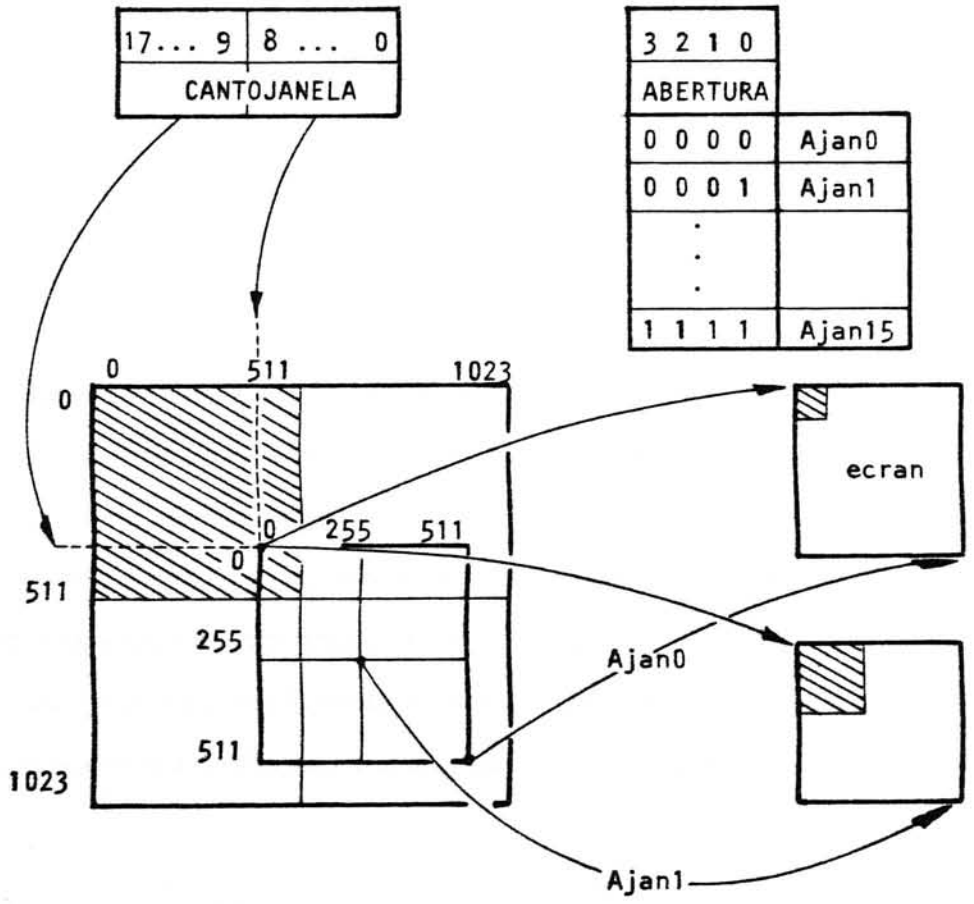


Fig. 3.11 – A definição de uma Janela.

HELIOS permite definir uma Janela por plano de identificação, graças a dois registos que lhes estão associados:

— registo CANTOJANELA que define, no espaço de varrimento 1024 × 1024, o canto superior-esquerdo de uma janela de lados paralelos aos bordos do ecran [fig. 3.11].

— registo ABERTURA que exprime a abertura de uma janela, Ajan, codificada por 4 bits [fig. 3.11].

Se a = Ajan + 1, onde o <= Ajan <= 15, a abertura da janela será.

$$(512/a) \times (512/a)$$

Saliente-se que a função Janela permite ainda simular translações independentes sobre os diferentes planos, assim como 16 níveis de efeitos de lupa, por passos unitários.

III. 2.1.4 – **O cálculo de visibilidade**

A existência de vários planos de identificação implica a escolha de um critério que defina, entre eles, uma certa prioridade de visualização. A implementação deste critério, integrada num módulo que designaremos genericamente por "Cálculo de visibilidade", torna-se relativamente complexa se se considerar o intervalo de tempo em que deverá operar; aquele cálculo é realizado para todos os pontos de uma imagem 512 × 512, 25 vezes por segundo.

Os planos que constituiem a memória de quadro são ordenados segundo um número de ordem, ao qual é associado uma certa prioridade de visualização [fig. 3.12].

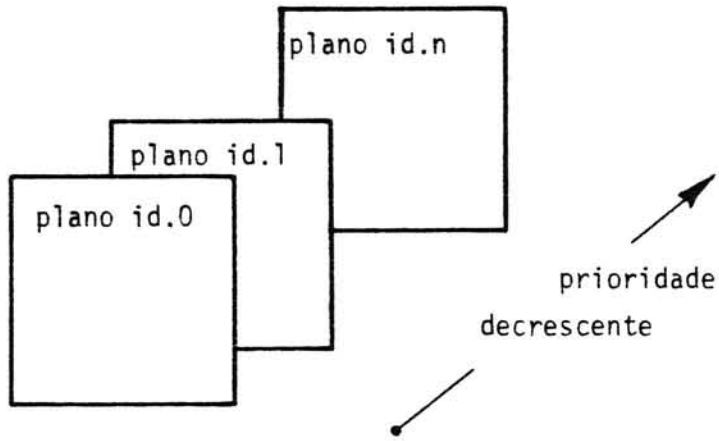


Fig. 3.12 – A prioridade de visualização dos planos.

Em princípio, o plano de identificação de ordem 0 é visualizado, pois é a ele que está associada a prioridade de grau mais elevado. Este plano pode tornar-se total ou parcialmente invisível graças à função janela [cf. III 2.1.3].

Todavia, generalizámos o cálculo de visibilidade ao nível da entidade face. Cada plano possui uma *Tabela de Visibilidade de Faces* [fig. 3.13] de capacidade igual ao número de faces possíveis (1024 × 1), contendo o *indicador de visibilidade* de cada uma delas (1 significa visível e 0 invisível).

Desta maneira, o cálculo de visibilidade resume-se à procura, de plano em plano, segundo as prioridades de visualização decrescentes, para cada ponto da imagem, do primeiro NFACE tendo o respectivo indicador de visibilidade igual a 1 [fig. 3.13].

Como a Tabela de Visibilidade das faces é acessível do exterior [cf. III. 4.5.3], o utilizador pode modificar em tempo real o atributo de visibilidade de qualquer face.

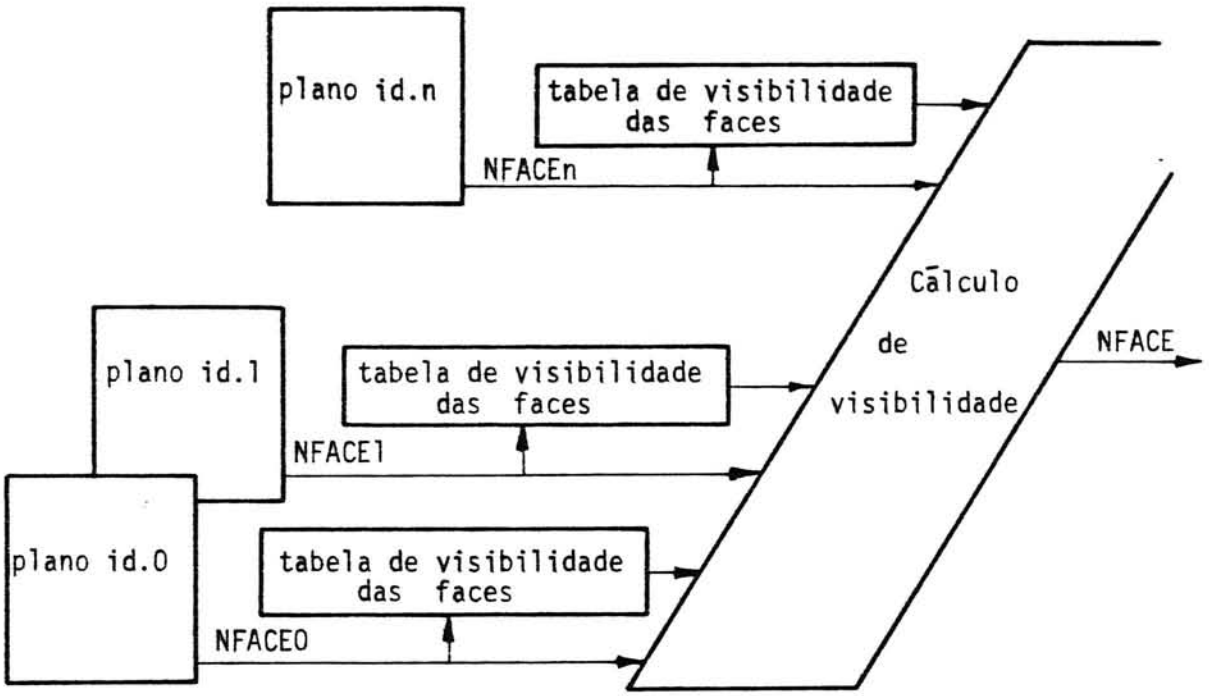


Fig. 3.13 – O cálculo de visibilidade ao nível dos planos de identificação.

III. 2.2 – **O plano de fundo**

As funções mencionadas para os planos de identificação são igualmente válidas para o Plano de Fundo. Contudo, algumas diferenças devem ser postas em evidência:

— o seu conteúdo

O "Plano" de Fundo regista um único número de face (NFACE), o qual afectará todos os pontos da imagem.

— a sua prioridade de visualização

O Plano de Fundo vem sempre classificado depois dos planos de identificação no que respeita à prioridade de visualização.

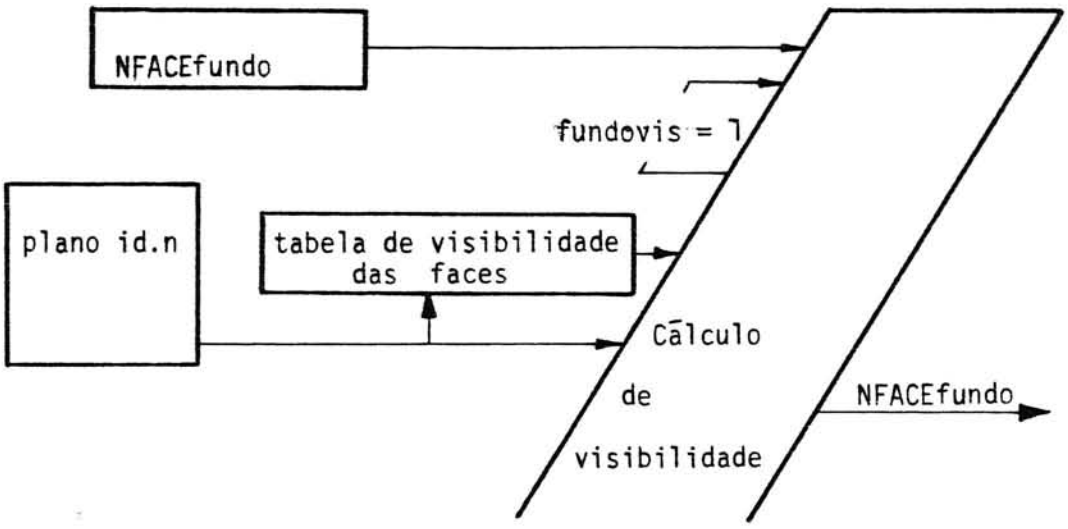


Fig. 3.14 – O cálculo de visibilidade ao nível do plano de fundo.

— o Plano de Fundo não dispõe de Tabela de Visibilidade das Faces.

Com efeito, é necessário garantir a existência de uma face a visualizar que, em último recurso, será a correspondente ao Plano de Fundo [fig. 3.14].

A grande vantagem do Plano de Fundo reside na facilidade de se poder afectar uma textura a toda a imagem, registando um único número de face (NFACE fundo);

Além disso, é possível uma modificação instantânea do fundo visualizado, pela simples modificação do registo daquele número de face.

III. 3 AS ETAPAS DA SINTESE INTERNA

As etapas da síntese interna põem em evidência dois processos paralelos (cálculo das texturas e cálculo de iluminação), que exibem duas etapas comuns: uma inicial, a *Tabela das Faces* e uma final, o *Módulo de Visualização* [fig. 3.4].

III. 3.1 – **A tabela das faces**

A Tabela das Faces contém, para cada uma das 1024 faces possíveis, uma certa informação que define [fig. 3.15]:

— o seu material

- indicador do nível do brilho
- apontador de uma textura colorida

— o seu referencial no espaço 3D

- vector normal
- matriz de projecção

Saliente-se que, sendo a Tabela das Faces acessível do exterior [cf. III. 4.5.3], será possível alterar instantaneamente as informações visuais de uma face, modificando o conteúdo do registo correspondente nesta tabela.

Por exemplo, pode-se modificar o aspecto colorido de uma face escolhendo (apontando) outra textura.

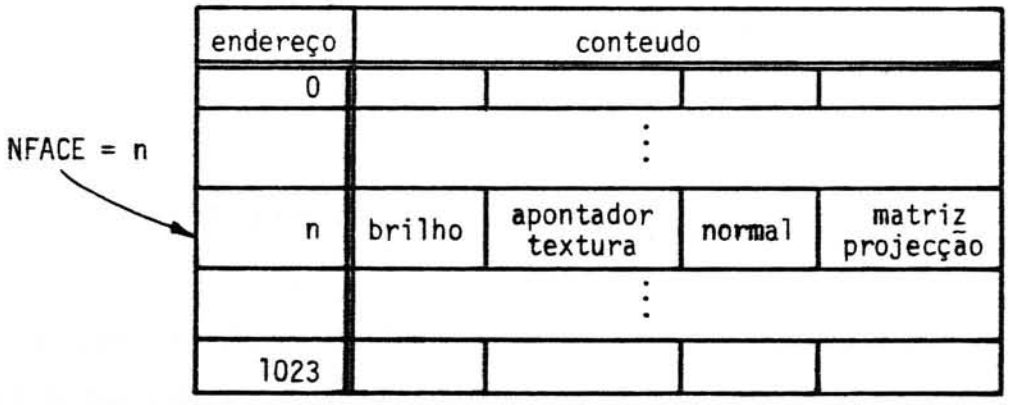


Fig. 3.15 – O conteúdo da Tabela das Faces.

III. 3.1.1 – **O Brilho**

A codificação do brilho de um material faz-se por intermédio de dois bits, admitindo-se quatro níveis: mate, semi-mate, acetinado e brilhante [fig. 3.16].

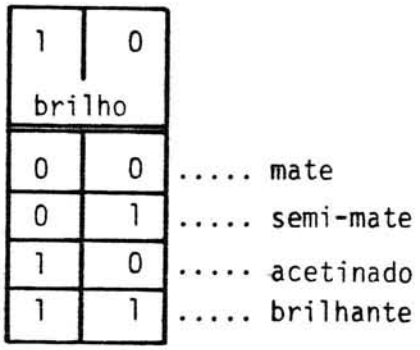


Fig. 3.16 – A codificação dos níveis de brilho.

O sistema permite ainda outros tipos de codificação, igualmente em dois bits, desde que se lhe transmita módulos de brilho especiais durante a inicialização do sistema [cf. III. 3.3].

III. 3.1.2 – **O apontador de textura**

O modelo de uma textura colorida é traduzido por uma matriz quadrada guardada na Memória de Texturas, cujo conteúdo reflete de alguma forma o aspecto colorido dos materiais simulados, através de uma ou mais cores [cf. III. 3.2.1]. Texturas de dimensões diferentes residem naquela memória e podem ser visualizadas simultaneamente.

O apontador (P9, ..., P0) define, ao mesmo tempo, a posição inicial e a dimensão da textura que se quer afectar a uma face [fig. 3.17].

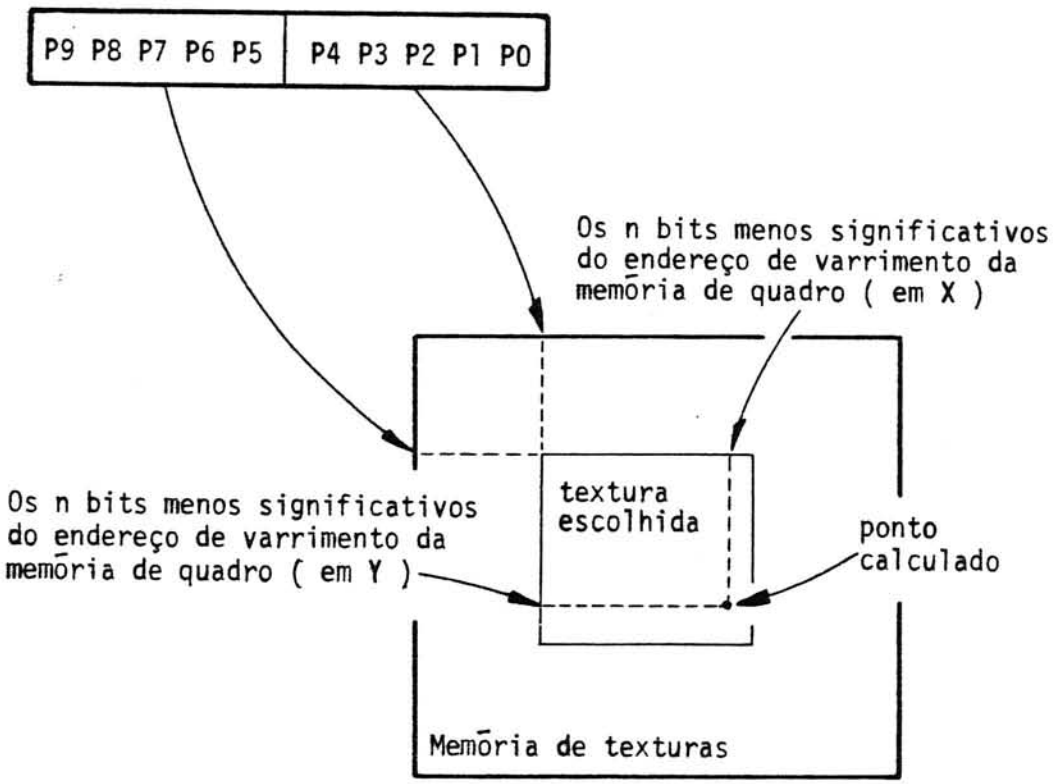


Fig. 3.17 – A dimensão da textura apontada é (2××n) × (2×× n).

A partir do apontador obtém-se o endereço de um ponto da memória de texturas, concatenando-o com os n bits menos significativos (em linha e em coluna) do endereço de varrimento da memória de quadro [fig. 3.17]; este procedimento é, evidentemente, transparente para o utilizador.

A dimensão da textura em questão, definida implicitamente no apontador respectivo, é determinada pelo algoritmo:

if P5 *or* P0 *then* dimensão 16 × 16 *else*

if P6 *or* P1 *then* dimensão 32 × 32 *else*

if P7 *or* P2 *then* dimensão 64 × 64 *else*

if P8 *or* P3 *then* dimensão 128 × 128 *else*

if P9 *or* P4 *then* dimensão 256 × 256 *else*

dimensão 512 × 512

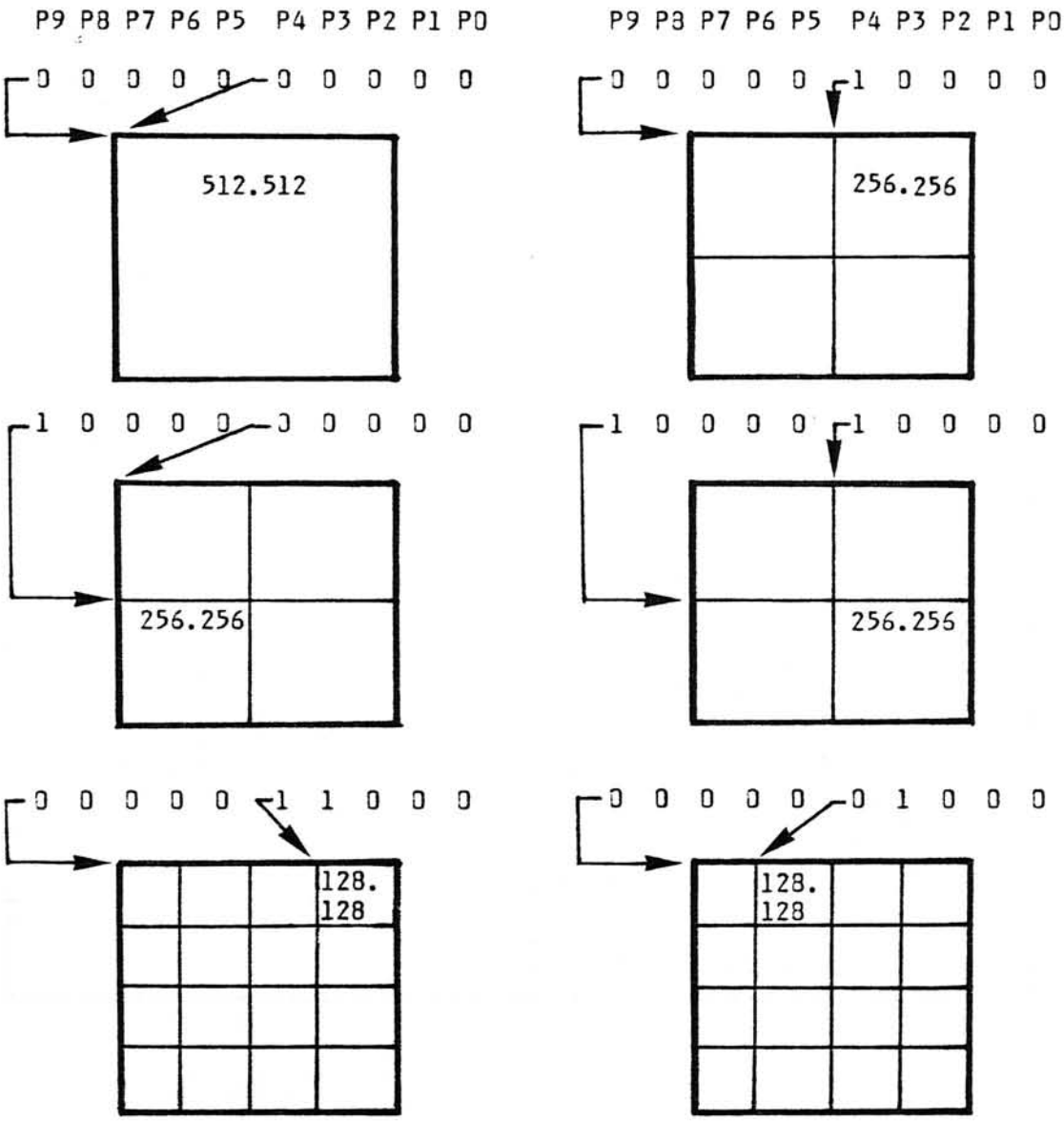


Fig. 3.18 – A escolha de uma textura.

Alguns exemplos, que consideram a existência de uma memória de texturas de capacidade 512 × 512 × cor, são ilustrados pela figura 3.18.

Um exemplo, que representa uma das numerosas configurações possíveis da memória de textura [fig. 3.19], põe em evidência a flexibilidade e eficiência desta técnica de endereçamento.

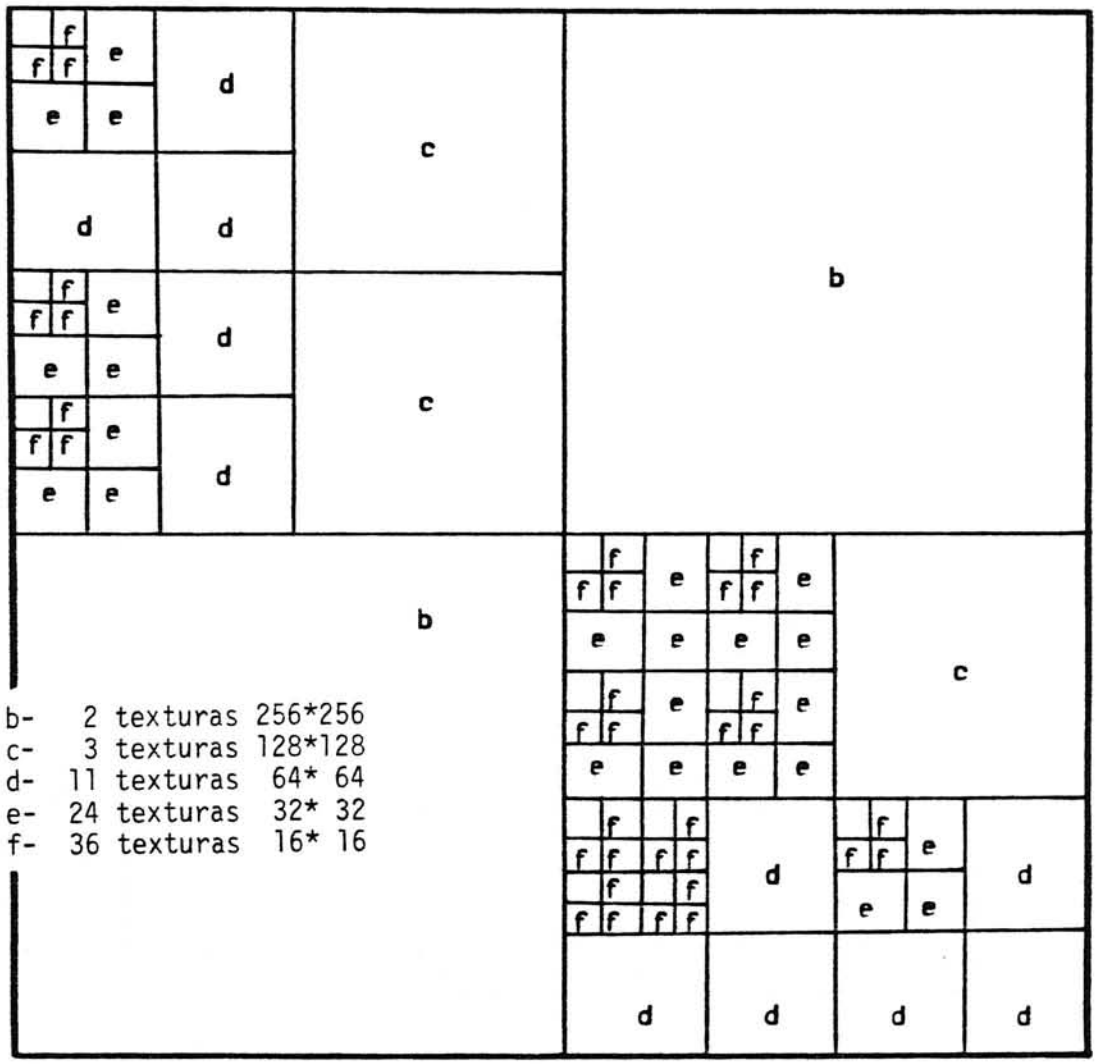


Fig. 3.19 – Uma configuração possível da memória de texturas.

Outras configurações são evidentemente possíveis; vejamos três exemplos:

1 – 768 texturas 16 × 16

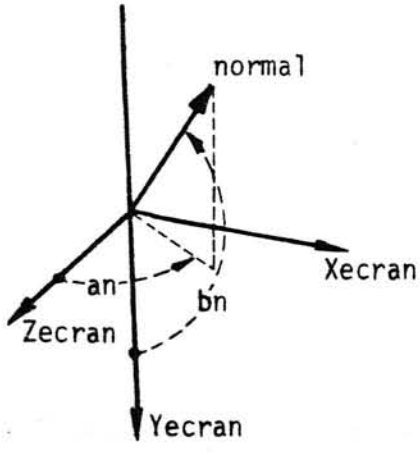
2 – 192 texturas 16 × 16 e 192 texturas 32 × 32

3 – 48 texturas 16 × 16, 48 texturas 32 × 32 e 48 texturas 64 × 64.

III. 3.1.3 – **O vector normal**

O vector normal de uma face é representado em coordenadas esféricas em relação ao referencial associado ao ecran [fig. 3.20].

Cinco bits são utilizados para representar cada um dos ângulos an e bn, o que autoriza 1024 direcções no meio-espaço situado em frente do ecran.



9	8	7	6	5	
		bn			
4	3	2	1	0	
0	0	0	0	0	0
		·			
		·			
		·			
1	1	1	1	1	≃π

4	3	2	1	0	
		an			
4	3	2	1	0	
0	0	0	0	0	-π/2
		·			
		·			
		·			
1	1	1	1	1	≃π/2

Fig. 3.20 – O vector normal.

III. 3.1.4 – A matriz de projecção

Como vimos, a projecção de uma textura sobre uma face toma em linha de conta o referencial desta última no espaço 3D. Com efeito, esta operação resume-se à determinação para cada ponto (Xe, Ye) do ecran, do ponto [Xf, Yf] correspondente no plano da face [fig. 3.21].

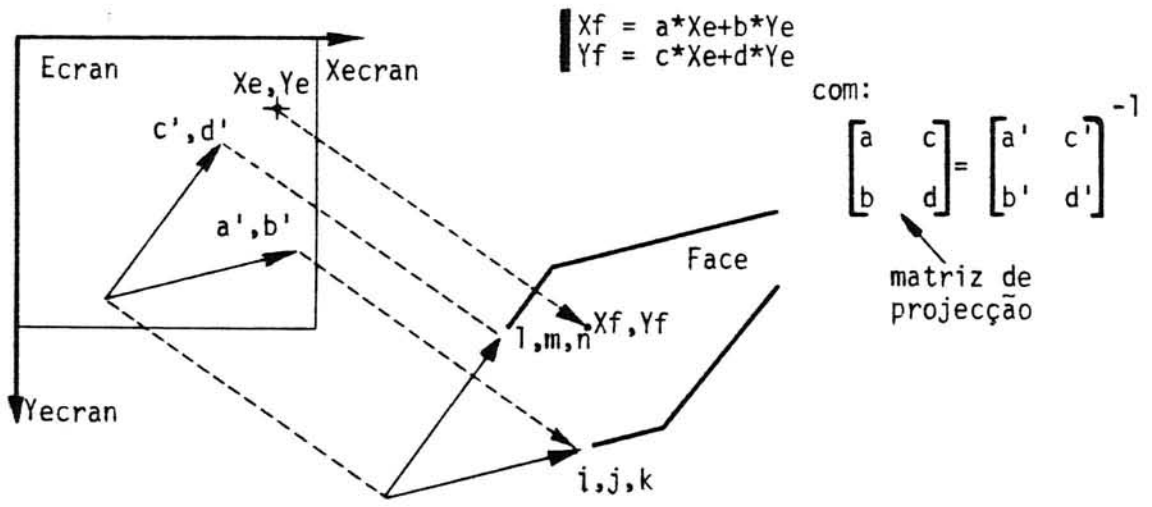


Fig. 3.21 – A projecção de uma textura.

Uma matriz de mudança de base, previamente calculada por "software", é registada na tabela das faces, para cada uma delas.

A codificação binária dos coeficientes reais a, b, c e d utiliza cinco bits para representar o valor absoluto de cada coeficiente e um bit de sinal unicamente para os coeficientes b e c; desta maneira são necessários 22 bits para representar a matriz de projecção de cada face [fig. 3.22].

	md					mc					mb					ma				
ed	3	2	1	0	ec	sc	3	2	1	0	eb	sb	3	2	1	0	ea	3	2	1	0
21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
d					c						b						a				

Fig. 3.22 – A codificação da matriz de projecção.

O valor v de cada coeficiente é expresso por:

$$v = (-1)^{sinal} \times mantissa \times (2^{(2 \times expoente - 3)})$$

e o seu valor absoluto pode variar entre 0 e 7,5.

III. 3.2 – **As Texturas coloridas**

Dois módulos "hardware" estão na base da síntese de texturas no seio de HELIOS:

— A Memória de Texturas

— A Projecção de Texturas [cf. III. 3.1.4]

III. 3.2.1 – **A memória de texturas**

A noção de textura integrada em HELIOS, tem por base uma memória de capacidade 512 × 512 × cor e comporta três originalidades importantes:

— cada ponto de uma textura exprime a cor intrínseca da face independentemente do seu referencial 3D e das condições de iluminação da cena. Para cada ponto podemos escolher uma entre 2^{12} cores possíveis [fig. 3.23].

11	10	9	8	7	6	5	4	3	2	1	0
G3	G2	G1	G0	R3	R2	R1	R0	B3	B2	B1	B0
GREEN				RED				BLUE			

Fig. 3.23 – A codificação das cores.

— cada textura é suposta definida no plano da face e não no plano do ecran. A projecção sobre o ecran é efectuada, para cada ponto visualizado, em função do referencial 3D associado à face [cf. III. 3.1.4].

— podem ser memorizadas e visualizadas simultaneamente [cf. III. 3.1.2] texturas de dimensões diferentes, 16 × 16, 32 × 32, 64 × 64, 128 × 128, 256 × 256 e 512 × 512.

Estas três propriedades em conjunto permitem considerar uma textura como uma simulação do aspecto de um certo material (madeira, tijolo, tecido, ...), independentemente da sua posição no espaço e das condições de iluminação; assim uma *única textura é necessária para todas as faces que digam respeito a um mesmo material.*

III. 3.3 – O cálculo de iluminação

O "modelo de iluminação" utilizado inclui unicamente o fenómeno da *reflexão*; a transparência não foi considerada. Contudo, a obtenção de figuras que exibam objectos transparentes é possível, graças à utilização de texturas que simulem o aspecto visual dos materiais vistos a transparência (ver capítulo V).

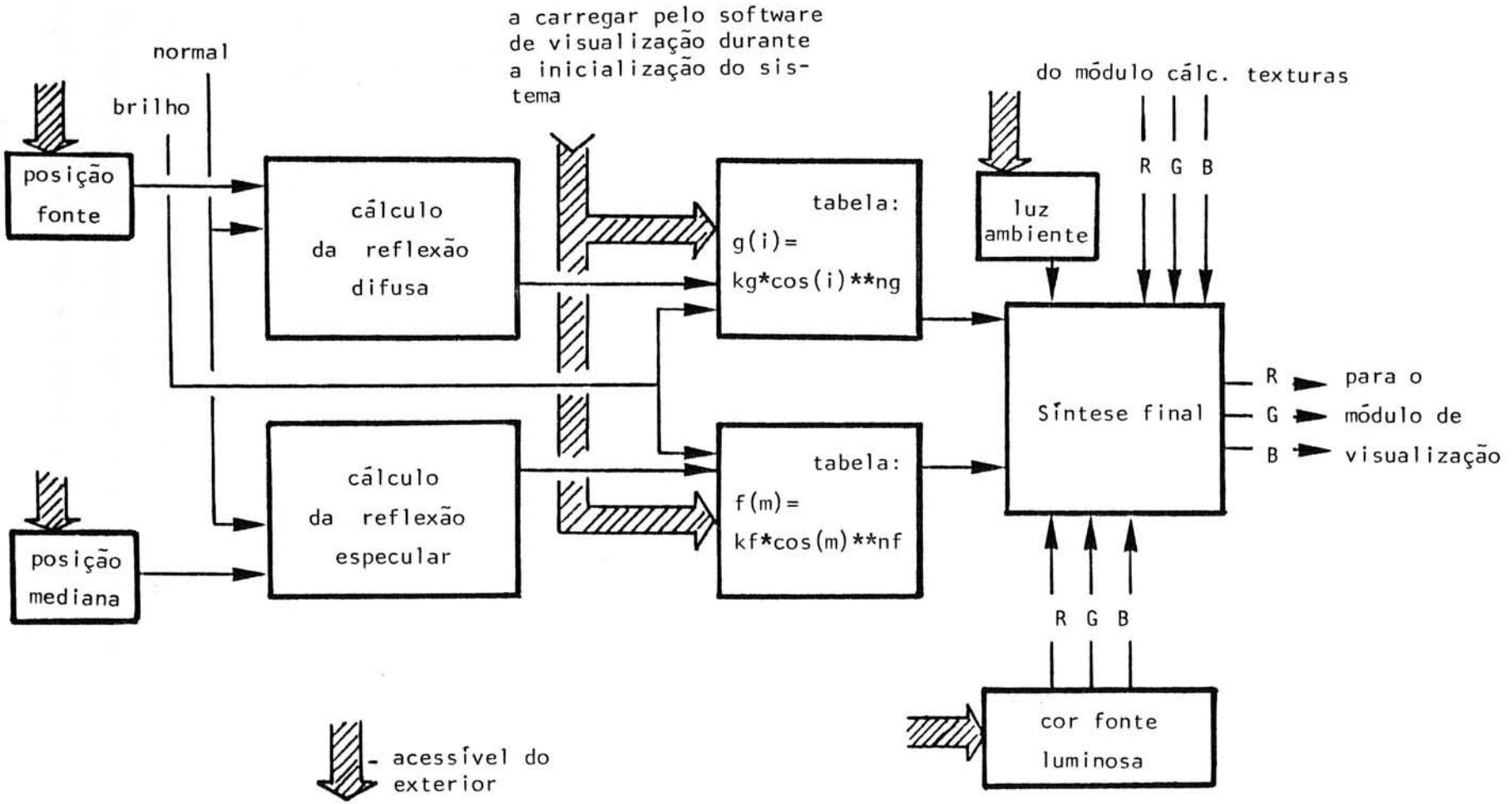


Fig. 3.24 – Esquema geral do módulo cálculo de iluminação.

A iluminação de um ponto é determinada pela soma de três termos:

— o primeiro simula a reflexão difusa do material e é função do ângulo de incidência i definido pelo normal e pela direcção da fonte luminosa,

— o segundo simula a reflexão especular e é função do ângulo m definido pela normal e pela direcção mediana entre a fonte luminosa e o observador,

— o último simula a luz ambiente

A iluminação pode-se então indicar por:

$$I = f(m) \times g(i) \times L$$

A figura 3.24 põe em evidência os parâmetros considerados pelo "modelo de iluminação" que se utilizou e com o qual é possível obter os efeitos devidos às reflexões difusas e especulares:

— vector normal da face que pode tomar 1024 direcções no meio-espaço situado em frente do ecran (6 graus aproximadamente entre duas posições consecutivas) [fig. 3.20],

— posição da fonte luminosa, considerada colocada numa esfera de raio infinito e permitindo 2048 direcções [fig. 3.25],

— posição mediana entre o observador e a fonte luminosa, para a qual são igualmente possíveis 2048 direcções [fig. 3.25],

— cor da fonte luminosa que admite 2×× 12 possibilidades [fig. 3.25]

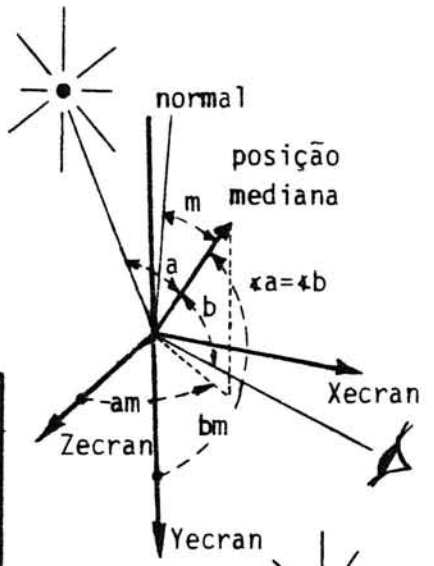
— brilho que simula quatro comportamentos diferentes dos materiais, no que diz respeito à reflexão especular [cf. I. 4.4.2]

O problema a resolver consistia em exprimir os dois ângulos m e i em função do vector normal, da posição da fonte luminosa e da posição mediana, todos expressos em coordenadas esféricas em relação ao referencial associado ao ecran.

Posição mediana

21 20 19 18 17 (bm) 4 3 2 1 0	
0 0 0 0 0	0
⋮	⋮
1 1 1 1 1	≃π

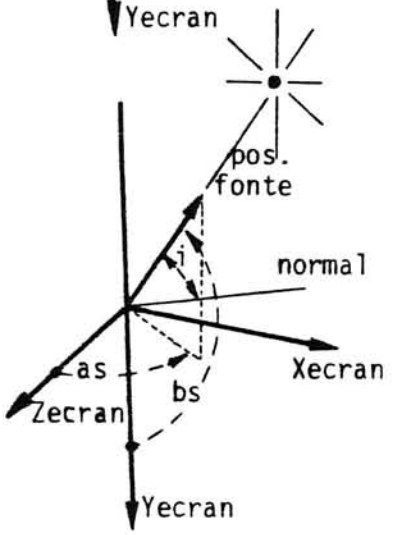
16 15 14 13 12 11 (am) 5 4 3 2 1 0	
0 0 0 0 0 0	-π/2
⋮	⋮
1 1 1 1 1 1	≃2π



Posição fonte

10 9 8 7 6 (bs) 4 3 2 1 0	
0 0 0 0 0	0
⋮	⋮
1 1 1 1 1	≃π

5 4 3 2 1 0 (as) 5 4 3 2 1 0	
0 0 0 0 0 0	- /2
⋮	⋮
1 1 1 1 1 1	≃2π



Luz ambiente

3 2 1 0	
0 0 0 0	nula
⋮	⋮
1 1 1 1	valor máximo

Cor fonte luminosa

11 10 9 8	7 6 5 4	3 2 1 0
G3 G2 G1 G0	R3 R2 R1 R0	B3 B2 B1 B0

Fig. 3.25 – A codificação de alguns parâmetros do modelo de iluminação.

Vamos tomar como exemplo o caso do ângulo m, pois as duas resoluções são praticamente idênticas.

A expressão deste ângulo pode escrever-se, numa primeira fase;

$$\cos(m) = 1/2 \times((\cos(bm-bn) - \cos(bm + bn)) \times \cos(an-am) + \cos(bn + bm) + \cos(bm-bn))$$

onde (an, bn) representam a direcção normal e (am, bm) a direcção da posição mediana.

Se introduzirmos uma função h tal que:

$$h(x) = \cos(x) + 1$$

podemos escrever:

$$2 \times h(m) = h(an-am) \times (h(bn-bm) - h(bn+ bm)) + 2\times (h(bn + bm)).$$

e consequentemente:

$$f(m) = f(h^{-1}(h(an-am)/2 \times(h(bn-bm) - h(bn + bm)) \times h(bn + bm))).$$

A função f(m) é registada numa tabela durante a inicialização do sistema, mas poderá ser modificada, ulteriormente, pelo utilizador [fig. 3.2 6]. Este tem assim a possibilidade de introduzir uma outra função, o que lhe permitirá visualizar "efeitos especiais".

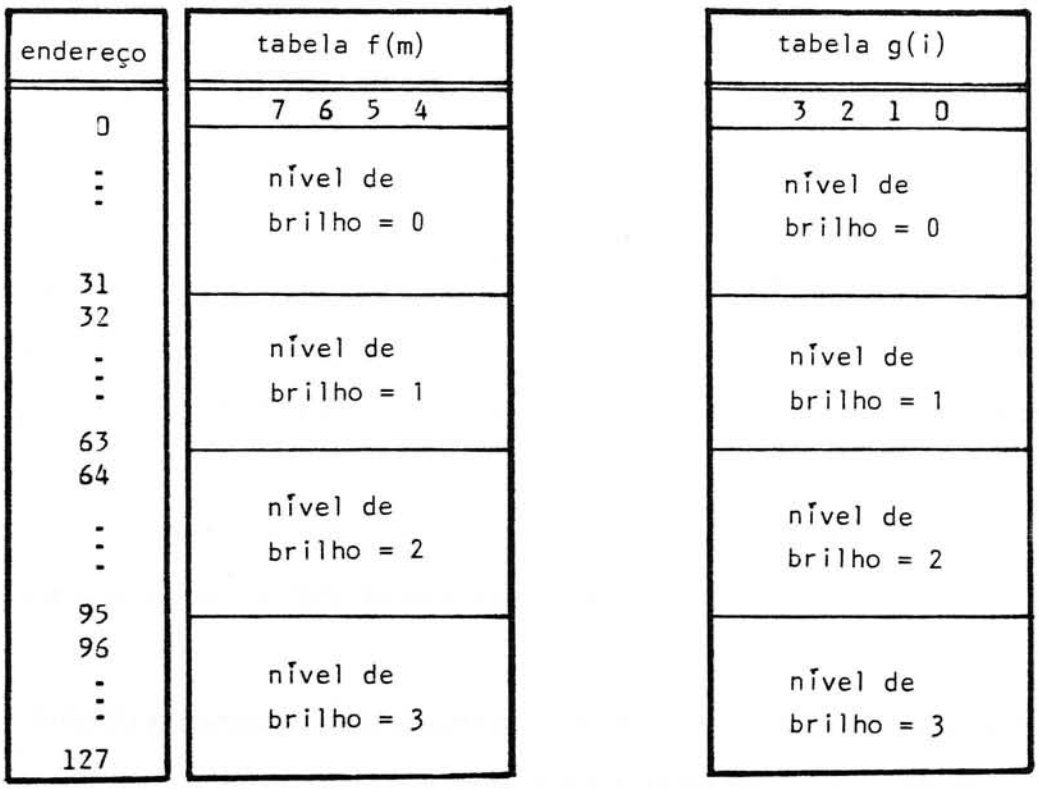


Fig. 3.26 – A codificação das tabelas g(i) e f(m).

III. 3.4 –O módulo de visualização

Este módulo, que não é acessível ao utilizador, assegura a compatibilidade entre a frequência da informação que lhe chega com a frequência própria do varrimento video [fig. 3.27].

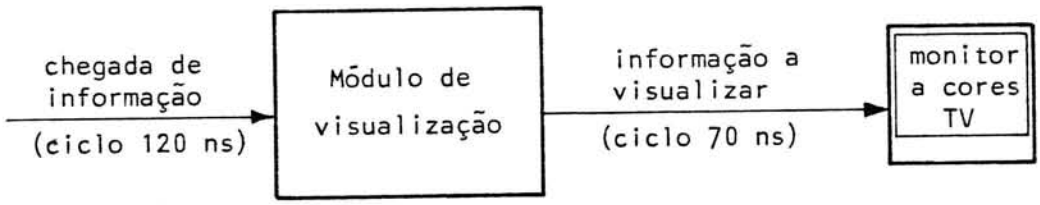


Fig. 3.27 – A missão principal do módulo de visualização.

III. 4 A INTERFACE COM O EXTERIOR

HELIOS admite acessos vindos do exterior possibilitando a leitura ou a modificação das suas memórias e registos [cf. III. 1.2]. Estes acessos são activados a partir de um posto de trabalho [cf. II. 5.4], que se encontra directamente ligado a uma unidade de tratamento [fig. 3.28].

III. 4.1 – A arquitectura de HELIOS vista do utilizador

HELIOS apresenta-se, visto do exterior, como um conjunto de módulos programáveis [cf. III. 3] todos com um objectivo comum: produzir imagens compatíveis com um monitor a cores de varrimento do tipo TV [fig. 3.28].

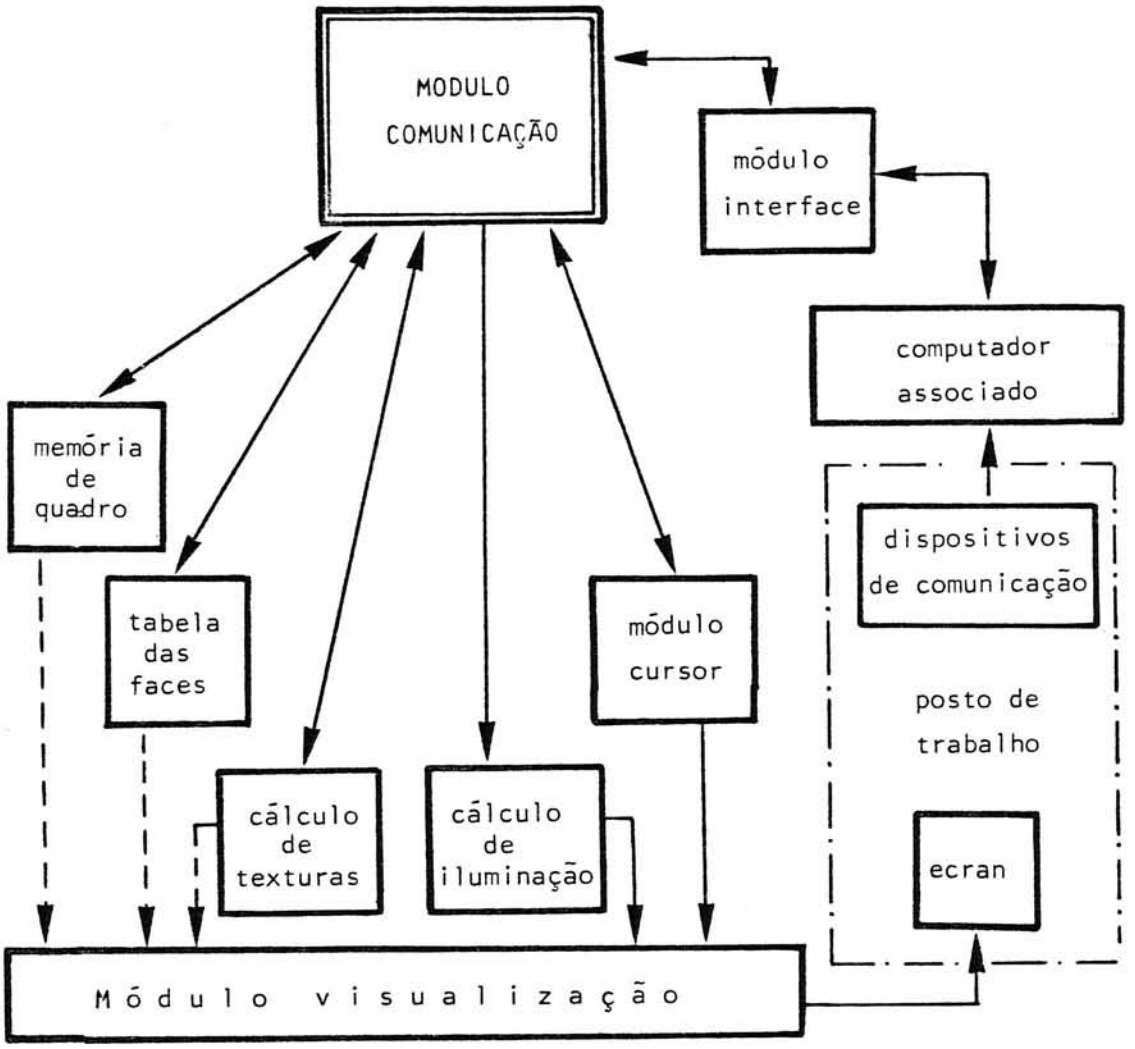


Fig. 3.28 – A arquitectura de HELIOS vista do exterior.

A missão do Módulo Interface [fig. 3.28], quase transparente para o utilizador, resume-se, praticamente, a assegurar a compatibilidade entre o comprimento do "bus" de dados interno de HELIOS (24 bits) e o comprimento do mesmo "bus" da unidade de tratamento associada.

Os módulos de HELIOS são acessíveis do exterior de uma forma indirecta, através do *Módulo de Comunicação,* cujas tarefas fundamentais são:

— recepção de um pedido de acesso que será memorizada num registo deste módulo,

— interpretação e execução efectiva do acesso.

Módulo	Variável	Formato
plano ident. (tabela visib.) cursor (visibilidade)	VISIBILIDADE CURVIS	UMBIT
tabela das faces	BRILHO	DOISBIT
iluminação plano ident. (janela) plano fundo (janela)	LUZAMBIENTE ABERTURAi ABERTURAf	QUATROBIT
iluminação (tabela brilho)	TABBRILHO	OITOBIT
plano identificação plano fundo tabela das faces tabela das faces	NFACE NFACE APONTADOR NORMAL	DEZBIT
plano das texturas iluminação cursor (cor)	COR CORFONTE CURCOR	DOZEBIT
cursor (posição)	CURPOS	DEZ8BIT
plano ident. (janela) plano fundo (janela)	CANTOJANELAi CANTOJANELAf	VINTEBIT
tabela das faces iluminação	PROJECÇÃO POSIÇÕES	VINTE2BIT

Fig. 3.29 – Os formatos utilizados.

II. 4.2 – **Os formatos de informação utilizados**

Os módulos de HELIOS e as funções que lhes estão associados exigem uma informação circulante, nas operações de entrada/saída, que apresentam comprimentos bastante diversificados (comprimentos entre 1 e 22 bits).

Podemos assim agrupar toda esta informação por formatos (em função do respectivo comprimento), de maneira a facilitar a sua utilização e, simultaneamente, a apresentação dos parágrafos que se seguem.

A recapitulação do estudo precedente [cf. III. 3] conduz-nos ao quadro da figura 3.29, onde o número de formatos definidos é, praticamente, metade do número de variáveis presentes.

O formato nBIT engloba todas as variáveis cujo comprimento é igual a n e a sua relação com o "bus" de dados interno (BDI) de HELIOS (BDI 23, ..., BDI 0) é ilustrado pela figura 3.30.

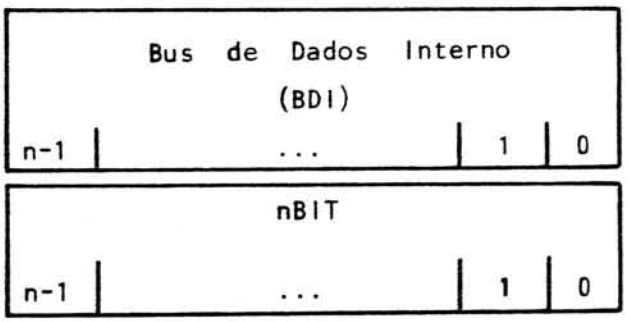


Fig. 3.30 – O formato nBIT.

III. 4.3 – **O módulo interface**

Este módulo foi concebido de tal maneira que nenhuma modificação "hardware" é necessária, qualquer que seja o comprimento da palavra da unidade de tratamento associada, desde que igual ou maior que oito [fig. 3.31].

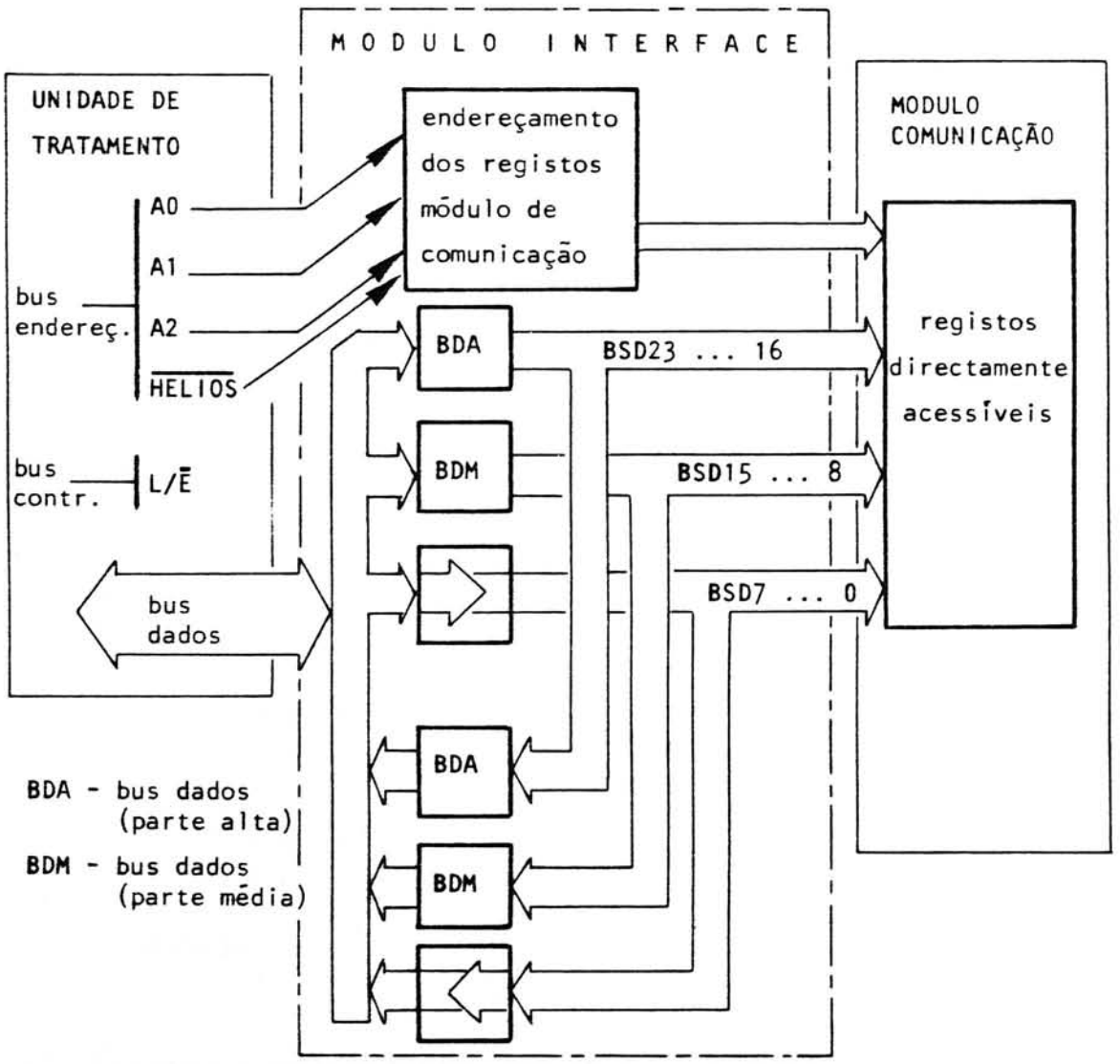


Fig. 3.31 – Esquema geral do módulo Interface.

Desta maneira, se se dispõe de uma unidade de tratamento de 24 bits ou mais, os acessos a um registo de HELIOS efectuam-se com uma única instrução do tipo:

OUT endereço registo (escrever num registo)

ou

INP endereço registo (ler um registo)

No caso de uma unidade de 16 bits:

OUT BDA [fig. 3.31 e fig. 3.32]

OUT endereço registo

ou

INP endereço registo

INP BDA

Agora, se se utiliza, por exemplo, um microcomputador de 8 bits:

OUT BDA

OUT BDM [fig. 3.31 e fig. 3.32]

OUT endereço registo

ou

INP endereço registo

INP BDM

INP BDA

Os endereços dos registos BDA ("bus" de dados alto) e BDM ("bus" de dados médio) são indicadas na figura 3.32.

registo	A2	A1	A0	$\overline{HELIOS}$	L/$\overline{E}$
BDM	0	0	1	0	a
BDA	0	1	0	0	a

a=0 - para HELIOS
a=1 - para a unidade associada

Fig. 3.32 – Os endereços dos registos BDM e BDA.

III. 4.4 – **O módulo de comunicação**

O *Módulo de Comunicação* comporta um conjunto de registos especiais [fig. 3.33], directamente acessíveis do exterior, cujos conteúdos definem toda a informação necessária à activação do *Processo de Comunicação* [cf. III. 4.5].

Dispõe-se de registos que permitem:

— escolher o módulo com o qual se pretende comunicar,

— endereçar elementos distintos no interior do módulo escolhido. Por exemplo, num plano de identificação é necessário definir uma posição entre as 512 × 512 possíveis,

— seleccionar a função a executar. No exemplo apontado é necessário definir, se se deseja escrever ou ler no plano, escrever ou ler na tabela de visibilidade, etc.

— definir o modo de acesso. Por exemplo, escolher o momento adequado à execução do acesso,

— utilizar uma compressão da informação circulante, para minimizar o caudal da comunicação entre a unidade de tratamento associada e HELIOS,

— finalmente, definir a informação a escrever ou verificar a informação lida.

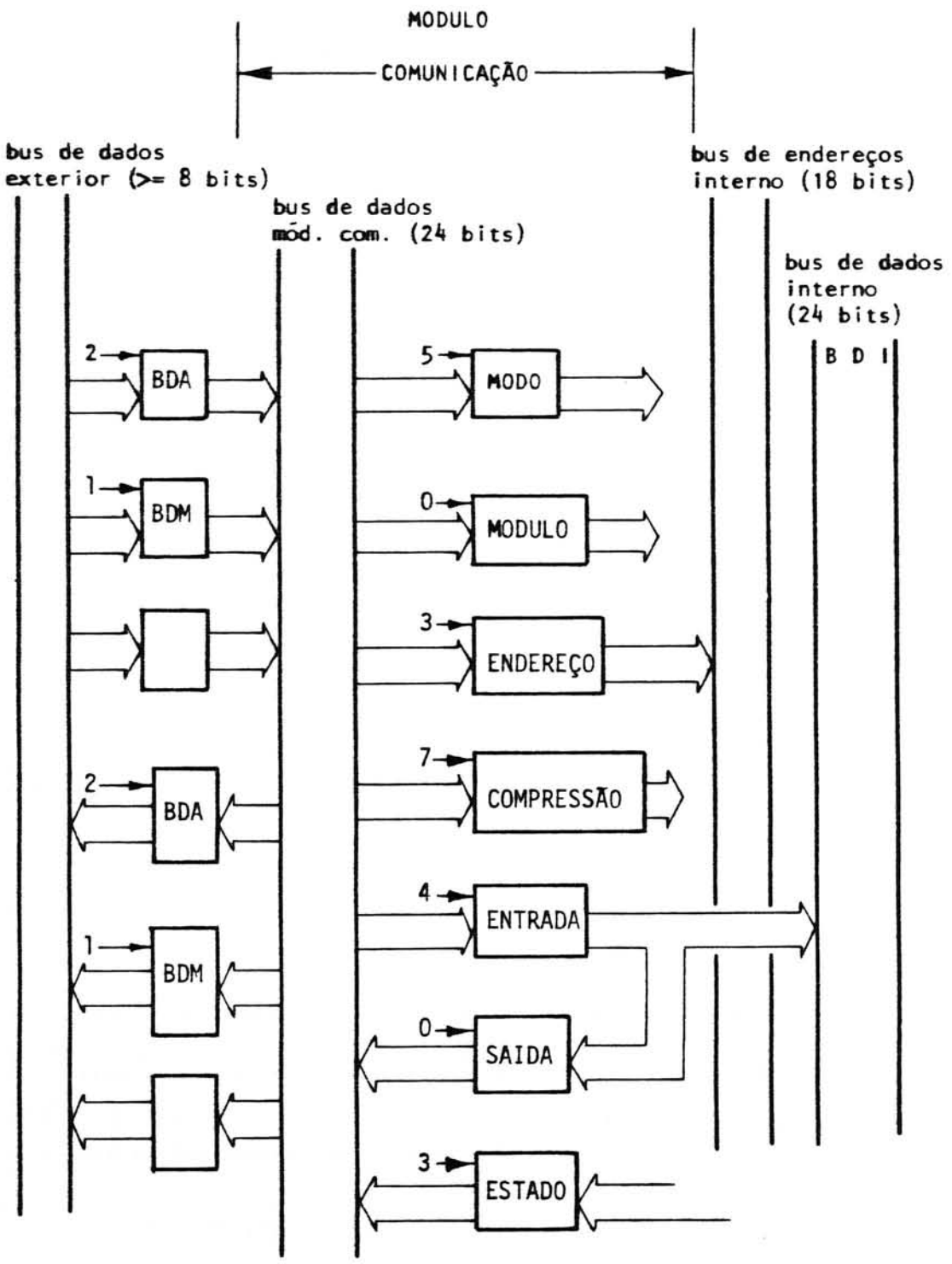


Fig. 3.33 – Os registos do módulo de Comunicação .

III. 4.5 – O processo de comunicação

O *Processo de Comunicação* promove a execução efectiva dos acessos em leitura e em escrita, possibilitando a verificação ou a modificação do estado dos módulos de HELIOS que não são directamente acessíveis do exterior.

A activação do Processo de Comunicação faz-se através da programação dos registos do módulo de comunicação (ver fig. 3.51 e fig. 3.54), aos quais estão associadas funções muito bem definidas e pormenorizadas de seguida.

III. 4.5.1 – A verificação do estado interno

O registo ESTADO faz parte do módulo de comunicação [fig. 3.34] e o seu conteúdo é, em geral, analisado antes de se proceder à activação do processo de comunicação; este procedimento permite deduzir o estado geral do sistema.

registo	A2	A1	A0	$\overline{\text{HELIOS}}$	L/$\overline{\text{E}}$
ESTADO	0	1	1	0	1

a. endereço

2	1	0
CUR VAL	ENTRADA LIVRE	COM LIVRE

b. formato.

Fig. 3.34 – O registo ESTADO.

Podemos distinguir três campos no formato deste registo [fig. 3.34]:

— COMLIVRE

Este campo define o estado do processo de comunicação, como se ilustra na figura 3.35.

COMLIVRE	significado
0	processo comunicação ocupado
1	processo comunicação livre

Fig. 3.35 – O campo COMLIVRE do registo ESTADO.

— ENTRADALIVRE

Nas operações de escrita, o módulo de comunicação aceita uma nova informação enquanto a precedente se encontra ainda numa fase de processamento [cf. III. 4.5.4]; foi previsto um registo auxiliar do registo ENTRADA, designado por ENTRADAAUX, cujo estado está relacionado com o campo ENTRADALIVRE do registo ESTADO [fig. 3.36].

ENTRADALIVRE	significado
0	ENTRADABUF ocupado
1	ENTRADABUF livre (mesmo se o processo de comunicação está activo)

Fig. 3.36 – O campo ENTRADALIVRE do registo ESTADO.

— CURVAL

Este campo está relacionado com o estado de visibilidade do cursor e a validade do ponto que ele aponta [cf. III. 4.5.5] [fig. 3.37].

CURVAL	significado
0	ponto não validado e cursor visível
1	ponto validado e cursor invisível

Fig. 3.37 – O campo CURVAL do registo ESTADO.

registo	A2	A1	A0	$\overline{\text{HELIOS}}$	L/$\overline{\text{E}}$
MODO	1	0	1	0	0

a. endereço

4	3 2 1 0
ENDIMAGEM	SINCRAC

b. formato.

Fig. 3.38 – O registo MODO.

III. 4.5.2 – **O modo de acesso**

Esta função está associada com o registo MODO, cujo formato se decompõe em dois campos [fig. 3.38].

— SINCRAC

Um pedido de acesso vindo do exterior é memorizado pelo módulo de comunicação e espera, de acordo com o campo SINCRAC [fig. 3.39], o momento adequado à sua execução,

	SINCRAC 3	2	1	0	significação
	0	0	0	0	- acesso sempre possível
	0	0	0	1	- acesso durante os retornos de quadro
	0	0	1	0	- acesso durante os retornos dos quadros pares
	0	0	1	1	- acesso durante os retornos dos quadros ímpares
*	0	1	0	0	- acesso a partir do início da visualização dos quadros pares
*	0	1	0	1	- acesso a partir do início da visualização dos quadros ímpares
*	0	1	1	1	- acesso a partir do início do retorno dos quadros pares
*	0	1	1	1	- acesso a partir do início do retorno dos quadros ímpares
	1	-	-	-	- acesso impossível

* - nestes casos, a partir do primeiro acesso, SINCRAC toma o valor zero.

Fig. 3.39 – O campo SINCRAC do registo MODO.

— ENDIMAGEM

O campo ENDIMAGEM (endereço imagem) está relacionado com o modo de endereçamento e tem uma certa influência sobre o significado da informação contida no registo ENDERECO [cf. III. 4.53].

III. 4.5.3 – **O endereçamento interno**

Esta função é realizada com a ajuda de dois registos: os registos MODULO e ENDERECO.

1 – Registo MODULO

No formato do registo MODULO distinguem-se dois campos [fig. 3.40]:

registo	A2	A1	A0	$\overline{\text{HELIOS}}$	L/$\overline{\text{E}}$
MODULO	0	0	0	0	0

a. endereço.

6 5 4	3 2 1 0
FUNÇÃO	MOD

b. formato.

Fig. 3.40 – O registo MODULO.

– MOD

O campo MOD especifica o módulo com o qual se pretende comunicar [fig. 3.41].

MOD 3 2 1 0	significado
0 0 0 0	plano identificação 0
. . .	. . .
0 1 1 0	plano identificação 6
0 1 1 1	memória das texturas
1 0 0 1	plano fundo
1 0 0 1	tabela das faces
1 0 1 0	cálculo de iluminação
1 0 1 1	cursor

Fig. 3.41 – O campo MOD do registo MODULO.

— FUNCAO

Este campo define, para o módulo escolhido, a função a executar.

Vejamos os diferentes casos possíveis:

— os Planos de Identificação [fig. 3.42],

— a Memória de Texturas [fig. 3.43],

— o Plano de Fundo [fig. 3.44],

— a Tabela das Faces [fig. 3.45],

— o Módulo Cálculo de Iluminação [fig. 3.46],

— o Cursor [fig. 3.47].

FUNÇÃO 6 5 4	MOD 3 2 1 0	significado	formato
0 0 0	plano id.0 : : plano id.6	escrever NFACE	DEZBIT
0 0 1		escrever VISIBILIDADE	UMBIT
0 1 0		escrever CANTOJANELA	VINTEBIT
0 1 1		escrever ABERTURA	QUATROBIT
1 0 0		ler NFACE	DEZBIT
1 0 1		ler VISIBILIDADE	UMBIT
1 1 0		ler CANTOJANELA	VINTEBIT
1 1 1		ler ABERTURA	QUATROBIT

Fig. 3.42 – Os acessos aos planos de identificação.

FUNÇÃO 6 5 4	MOD 3 2 1 0	significado	formato
0 0 0	0 1 1 1 (mem. text.)	escrever COR	DOZEBIT
1 0 1		ler COR	DOZEBIT

Fig. 3.43 – Os acessos à memória de texturas.

FUNÇÃO 6 5 4	MOD 3 2 1 0	significado	formato
0 0 0	1 0 0 0	escrever NFACE	DEZBIT
0 0 1	(plano		
0 1 0	fundo)	escrever CANTOJANELA	VINTEBIT
0 1 1		escrever ABERTURA	QUATROBIT
1 0 0		ler NFACE	DEZBIT
1 0 1			
1 1 0		ler CANTOJANELA	VINTEBIT
1 1 1		ler ABERTURA	QUATROBIT

Fig. 3.44 – Os acessos ao plano de fundo.

FUNÇÃO 6 5 4	MOD 3 2 1 0	significado	formato
0 0 0	1 0 0 1	escrever APONTADOR	DEZBIT
0 0 1	(tabela das	escrever PROJECÇÃO	VINTE2BIT
0 1 0	faces)	escrever BRILHO	DOISBIT
0 1 1		escrever NORMAL	DEZBIT
1 0 0		ler APONTADOR	DEZBIT
1 0 1		ler PROJECÇÃO	VINTE2BIT
1 1 0		ler BRILHO	DOISBIT
1 1 1		ler NORMAL	DEZBIT

Fig. 3.45 – Os acessos à tabela das faces.

FUNÇÃO 6 5 4	MOD 3 2 1 0	significado	formato
0 0 0	1 0 1 0	escrever TABBRILHO	OITOBIT
0 0 1	(cálculo de	escrever CORFONTE	DOZEBIT
0 1 0	iluminação)	escrever POSIÇÕES	VINTE2BIT
0 1 1		escrever LUZAMBIENTE	QUATROBIT

Fig. 3.46 – Os acessos ao módulo Cálculo de iluminação.

FUNÇÃO 6 5 4	MOD 3 2 1 0	significado	formato
0 0 0	1 0 1 1 (cursor)	escrever CORCUR	DOZEBIT
0 0 1		escrever CURVIS	UMBIT
0 1 0		escrever POSCUR	DEZ8BIT
0 1 1			
1 0 0			
1 0 1			
1 1 0			
1 1 1		ler POSCUR	DEZ8BIT

Fig. 3.47 – Os acessos ao cursor.

registo	A2	A1	A0	$\overline{\text{HELIOS}}$	L/$\overline{\text{E}}$
ENDEREÇO	0	1	1	0	0

b. formato

17 ... 1 0
ENDEREÇO

a. endereço

Fig. 3.48 – O registo ENDERECO.

2 – Registo ENDERECO

O registo ENDERECO define o endereço da próxima posição à qual se pretende aceder, dentro do módulo escolhido [fig. 3.48].

O endereço (conteúdo do registo em questão) é incrementado automaticamente de uma unidade sempre que é executado um acesso, ficando desta maneira a apontar a posição seguinte.

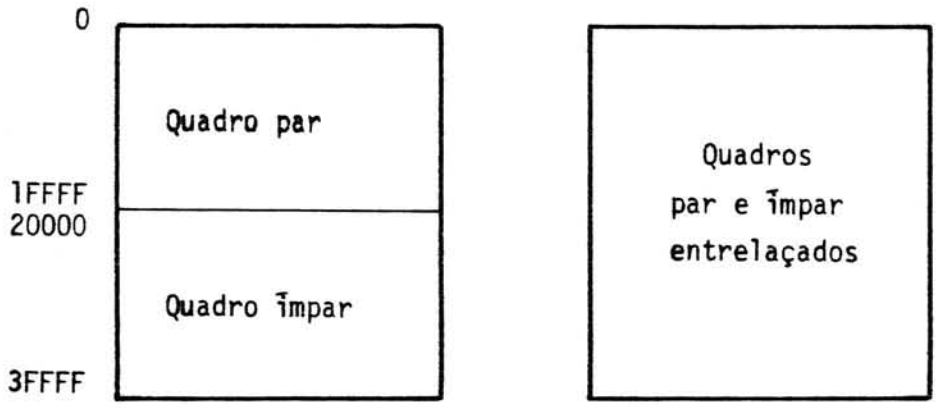
Consideremos agora a influência do campo ENDIMAGEM (endereço imagem) do registo MODO [cf. III. 4.5.2], sobre o significado do conteúdo do registo ENDERECO.

Regra geral, ENDIMAGEM apresenta o valor lógico 1, salvo no caso dos planos de identificação, onde podem ser especificados ambos os valores 0 e 1.

Se ENDIMAGEM = 0, aqueles módulos de memória são vistos do exterior organizados segundo os dois quadros que compõem a imagem [fig. 3.49. a]

Pelo contrário, se ENDIMAGEM = 1, aqueles módulos apresentam-se organizados de acordo com o varrimento do tipo TV, ou seja, com os quadros pares e ímpares entrelaçados [fig. 3.49.b].

Podemos assim, se ENDIMAGEM = 0, preencher um quadro (par ou ímpar) num plano de identificação enquanto que o outro é visualizado, aproveitando ao mesmo tempo o facto do registo ENDERECO ser incrementado automaticamente de uma unidade, por cada acesso.



a. ENDIMAGEM = 0 b. ENDIMAGEM = 1.

Fig. 3.49 – Os módulos de memória vistos do exterior de acordo com ENDIMAGEM.

III. 4.5.4 – **O acesso em leitura**

A execução efectiva de um acesso em leitura resulta da activação de um processo "hardware", designado por *Processo de Comunicação,* através de uma sequência 'software" especializada [fig. 3.51]; salientemos o paralelismo nas execuções deste processo e da sequência "software" referida, a partir:

— do momento que se transmite ao registo ENDERECO o seu novo conteúdo,

— da leitura do registo SAIDA [fig. 3.50].

registo	A2	A1	A0	$\overline{\text{HELIOS}}$	L/$\overline{\text{E}}$
SAIDA	0	0	0	0	1

a. endereço.

23 ... 1 0
SAIDA

b. formato compatível com o módulo escolhido

Fig. 3.50 – O registo SAÍDA.

Após a activação do processo de comunicação em leitura (COMLIVRE = 0), a execução efectiva desta operação é indicada por COMLIVRE = 1. A partir deste momento o registo SAÍDA, acessivel do exterior, contém a informação desejada, cujo formado dependerá do módulo acedido.

III. 4.5.5 – **O acesso em escrita**

Um acesso em escrita resulta igualmente da execução de um processo "hardware", activado por uma sequência "software" [fig. 3.54]. Contudo, em relação à leitura, novos registos são tomados em linha de conta:

```
PROCEDURE SEQUENCIA_LEITURA (MODO_ESCOLHIDO, ... );
BEGIN
   BOOLEAN: COM_LIVRE, ESTADO(0..2);
   INTEGER: INFO_A_LER, ENDERECO, SAIDA, MODO, MODULO;
   PROCEDURE: LER, ESCREVER;
      ...
   LER (ESTADO); COM_LIVRE:=ESTADO(0);
   WHILE NOT COM_LIVRE DO
   BEGIN
      LER(ESTADO); COM_LIVRE:=ESTADO(0);
   END;
   ESCREVER(MODO,MODO_ESCOLHIDO);ESCREVER (MODULO,MODULO_ESCOLHIDO);
   ESCREVER(ENDERECO,ENDERECO_INICIAL); (* apelo implicito ao processo
                                           hardware "COM_LEITURA" *)
   WHILE INFO_A_LER DO
   BEGIN
      LER(ESTADO);COM_LIVRE:=ESTADO(0);
      WHILE NOT COM_LIVRE DO
      BEGIN
         LER(ESTADO);COM_LIVRE:=ESTADO(0);
      END;
      LER(SAIDA);   (*apelo implicito ao processo hardware "COM_LEITURA" *)
         ...
      INFO_A_LER:=INFO_A_LER-1;
   END;
END (* SEQUENCIA_LEITURA *);

PROCEDURE LER (REG_HELIOS);
BEGIN
   ...
   REG_UNIDADE_ASSOCIADA:=REG_HELIOS;
END (* LER);

PROCEDURE ESCREVER(REG_HELIOS,INFO);
BEGIN
   ...
   REG_HELIOS:=INFO;
END;

PROCEDURE COM_LEITURA;(* processo hardware *)
BEGIN
   ESTADO(0):=FALSE;     (* processo de comunicação ocupado *)
   SAIDA:=EXECUCAO_LEITURA (MODO,MODULO,ENDERECO);
   ENDERECO:=ENDERECO+1;
   ESTADO(0):=TRUE;      (* processo de comunicação livre *)
END (*COM_LEITURA *);
```

Fig. 3.51 – Sequencia normal para activar o processo de leitura.

— Registo COMPRESSAO

O conteúdo do registo COMPRESSAO informa o processo de comunicação sobre o *factor de repetição* ou *de compressão* da operação de acesso em escrita [fig. 3.52]. Este factor é igual ao número de posições consecutivas de memória do módulo escolhido que vão ser preenchidas com o conteúdo do registo ENTRADA, a primeira delas definida pelo conteúdo do registo ENDERECO.

registo	A2	A1	A0	$\overline{\text{HELIOS}}$	L/$\overline{\text{E}}$
COMPRESSÃO	1	1	1	0	0

a. endereço.

n	7	6	5	4	3	2	1	0
	COMPRESSÃO							
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
. . .					. . .			
255	1	1	1	1	1	1	1	1

b. formato.

Fig. 3.52 – O registo COMPRESSAO.

— Registo ENTRADA

O conteúdo do registo ENTRADA define a informação a memorizar, a qual apresentará um formato compatível com o módulo seleccionado [cf. III. 3.2] [fig. 3.53].

registo	A2	A1	A0	$\overline{\text{HELIOS}}$	L/$\overline{\text{E}}$
ENTRADA	1	0	0	0	0

a. endereço.

23 ... 1 0
ENTRADA

b. formato compatível com o módulo escolhido.

Fig. 3.53 – O registo ENTRADA.

```
PROCEDURE SEQUENCIA_ESCRITA (MODO_ESCOLHIDO, ... );
BEGIN
   BOOLEAN: COM_LIVRE,ENTRADA_LIVRE,ESTADO(0..2);
   INTEGER: INFO_A_ESCREVER,ENDERECO,ENTRADA,COMPRESSAO,MODO,MODULO,N,INFO;
   PROCEDURE: LER,ESCREVER;
      ...
   LER (ESTADO); COM_LIVRE:=ESTADO (0);
   WHILE NOT COM_LIVRE DO
   BEGIN
      LER (ESTADO); COM_LIVRE:=ESTADO(0);
   END;
   ESCREVER(MODO,MODO_ESCOLHIDO); ESCREVER(MODULO,MODULO_ESCOLHIDO);
   ESCREVER (ENDERECO,ENDERECO_INICIAL);
      ...
   WHILE INFO_A_ESCREVER DO
   BEGIN
      ESCREVER (COMPRESSAO,N);
      ESCREVER (ENTRADA,INFO);  (* apelo implicito ao processo hardware
                                   "ACTIVACAO"   *)
      INFO_A_ESCREVER:=INFO_A_ESCREVER-N;
      LER (ESTADO); ENTRADA_LIVRE:=ESTADO(1);
      WHILE NOT ENTRADA_LIVRE DO
      BEGIN
         LER (ESTADO); ENTRADA_LIVRE:=ESTADO(1);
      END;
         ...
   END;
END  (*  SEQUENCIA_ESCRITA  *);

PROCEDURE ACTIVACAO;     (* processo hardware *)
BEGIN
   ...
   ESTADO(1):= FALSE;    (* entrada ocupada *)
   ESTADO(0):= FALSE;    (* processo "COM_ESCRITA" ocupado  *)
END (* ACTIVACAO *);

PROCEDURE COM_ESCRITA;   (* processo hardware *)
BEGIN
   ...
   WHILE NOT ESTADO(1) DO
   BEGIN
      CONTADOR_ESCRITA:=COMPRESSAO; ENTRADA_AUX:=ENTRADA;
      ESTADO(1):=TRUE     (* entrada livre *)
      WHILE CONTADOR_ESCRITA>0 DO
      BEGIN
         EXECUCAO_ESCRITA (MODO,MODULO,ENDERECO,ENTRADA_AUX);
         ENDERECO:=ENDERECO+1;
         CONTADOR_ESCRITA:=CONTADOR_ESCRITA-1;
      END;
   END;
   ESTADO(0):=TRUE;      (* processo de comunicação livre   *)
END     (* COM_ESCRITA *);
```

Fig. 3.54 – Sequência normal para activar o processo de escrita.

Considerando igual a n o conteúdo do registo COMPRESSAO, são automaticamente activados n acessos em escrita sobre o módulo escolhido, a partir do endereço especificado pelo registo ENDERECO, logo que seja inicializado o registo ENTRADA [fig. 3.54].

No final destas operações, o conteúdo do registo ENDERECO será igual a

endereço inicial + n

III. 4.5.6 – **O controlo do cursor**

HELIOS dispõe, no espaço imagem 512 × 512, de um cursor colorido destinado à recolha de coordenadas e à designação de entidades visualizadas.

Por "software" [cf. III. 4.5.3] podemos:

— escolher a sua cor (formato DOZEBIT) [fig. 3.55],

— modificar o seu indicador de visibilidade (formato UMBIT) [fig. 3.55],

— modificar ou ler a sua posição no espaço imagem (formato DEZ8BIT) [fig. 3.55].

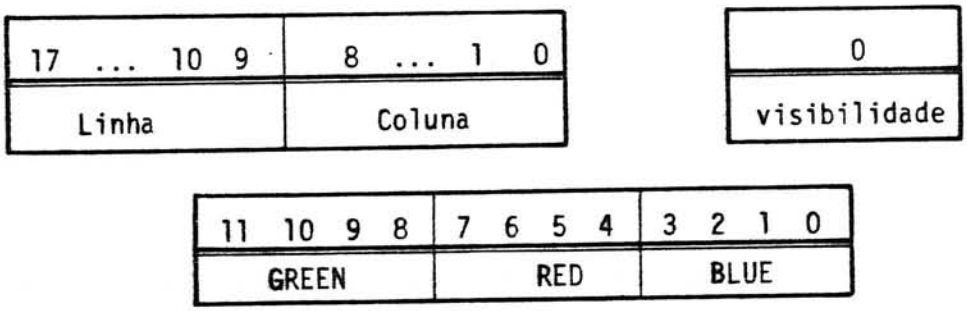


Fig. 3.55 – O formato das informações associadas ao cursor.

Por "software" podemos igualmente verificar o valor associado ao indicador de visibilidade deste cursor [cf. III. 4.5.1].

Um conjunto de teclas, considerado directamente pelo "hardware", controla o indicador de visibilidade e a posição do cursor sobre o ecran [fig. 3.56]

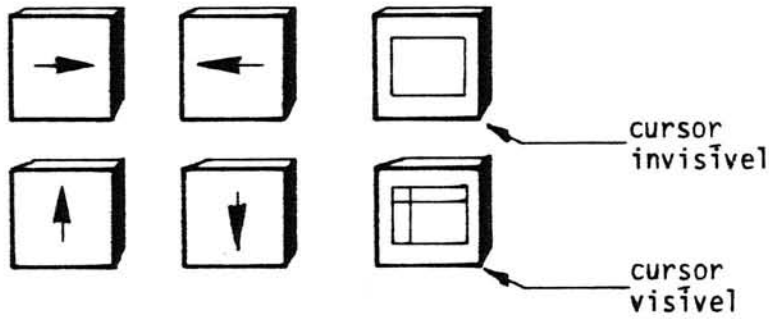


Fig. 3.56 – Um conjunto de teclas para o controlo do cursor.

III. 5 ALGUMAS CONSIDERAÇÕES SOBRE O SOFTWARE DE VISUALIZAÇÃO

Não se pretende neste momento estudar de maneira aprofundada a estrutura geral ou uma realização particular do "software" de visualização adaptado a HELIOS, mas simplesmente pôr em evidência as funções deste "software" que evitarão ao utilizador um conjunto de tarefas fastidiosas, características de uma programação ao nível máquina. Assim, o que se segue deve ser interpretado como um conjunto de pistas a considerar numa realização futura.

As tarefas fundamentais do "software" de visualização têm por objectivo [cf. I. 3.2]:

— estabelecer a independência entre o "software" e a máquina,

— oferecer uma utilização óptima das possibilidades da máquina; este objectivo exige do "software" a capacidade para determinar as funções que são efectivamente realizadas pelo processador gráfico, para assim poder aproveitar ao máximo as facilidades que este oferece.

— facilitar a missão do utilizador, tornando "transparente" um certo número de operações delicadas e fastidiosas inerentes a uma programação em linguagem máquina.

Destas tarefas básicas resultam algumas funções particulares, as quais devem ser consideradas como um nó mínimo indispensável a uma utilização ao mesmo tempo agradável e eficiente:

— Inicialização do sistema, operação que o coloca num estado coerente.

- inicialização dos registos (Por exemplo, MODO, ...)
- inicialização das memórias (Por exemplo, Memória de Quadro, Tabelas de Visibilidade, Tabelas das Faces, Memórias de Texturas, Tabelas do módulo Cálculo de Iluminação).

— ao nível dos módulos de Comunicação e Interface

- conjunto de primitivas de entrada/saída que tornem "transparente" a manipulação dos registos directamente acessíveis do exterior,
- utilização "transparente" dos registos BDA e BDM da Interface, independentemente do comprimento da palavra da unidade de tratamento associada,
- utilização fácil dos diferentes modos de acesso (manipulação do registo MODO),
- manipulação do cursor.

— ao nível dos Planos de Identificação

- utilização das facilidades do módulo de comunicação, em particular, a compressão da informação circulante no preenchimento destes planos,
- primitivas que permitam a identificação das faces visualizadas,
- gestão das Tabelas de Visibilidade e da função Janela.

— ao nível da Tabela das Faces

- manipulação do apontador de texturas e do nível de brilho,
- actualização automática do vector normal e da matriz de projecção.

— ao nível da Memória de Texturas

- facilidades de criação de texturas (escolha de cores, preenchimento das matrizes, ...),
- memorização de novas texturas,
- gestão do espaço livre desta memória.

— ao nível do módulo Cálculo de Iluminação

- possibilidade de modificar as tabelas de brilho,
- gestão dos parâmetros que dizem respeito à fonte luminosa e à intensidade de luz ambiente.

— finalmente, aproveitamento das capacidades do terminal que permitem um dinamismo em tempo real.

- compressão da informação circulante
- função Janela
- visibilidade/invisibilidade das faces
- modificação do conjunto de texturas visualizadas
- modificação dos parâmetros considerados pelo modelo de iluminação.

III. 6 – CONCLUSÃO

HELIOS foi considerado como um caso particular de um sistema informático, composto por: uma parte "hardware" (a máquina propriamente dita) e uma parte "software", da qual se realça o "software" de visualização. Com este pretende-se oferecer uma utilização eficaz e natural das possibilidades do "hardware", retirando ao utilizador algumas tarefas fastidiosas com origem numa programação ao nível máquina.

Desta maneira, o objectivo fundamental deste capítulo consistiu em fornecer informações sobre HELIOS que serão certamente necessárias a um futuro desenvolvimento do "software" de visualização.

Examinou-se o princípio de funcionamento e os meios de comunicação com o exterior, mas deixou-se a justificação das soluções utilizadas para o capítulo seguinte.

A parte final deste capítulo inclui uma lista das principais tarefas da responsabilidade do "software" de visualização, adaptadas ao caso particular de HELIOS.

CAPÍTULO IV

REALIZAÇÃO DE UM PROTÓTIPO

CAPÍTULO IV: REALIZAÇÃO DE UM PROTÓTIPO

IV. 1. INTRODUÇÃO

Este capítulo apresenta e justifica, no estado de desenvolvimento tecnológico actual da micro-electrónica, as soluções utilizadas na a realização de um protótipo que responde as características enunciadas no Capítulo III.

Este protótipo, realizado com a colaboração do Laboratório de Micro-electrónica do IMAG (Laboratoire d'Informatique et de Mathématiques Appliquées de Grenoble), tinha como objectivo fundamental avaliar a possibilidade de realização e a validade dos princípios de base de HELIOS, em particular, a manipulação ao nível "hardware" da noção de face 3D, em vez do ponto "anónimo" considerado pela generalidade dos terminais baseados num monitor de varrimento TV.

Um estudo preliminar mostrou-nos a viabilidade de um projecto deste tipo, apesar de um certo número de problemas que não apresentavam soluções evidentes. Com efeito, o ritmo do sinal video impõe tempos de tratamento extremamente curtos em relação aos apresentados pela maioria das funções integradas, como sejam as memórias MOS de grande capacidade; o volume dos Planos de Identificação ou da Memória de Texturas exige o emprego de tais unidades, cujo tempo de ciclo é nitidamente incompatível com a frequência do sinal video. Problemas deste género foram ultrapassados graças à utilização de técnicas de paralelismo e "pipe-line".

O protótipo foi concebido e realizado de acordo com um processo evolutivo que permitiu, em cada etapa de realização, a visualização de imagens, facto que se traduziu por um certo número de vantagens:

— facilidade de teste, para os quais o terminal com a respectiva imagem serviram frequentemente como "analisador lógico",

— visualização de resultados com origem em simulações "software", as quais precederam a realização "hardware" de alguns módulos (cálculo de texturas e cálculo de iluminação).

Salienta-se que estas simulações permitiram, em particular para os cálculos de texturas e de iluminação, rectificar e ajustar certos pormenores.

IV. 2. ALGUMAS CONSIDERAÇÕES GERAIS

IV. 2.1 – Os conceitos de paralelismo e de "pipe-line" utilizados

A figura 4.1 mostra que a obtenção de uma imagem quadrada de resolução 512 × 512, num monitor de varrimento TV, impõe tempos de visualização, por ponto, da ordem de 70 ns.

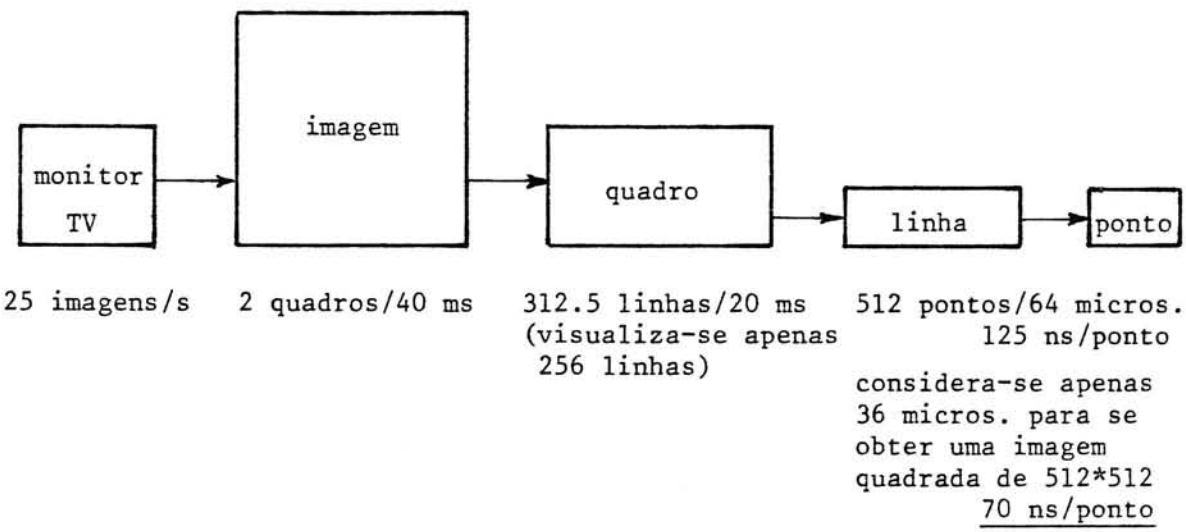


Fig. 4.1 – Os tempos impostos pelo varrimento do tipo TV (25 imagens/s).

Se se considerar o tempo exigido pelo tratamento de cada ponto (no nosso caso este tratamento inclui, nomeadamente, a leitura da memória de quadro, o cálculo de visibilidade, ...), verifica-se imediatamente a necessidade de decompor este tratamento em n tratamentos elementares com:

$$n = T/t$$

onde T exprime o tempo total de tratamento (por ponto) e t o tempo elementar.

Encontramos desta maneira intuitiva a noção de estrutura "pipe-line", onde cada processo elementar é executado por um andar específico independente dos restantes [fig. 4.2]. Assim, por exemplo, se o andar K (processo elementar K) considera o ponto de ordem p, no mesmo instante, o andar k−1 (processo elementar K−1) considera o ponto de ordem p + 1.

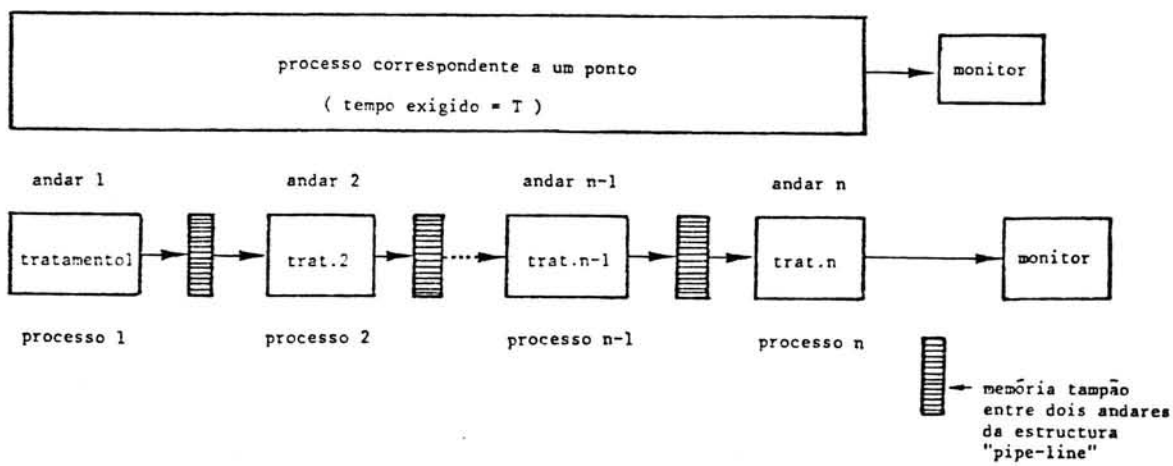


Fig. 4.2 – A estrutura "pipe-line" utilizada.

No nosso caso, o tempo t, da ordem de 70 ns, originava ainda sérios problemas, em particular ao nível dos processos elementares indecomponíveis que exigem tempos de tratamento superiores a t; por exemplo, as memórias MOS cujo ciclo pode atingir 350 ns. Por esta razão, um certo paralelismo é igualmente necessário em alguns andares da estrutura "pipe-line".

A figura 4.3 ilustra um destes casos, representando o andar genérico K a que corresponde o processo elementar K indecomponível e que exige um tempo de tratamento situado entre 70 e 140 ns. O *grau de paralelismo* ao nível deste andar é igual a 2.

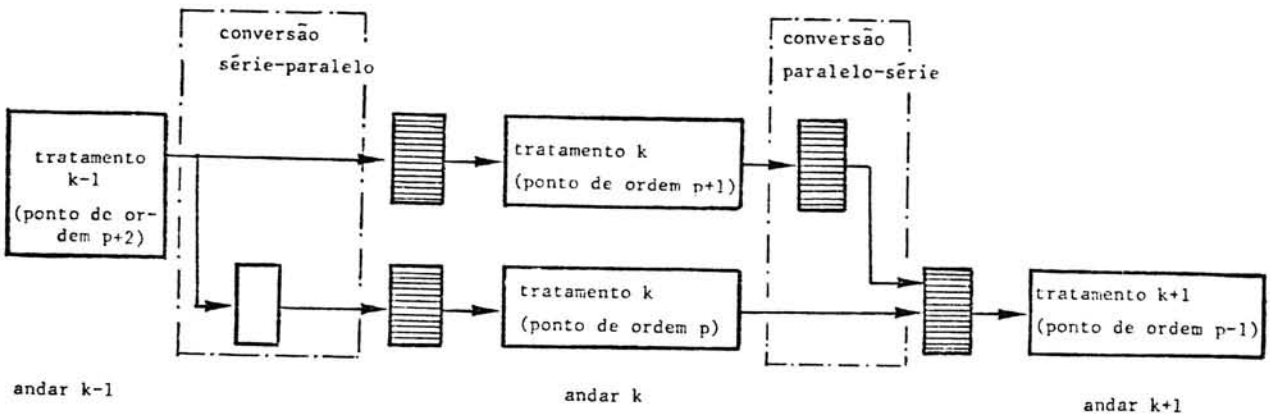


Fig. 4.3 – Um paralelismo adicional ao nível do andar K.

Regra geral teremos:

grau de paralelismo do andar i = (tempo de tratamento do processo elementar i)/t

No caso particular de HELIOS, os tratamentos importantes a executar face ao tempo elementar de 70 ns disponível por andar, criava graves problemas:

— torna-se importante o número de andares da estrutura "pipe-line",

— torna-se igualmente importante o número de processos elementares indecomponíveis cujo tempo de tratamento é superior a 70 ns.

Esta situação exigiu a pesquisa de uma solução complementar que minimizasse este género de problemas.

A técnica utilizada consistiu em aproveitar o tempo total de uma linha video (64 microsegundos) e não o tempo útil de visualização, no nosso caso da ordem de 36 microsegundos [fig. 4.1]. Isto significa que a informação chega ao fim da cadeia "pipe-line" à cadência de um ponto todos os 120 ns (*processo de preparação à visualização*), sendo visualizada posteriormente à cadência imposta de um ponto todos os 70 ns (*processo de visualização*).

Uma solução deste tipo foi adoptada e posta em funcionamento graças a um duplo "tampão" (buffer) que permite o assincronismo entre o processo de preparação à visualização de uma linha e o processo de visualização da mesma. Esta função é realizada ao nível do módulo de visualização [fig. 4.4]

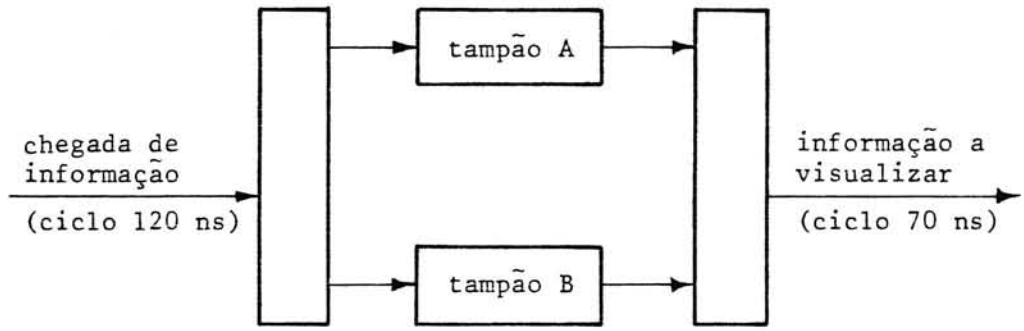


Fig. 4.4 – Esquema simplificado do módulo de visualização.

Resumindo:

— HELIOS utiliza uma estrutura "pipe-line" em que o tempo de tratamento elementar é da ordem de 120 ns [fig. 4.2],

— os andares "pipe-line" que exigam um tempo de tratamento superior a 120 ns comportam um paralelismo adicional [fig. 4.3],

— um módulo de Visualização prevê um assincronismo entre os processos de preparação à visualização e o de visualização propriamente dita [fig. 4.4].

IV. 2.2 – **As tecnologias utilizadas**

Um estudo inicialmente realizado mostrou rapidamente que os tempos em jogo não permitiriam a utilização de microprocessadores, perdendo-se desta maneira todas as facilidades que são oferecidas por tal tecnologia (baixo custo, tempo de desenvolvimento reduzido, pequeno volume do sistema, ...). Como consequência HELIOS comporta onze placas de lógica "hardware", agrupadas electricamente pelo "backplane", cada uma comportando cerca de oitenta circuitos integrados, distribuidas da seguinte maneira:

— 3 placas . . . . . 2 planos de identificação

— 3 placas . . . . . cálculo e memorização de texturas

— 1 placa . . . . . interface, plano de fundo e controlo cursor

— 1 placa . . . . . módulo de controlo e de comunicação

— 2 placas . . . . . módulo cálculo de iluminação

— 1 placa . . . . . módulo de visualização

IV. 2.2.1 - **Os circuitos integrados utilizados**

Foram utilizadas diferentes tecnologias de circuitos integrados, em função das características fundamentais de cada uma delas:

— Os grandes módulos de memória (planos de identificação e memória de texturas) exigem uma tecnologia que apresente uma elevada escala de integração; por este motivo utilizam memórias MOS dinâmicas, caracterizadas por uma grande capacidade (16K e 64K) e pelo baixo consumo.

— Os módulos de cálculo e de controlo, onde a principal preocupação reside no tempo disponível, foram implementados com a ajuda de circuitos integrados da família TTL Schottky (ou simplesmente, TTL S), muito rápidos, mas caracterizados por um consumo bastante elevado.

— Na maior parte das restantes situações utilizou-se a família TTL Low Power Schottky (TTL LS), que exibem simultaneamente um consumo relativamente baixo e tempos de resposta muito interessantes.

IV. 2.3 – **A arquitectura geral de HELIOS**

A figura 4.5 apresenta, de forma simplificada, a arquitectura geral de HELIOS e põe claramente em evidência os seus dois modos de funcionamento:

— modo *acesso exterior,* cujo campo de acção é representado pelo "bus" de dados interno (BDI),

— modo *refrescamento do ecran* representado pelos "bus pipe-line".

O presente estudo toma como base esta estrutura [fig. 4.5] e desenvolve sucessivamente cada um dos seus módulos através da definição e da justificação das suas funções e dos princípios que inspiraram as respectivas realizações.

Considerar-se-á portanto:

— o módulo interface,

— o módulo de comunicação,

— a memória de quadro,

— a tabela das faces,

— o cálculo das texturas,

— o cálculo de iluminação,

— o módulo de visualização,

— o cursor,

— o módulo de controlo.

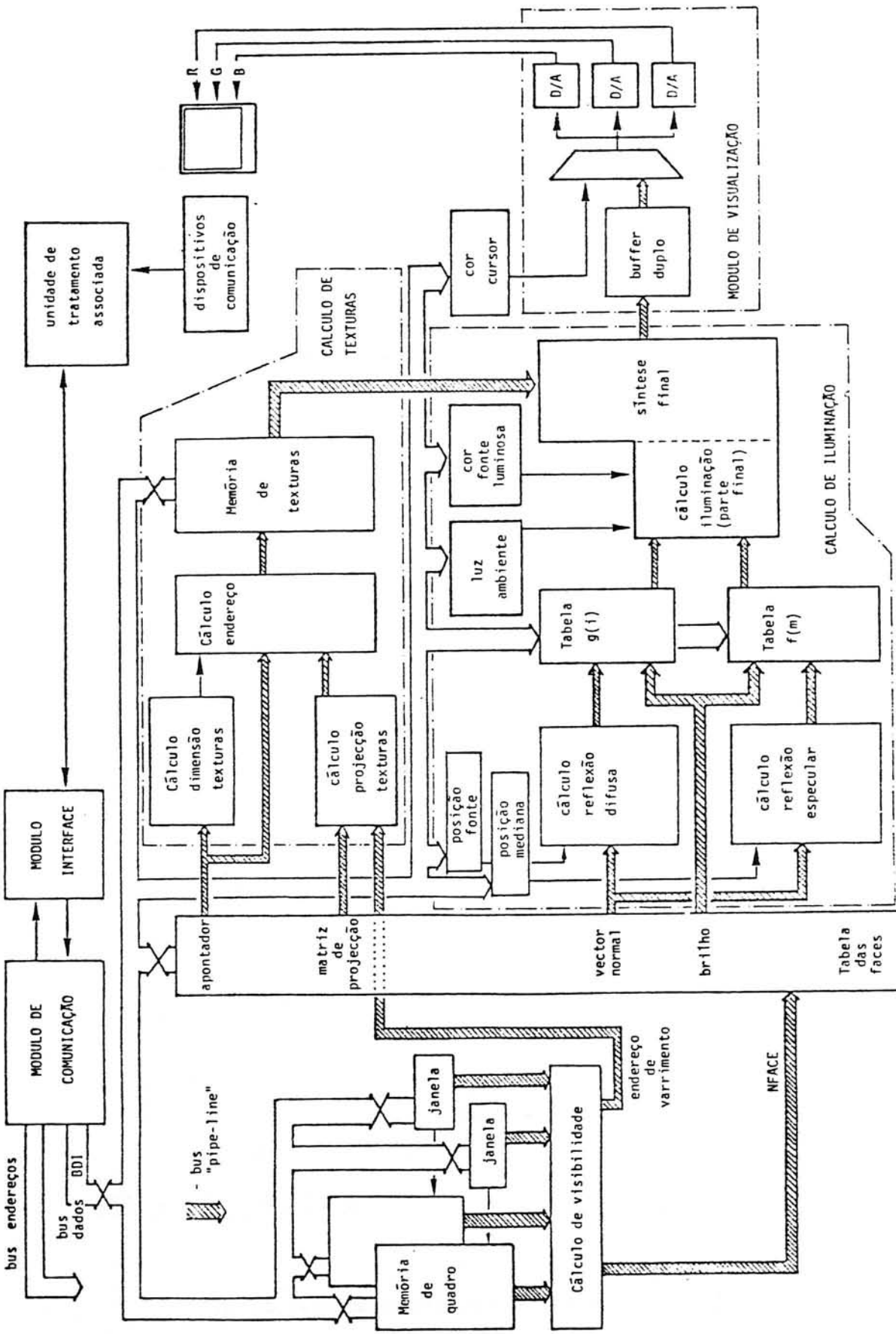


Fig. 4.5 - A arquitectura geral de HELIOS.

IV. 3. O MODULO INTERFACE

O objectivo fundamental do módulo interface [cf. III. 4.3] consiste em tornar compatível a largura do "bus" de dados da unidade de tratamento com o "bus" de 24 bits do módulo de comunicação [fig. 4.6]. A interconexão entre a unidade de tratamento e HELIOS (através da interface) deve ser possível sem que alguma modificação "hardware" seja necessária.

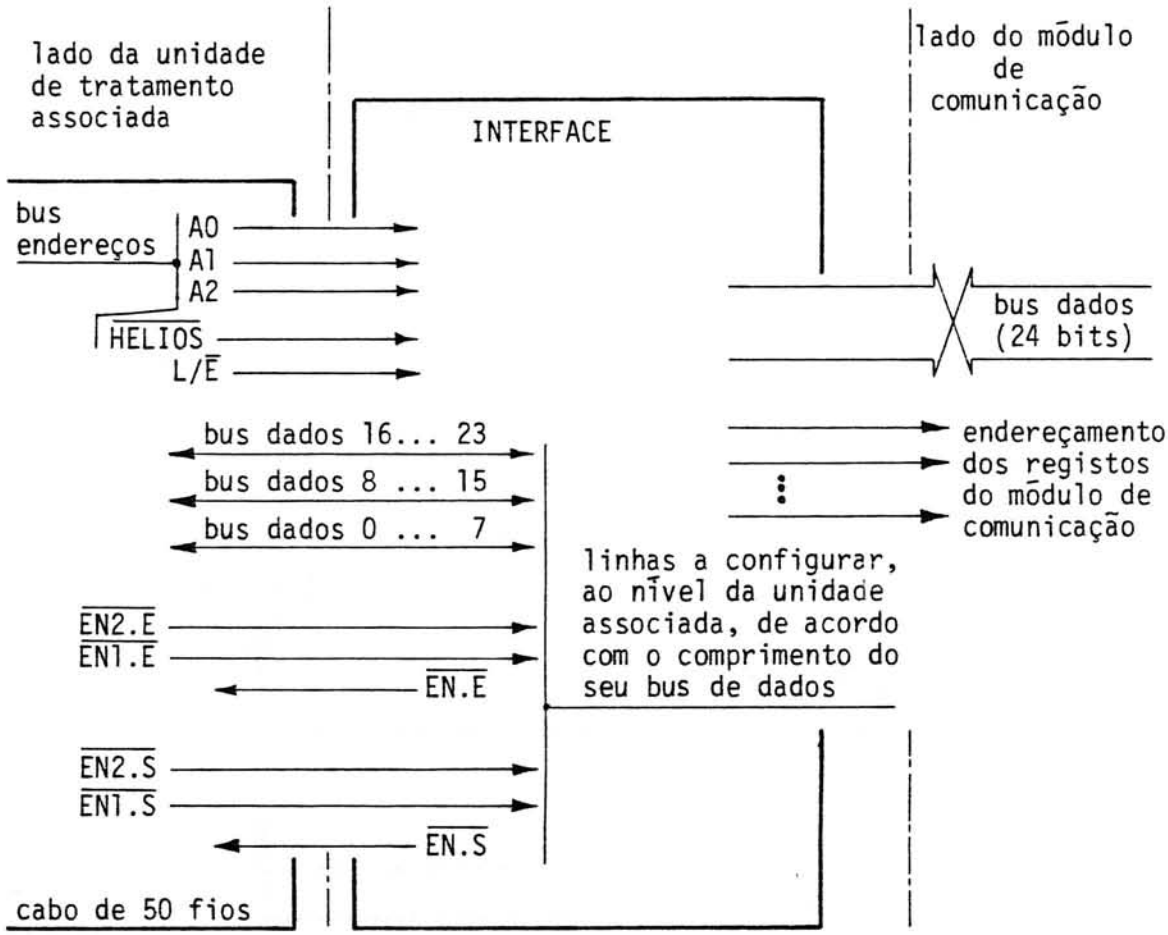


Fig. 4.6 – Ligação de uma unidade de tratamento a HELIOS.

O sinal $\overline{\text{HELIOS}}$ valida a informação presente em A_0, A_1 e A_2, os três bits menos significativos do "bus" de endereços da unidade associada, indicando que decorre um acesso ao terminal.

O sinal $L/\overline{E}$ toma os valores:

1 – acesso em leitura (em direcção à unidade de tratamento)

0 – acesso em escrita (em direcção a HELIOS).

Os restantes sinais são configurados ao nível de unidade associada, conforme a largura do respectivo "bus" de dados.

Examinemos as três configurações possíveis sobre as os esquemas das figuras correspondentes:

— ligação a uma unidade de 8 bits [fig. 4.7]

— ligação a uma unidade de 16 bits [fig. 4.8]

— ligação a uma unidade de 24 (ou mais) bits [fig. 4.9].

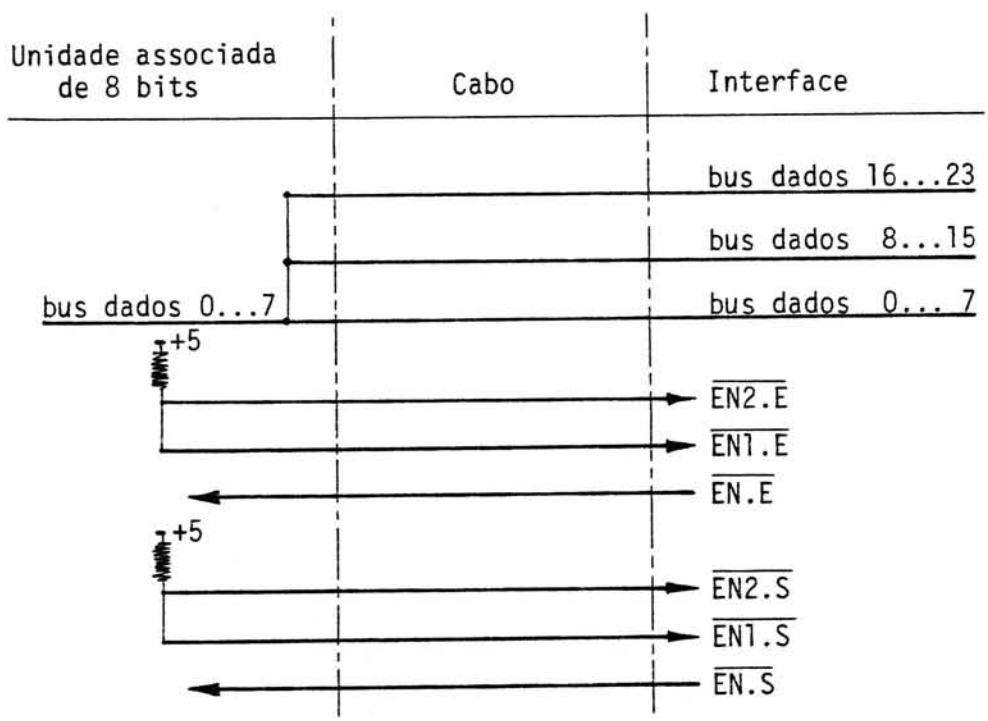


Fig. 4.7 – Ligação de uma unidade de 8 bits.

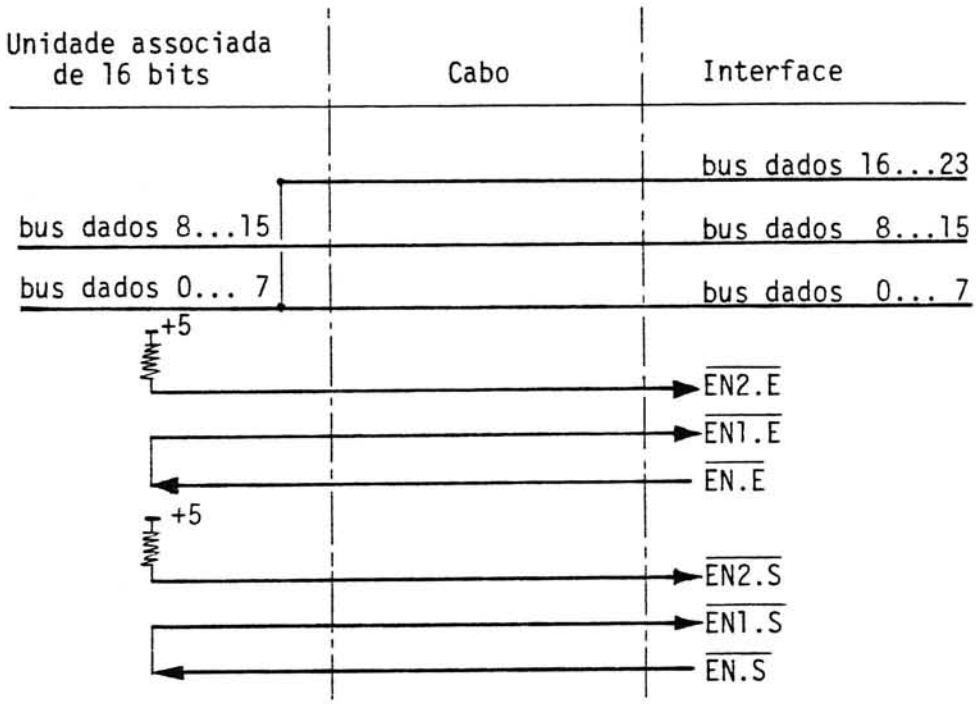


Fig. 4.8 – Ligação de uma unidade de 16 bits.

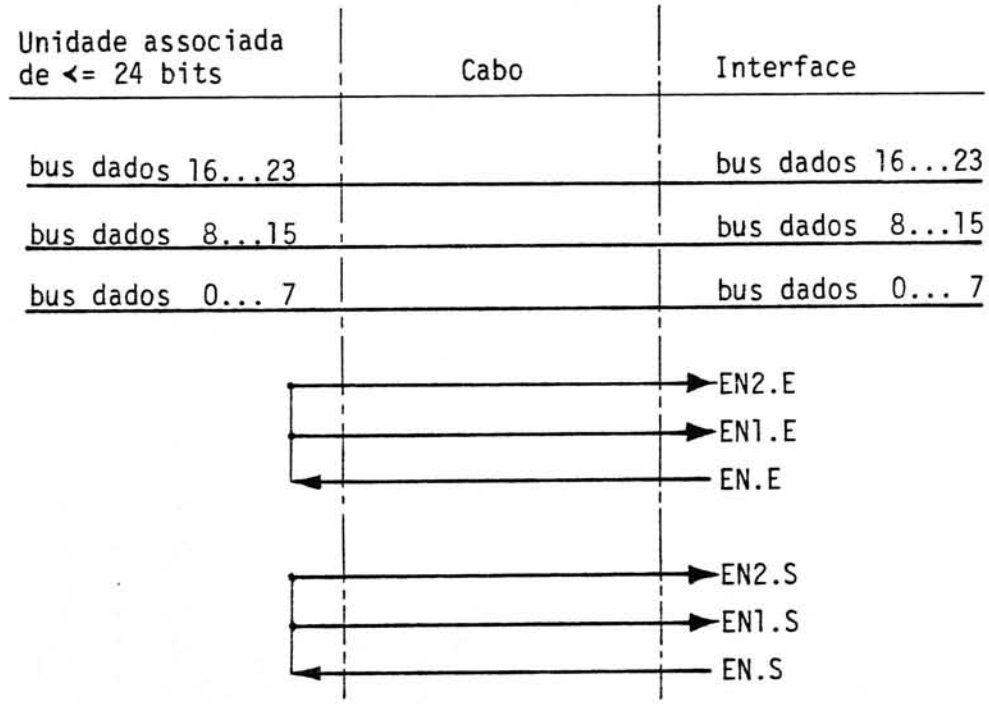


Fig. 4.9 – Ligação de uma unidade de 24 ou mais bits.

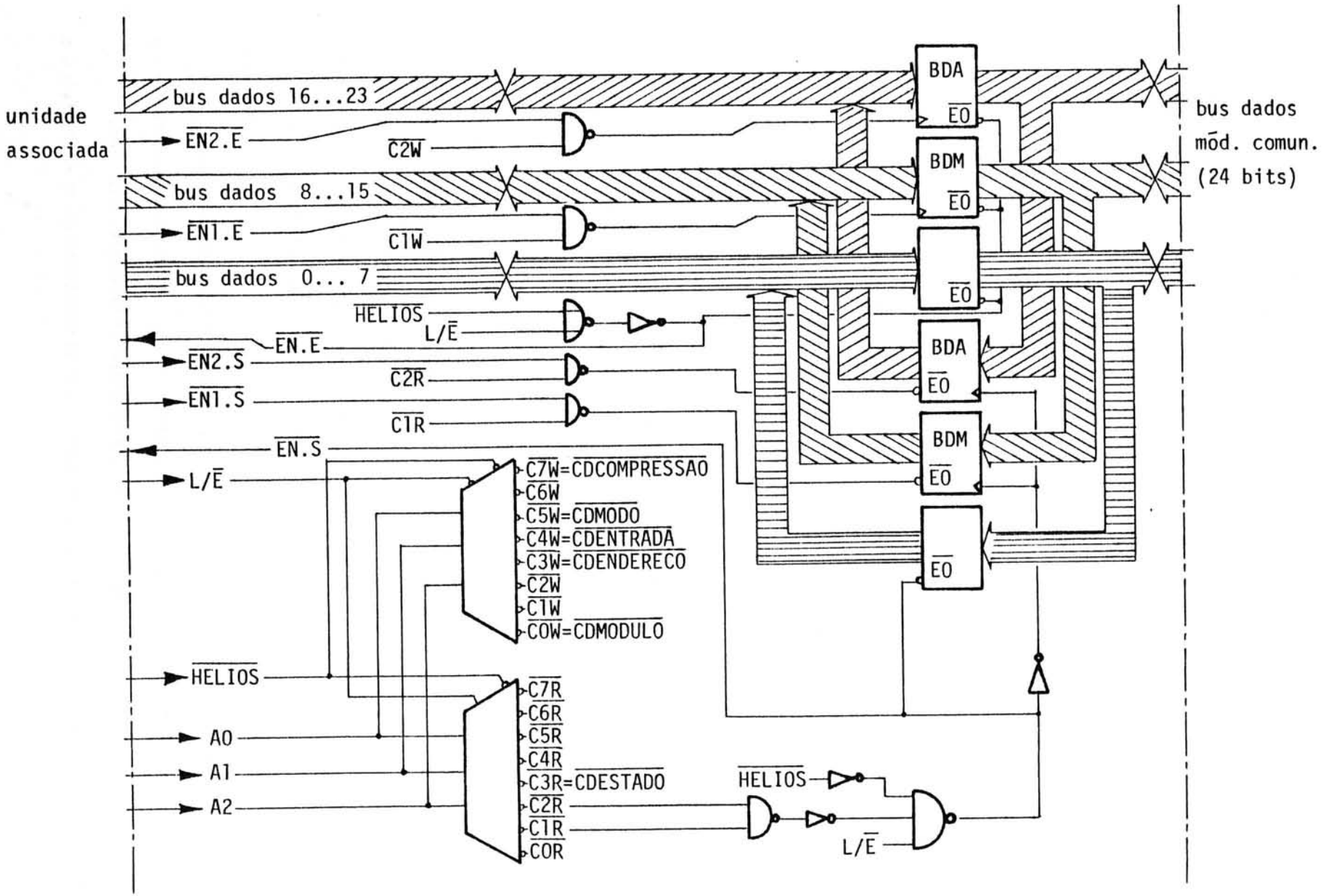


Fig. 4.10 – O princípio de realização do módulo interface.

A figura 4.10 representa simbolicamente o princípio de realização do módulo interface, cujo modo de funcionamento foi já exposto no capítulo precedente [cf. III. 4.3].

IV. 4. O MODULO DE COMUNICAÇÃO

Este módulo estabelece as relações de entrada/saída com o mundo exterior, através do módulo interface; estas relações estão dependentes dos conteúdos dos registos do módulo de comunicação, os quais têm uma acção muito bem definida no processo de comunicação [cf. III, 4.5].

Os acessos exteriores em leitura e escrita foram concebidos de maneira a privilegiarem esta última em relação à primeira.

Torna-se evidente, com efeito, que as operações mais críticas, do ponto de vista do tempo disponível, se situam ao nível do preenchimento das memórias de grande capacidade, em particular, os planos de identificação. A facilidade de visualização sequências animadas em tempo real depende da execução extremamente rápida dos acessos em escrita, o que justifica as possibilidades e a relativa complexidade da sua realização.

O mecanismo de escrita integra as facilidades seguintes:

— toma em linha de conta pedidos de escrita, admitindo níveis de compressão de informação que podem atingir o valor 255,

— toma em linha de conta um novo pedido, enquanto o anterior se encontra em fase de execução,

— executa os acessos propriamente ditos com a possibilidade de escrita em paralelo de até 4 elementos, durante um ciclo de memória (ciclo de aproximadamente 480 ns).

Uma outra característica interessante, comum aos dois tipos de acesso, tem a ver com a possibilidade de os sincronizar com o varrimento TV do ecran [registo MODO, cf. III. 4.5.2].

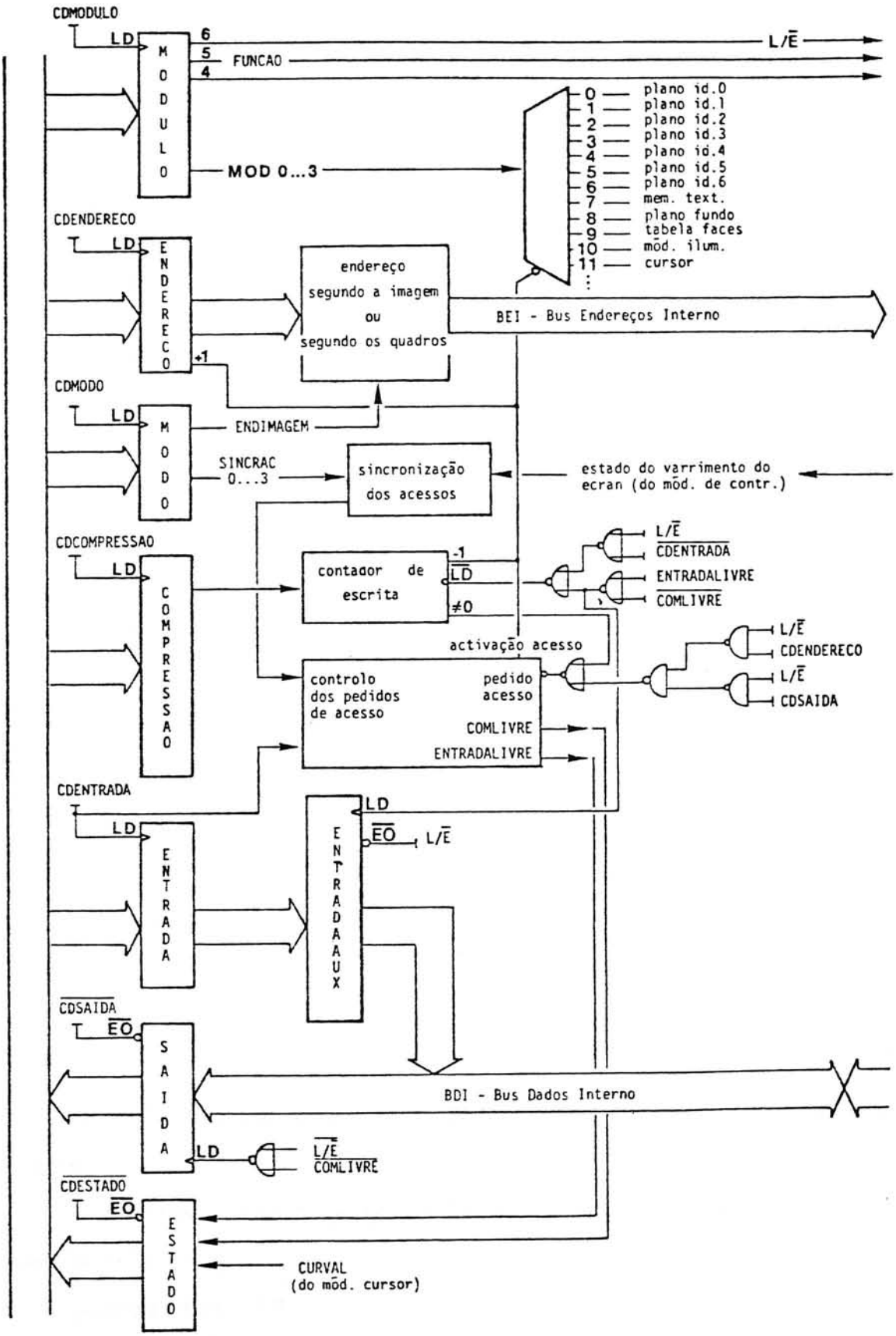


Fig. 4.11 – O princípio de realização do módulo de comunicação.

O conjunto de características indicadas permitem a visualização de sequências animadas em tempo real se:

— a informação chegar ao módulo de Comunicação a um ritmo adequado; contudo, a facilidade de comprimir a informação circulante [registo COMPRESSAO, cf. 4.5.5] diminui significativamente as exigências ao nível da largura de banda do canal de transmissão.

— os planos de identificação admitirem acessos exteriores de maneira quase permanente, sem perturbarem a imagem visualizada.

O princípio de realização deste módulo é representado pela figura 4.11, da qual se põe em relevo o controlo dos pedidos de acesso . As "decisões" rápidas que devem ser tomadas por esta parte do módulo de comunicação requerem uma implementação cuidada, onde vários circuitos integrados Schottky têm funções importantes. Indicamos, como exemplo, que a execução dos algoritmos do processo de comunicação indicados pelas figuras 3.51 e 3.54 [cf. III. 4.5.4 e 4.5.5], em grande parte da responsabilidade deste circuito de controlo, exige que sejam realizadas ciclicamente, todos os 120 ns, um certo número de funções elementares.

IV. 5. A MEMÓRIA DE QUADRO

IV. 5.1 – Os planos de identificação

As principais funções que o utilizador pode solicitar aos planos de identificação são as seguintes:

— memorização e leitura das informações de identificação relativas à imagem visualizada [cf. III. 2.1.1],

— Janela [cf. III. 2.1.3],

— cálculo visibilidade [cf. III. 2.1.4].

IV. 5.1.1 – **A memorização das informações de identificação**

O facto de se ter utilizado os 64 microsegundos de uma linha video para preparação da informação a visualizar [cf. IV. 2.1] vai ocupar totalmente os planos de identificação com o refrescamento do ecran, salvo durante os retornos de quadro (cerca de 3,5 ms por quadro).

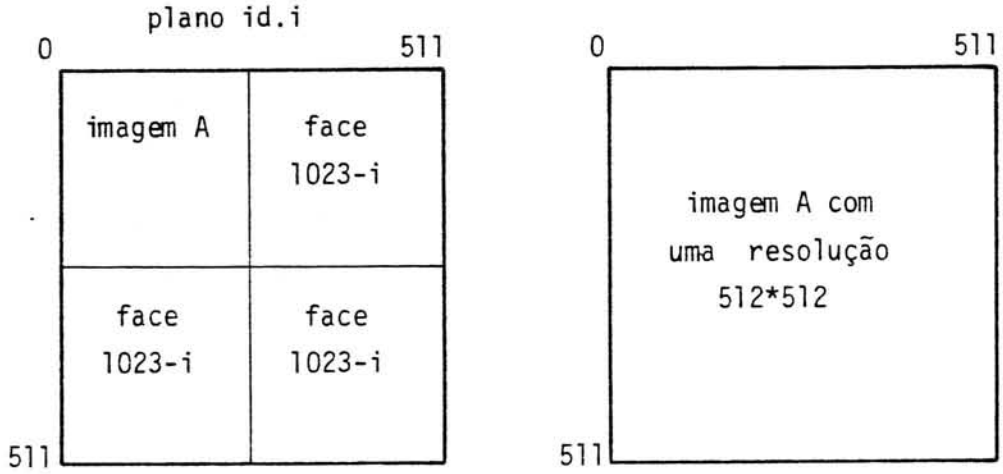
Por outro lado, a criação de sequências animadas em tempo real exige uma disponibilidade quase total dos planos referidos em relação aos acessos vindos do exterior [cf. IV. 4].

Este conflito é menos evidente ao nível dos outros módulos de memória (tabela de visibilidade, tabela das faces, memória de texturas), as quais podem ser modificadas durante os retornos de quadro, sem se perder significativamente em eficiência do sistema. A memória de texturas, por exemplo, não põe problemas de maior (apesar da sua capacidade), pois, após a respectiva inicialização, será muito pouco provável a modificação total do seu conteúdo de uma só vez, mais simplesmente algumas texturas de tempos a tempos.

A realização dos planos de identificação, com uma capacidade 512 × 512 × 10, "recomenda" a utilização de memórias "rápidas" com uma elevada escala de integração, o que dirige a escolha para os circuitos MOS dinâmicos de 64K bits (ciclo da ordem de 235 ns, MOSTEK, MK 4164-10) para imagens 512 × 512, ou de 16K bits (ciclo da ordem de 375 ns, MOSTEK, MK 4116-3) para imagens 256 × 256.

Esta escolha, tendo como critério a resolução da imagem, exige que esta se apresente com uma resolução 512 × 512, mesmo que se utilize as memórias de 16K, esperando que as de 64K se tornem de uso corrente [fig. 4. 12.a]. Pretende-se, desta maneira, assegurar uma passagem fácil das memórias de 16K às de 64K.

A figura 4.12.b. ilustra o resultado obtido com a função Janela de abertura 1 [cf. III. 1.1.3] que permite voltar a uma imagem única com uma resolução 256 × 256.



a. Imagem de resolução 512 × 512.

b. imagem de resolução 256 × 256 obtida graças à função Abertura-janela (A jan = 1).

Fig. 4.12 – Um plano de identificação implementado com memórias de 16K (capacidade total 64K × 10).

Uma outra questão a considerar relaciona-se com o grau de paralelismo a introduzir neste andar "pipe-line"; a versão 64K (ciclo 235 ns) requer apenas um grau de paralelismo igual a 2, sendo igual a 4 para a versão 16K (ciclo de 375 ns).

Para se preservar uma compatibilidade entre a utilização das memórias de 16K e 64K, escolheu-se um grau de paralelismo igual a 4.

Com base no que se indicou, decompõe-se os 64 microsegundos de uma linha video em 128 ciclos [fig. 4.13], correspondendo a cada um 4 pontos consecutivos (na horizontal) da imagem.

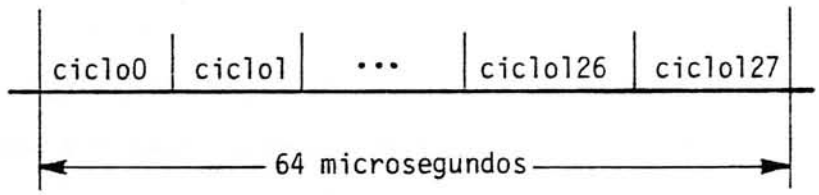


Fig. 4.13 – Decomposição do tempo correspondente a uma linha video em 128 ciclos.

O princípio de funcionamento inerente às memórias dinâmicas escolhidas exige, pelo menos, 128 ciclos de "refrescamento" todos os 2 ms.

A técnica utilizada para resolver este problema resumiu-se a juntar 5 ciclos de refrescamento a cada linha video [fig. 4.14].

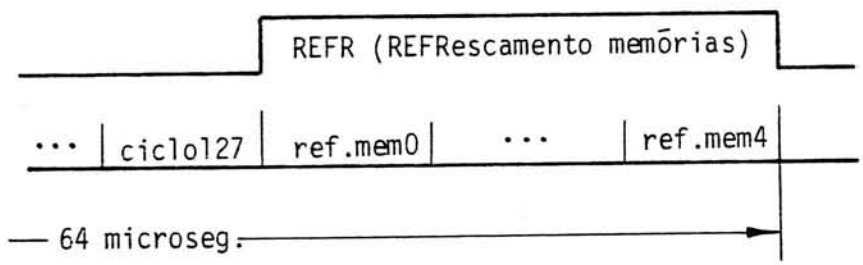


Fig. 4.14 – Decomposição do tempo correspondente a uma linha video em 133 ciclos.

Quanto aos acessos vindos do exterior, ficam limitados aos tempos de retorno de quadro, na versão (actual) que utiliza memória de 16K com um ciclo de 375 ns. Pelo contrário, numa futura versão que empregue as memórias de 64K (ciclo de 235 ns), cada um dos 133 ciclos referidos podem ser decompostos em dois; um permitindo os acessos internos e o outro considerando os acessos exteriores [fig. 4.15].

Nesta situação os planos de identificação ficam permanentemente acessíveis sem que se note qualquer distúrbio na imagem visualizada; este facto satisfaz à condição imposta pela criação de sequências animadas em tempo real [cf. IV. 4].

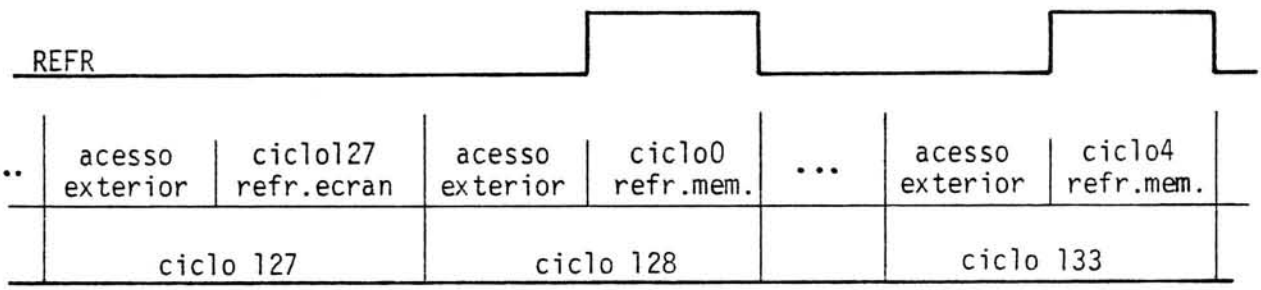


Fig. 4.15 – Diagrama de tempos de uma linha video.

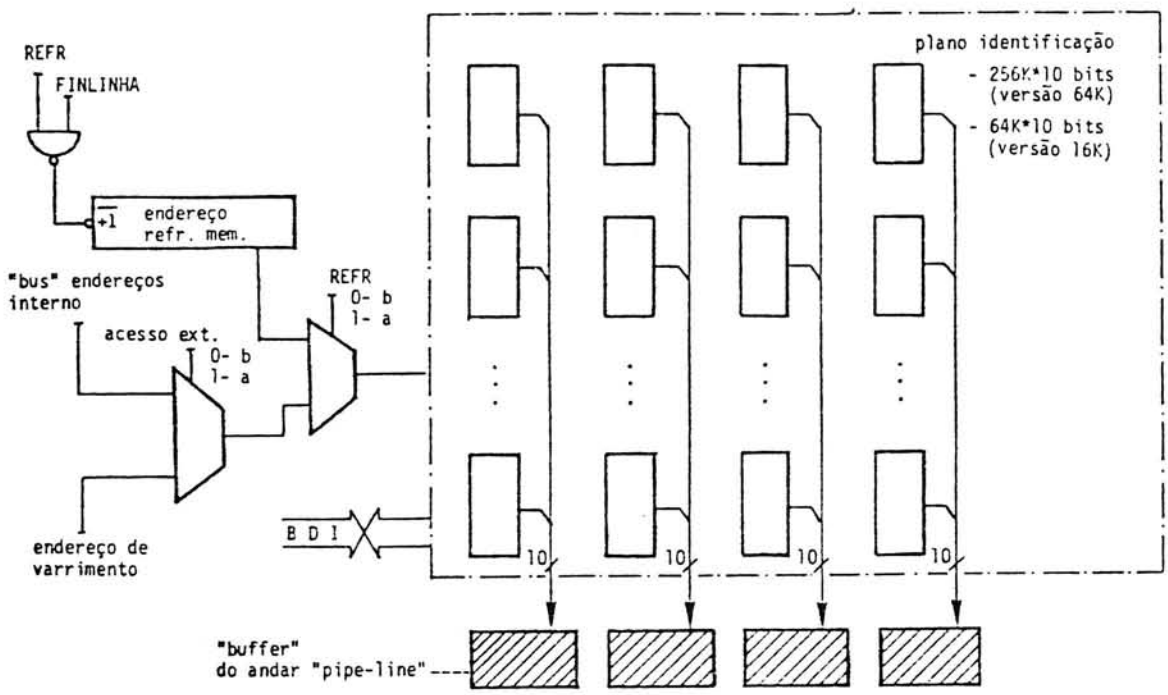


Fig. 4.16 – A organização geral de um plano de identificação.

Saliente-se que o paralelismo introduzido no andar do plano de identificação [fig. 4.16] requer uma conversão "paralelo-série", realizada com base em registos com saídas de "3 estados", a funcionarem também como memória de "tampão" (buffer) do andar "pipe-line" [fig. 4.17].

	versão 16K	versão 64K
capacidade	64K*10 bits	256K*10 bits
acessos exteriores	durante os retornos de quadro	sempre
grau de paralelismo	4	4
	conversão imediata entre as duas versões	

Fig. 4.17 – O princípio de realização da conversão "paralelo-série" num plano de identificação.

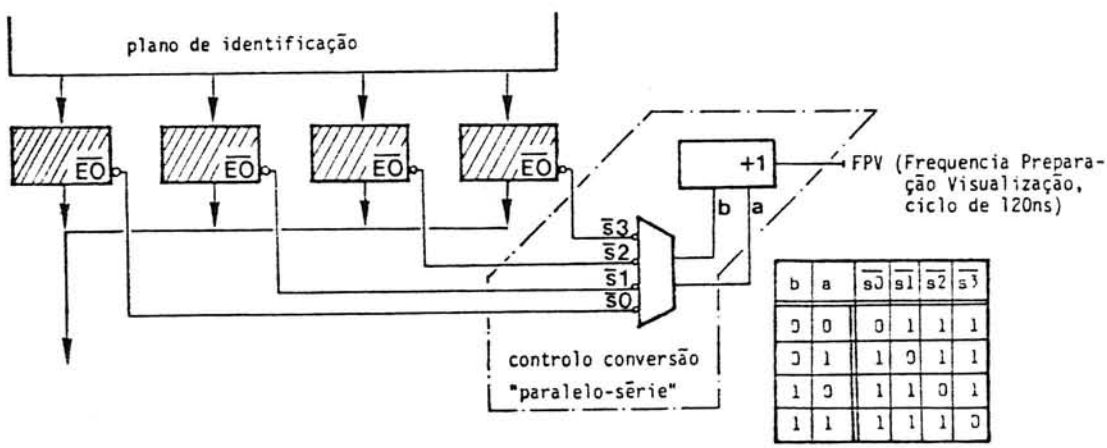
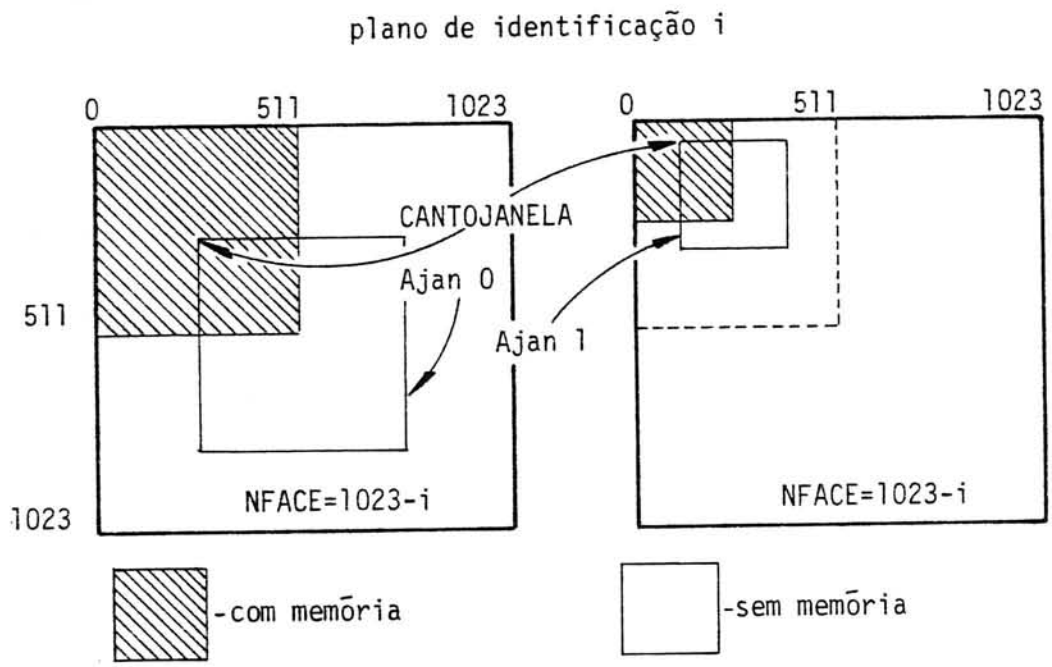


Fig. 4.18 – Algumas características das versões 16K e 64K do plano de identificação.

As considerações apresentadas anteriormente são resumidas no quadro da figura 4.18.

IV. 5.1.2 – **A função janela**

No espaço virtual 1024 × 1024 de um plano de identificação [fig. 4.19] é possível definir uma janela de visualização a partir do seu canto superior esquerdo e da sua abertura [cf. III. 2.1.3].



a. versão 64K

b. versão 16K

Fig. 4.19 – O espaço virtual do plano de identificação.

— Definição do canto superior esquerdo

O princípio de realização desta função é indicado na figura 4.20 e baseia-se num registo de 20 bits., (registo CANTOJANELA) acessível através do módulo de comunicação. O conteúdo deste registo contém a coluna e a linha que definem o ponto de partida para o varrimento do plano de identificação.

Como esta função é independente de plano para plano, existirá obviamente um registo CANTOJANELA em cada um deles.

Saliente-se ainda na figura 4.20 o circuito que controla a conversão "paralelo-série" já referida [cf. IV. 5.1.1], o qual se responsabiliza igualmente pela situação que decorre do endereço de varrimento alcançar a zona "ausência de memória"; nesta altura a face identificada por NFACE = 1023-i (para o plano de identificação i) é considerada como a única a visualizar.

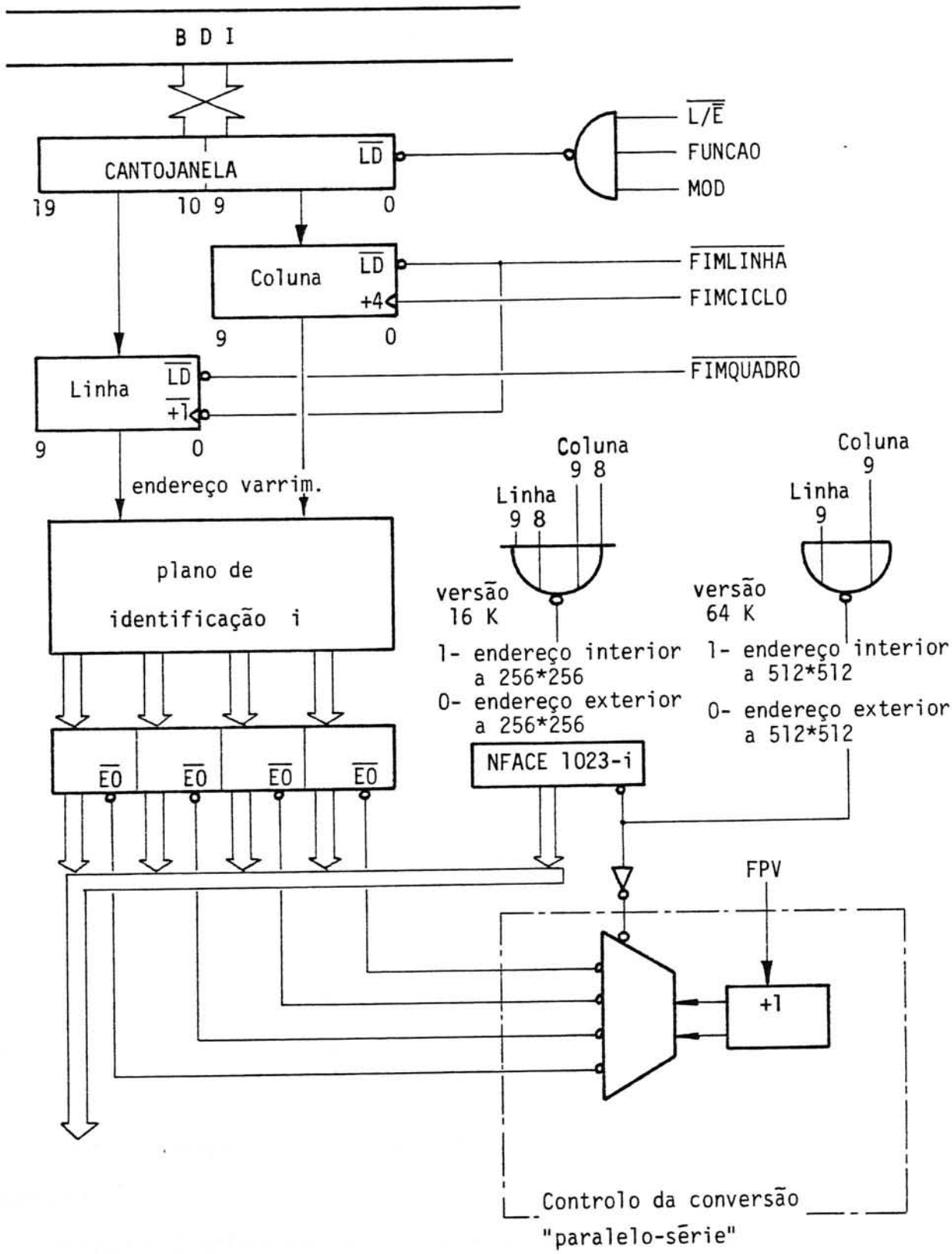


Fig. 4.20 – O princípio de realização do plano de identificação; a função CANTOJANELA e o controlo da conversão "paralelo-série".

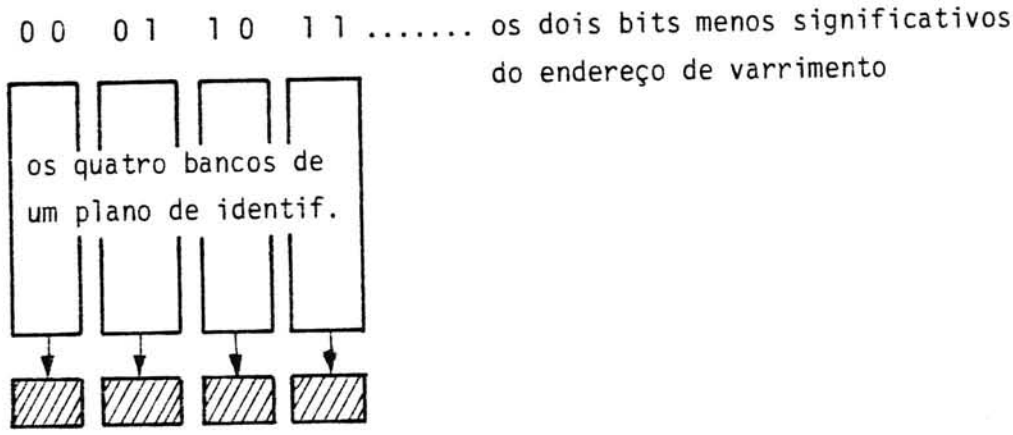


Fig. 4.21 – Definição implícita dos dois bits menos significativos do endereço de varrimento de um plano de identificação.

Vejamos agora outro pormenor importante: os dois bits menos significativos do endereço de varrimento do plano de identificação não são considerados, uma vez que esta informação está implícita na leitura de 4 pontos em paralelo [fig. 4.21], motivada pela necessidade de um paralelismo de grau 4 neste andar "pipe-line".

— Definição da Abertura da Janela

O quadro da figura 4.22 resume a acção da função Abertura-Janela.

Pode-se constatar a influência desta função sobre a frequência de varrimento horizontal do plano de identificação, assim como sobre a frequência de conversão "paralelo-série". Obviamente esta influência repercute-se de maneira análoga sobre o varrimento vertical daquele plano.

Em todos estes casos, as frequências referidas são divididas por um número igual a abertura + 1.

A figura 4.23 ilustra a princípio de realização desta função, onde se encontra um registo ABERTURA, de 4 bits, acessível através do módulo de comunicação.

	prep. à visualização			visualização		
	ciclo t	ciclo t+1	ciclo t+2	ciclo x	ciclo x+1	ciclo x+2
Abertura0	A B C D	E F G H	I J K L	A B C D	E F G H	I J K L
Abertura1	A B C D	A B C D	E F G H	A A B B	C C D D	E E F F
Abertura2	A B C D	A B C D	A B C D	A A A B	B B C C	C D D D

Fig. 4.22 – O princípio de funcionamento da função Abertura-Janela.

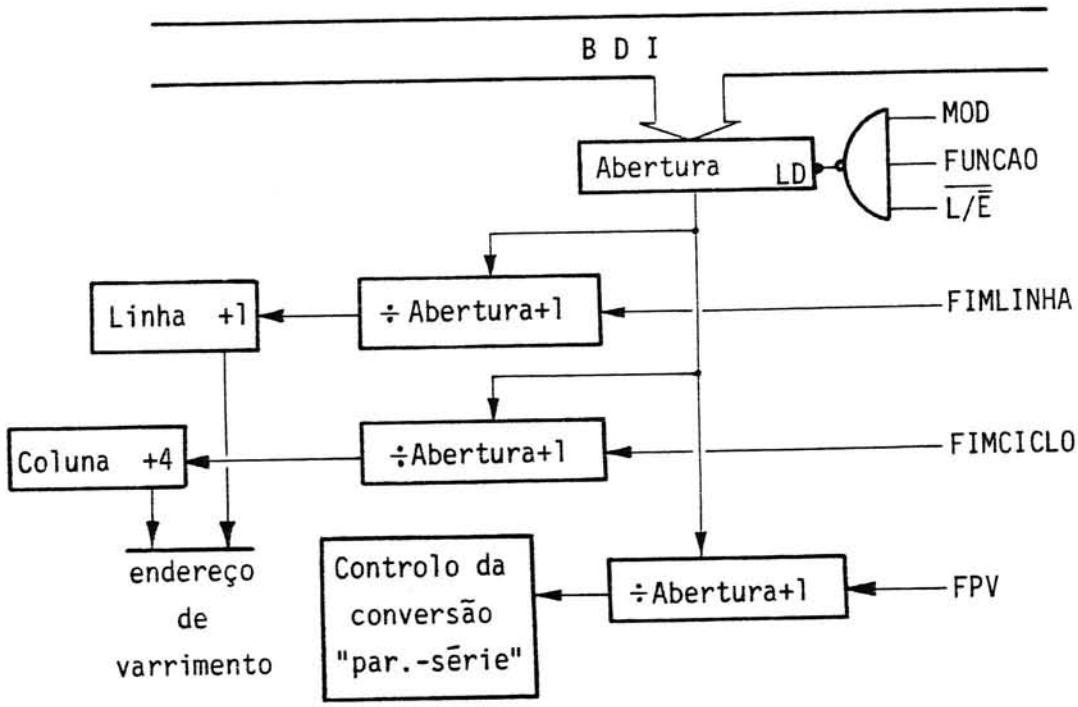


Fig. 4.23 – O princípio de realização da função Abertura-Janela.

IV. 5.1.3 – O cálculo de visibilidade

De acordo com as prioridades de visualização associadas a cada plano de identificação [cf. III. 2.1.4], este módulo responsabiliza-se por colocar no "bus" comum a todos os módulos da memória de quadro ("bus pipe-line"), um valor NFACE que identifique a face à qual pertence o ponto visível, assim como o endereço de varrimento do plano onde este ponto se encontra memorizado [fig. 4.24].

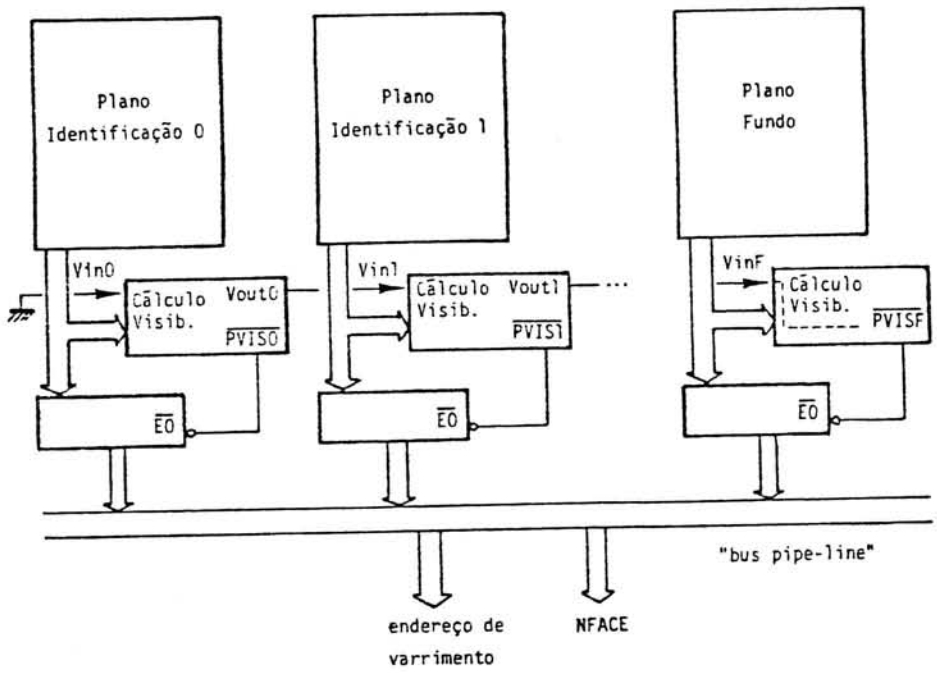
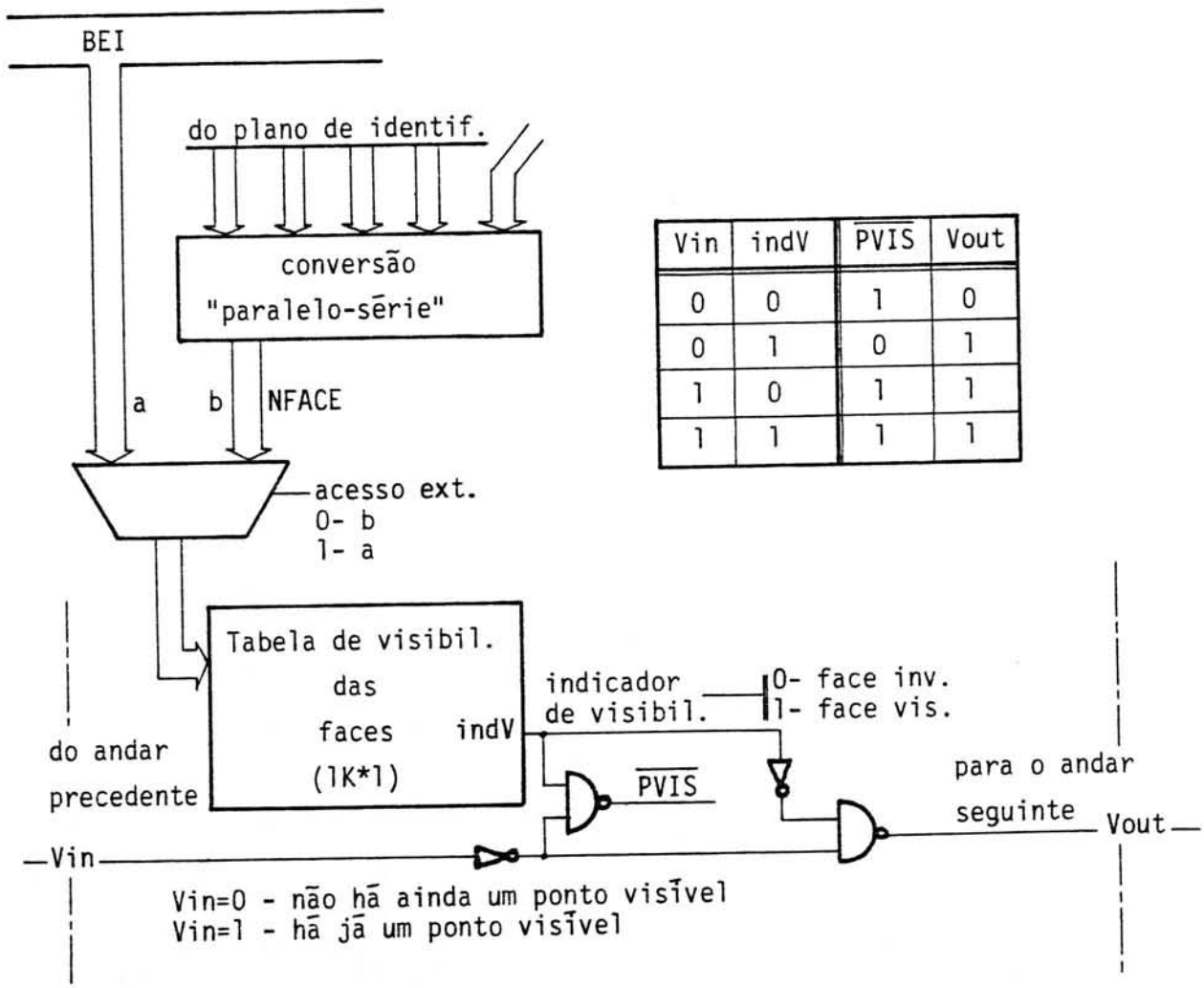


Fig. 4.24 – A missão do cálculo de visibilidade.

A realização do cálculo de visibilidade encontra-se distribuida pelos diferentes módulos que compõem a memória de quadro, como se indica na figura 4.25.

O tempo disponível de 120 ns por ponto exige uma lógica à base de circuitos integrados Schottky que, apesar de simples, deve caracterizar-se por possuir uma resposta bastante curta. Com efeito, numa informação de visibilidade propaga-se desde o plano de identificação zero até ao plano de fundo, passando pelos restantes planos de identificação.

A figura 4.26 resume, finalmente, o princípio de realização do módulo memória de quadro, assim como as interconexões entre os seus diferentes componentes.



Vin	indV	$\overline{PVIS}$	Vout
0	0	1	0
0	1	0	1
1	0	1	1
1	1	1	1

Fig. 4.25 – O princípio de realização do cálculo de visibilidade ao nível de um plano de identificação.

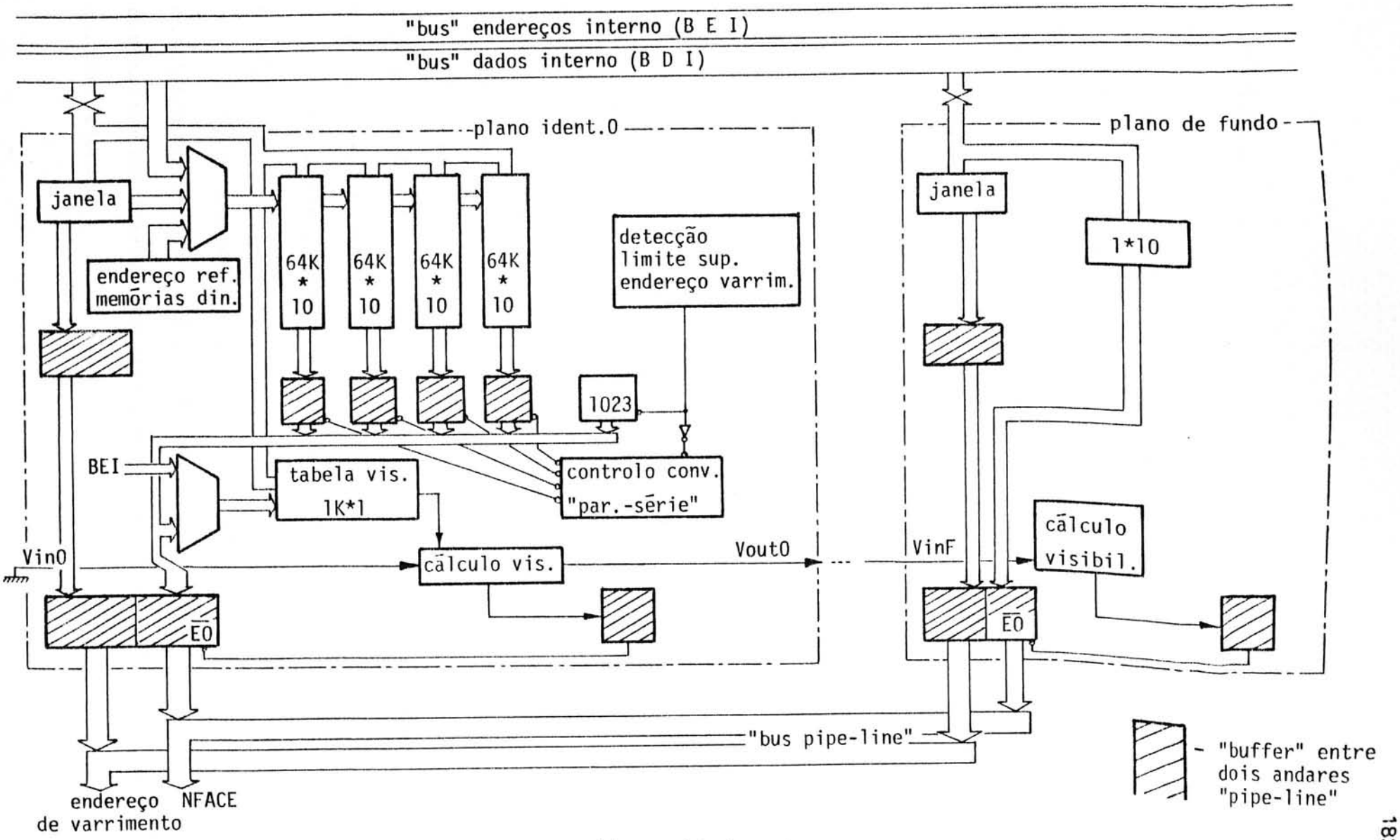


Fig. 4.26 – Organização geral da memória de quadro.

IV. 5.2 – **O plano de fundo**

O plano de fundo é um caso muito particular de um plano de identificação, uma vez que a sua capacidade se limita a uma única face, a qual afectará todo o ecran [cf. III. 2.2]. Além disso, o cálculo de visibilidade reveste aqui uma forma extremamente simplificada [fig. 4.24], de acordo com o que já indicámos no capítulo precedente [cf. III. 2.2].

Estas características em conjunto tiveram uma influência fundamental na simplicidade do seu princípio de realização [fig. 4.26].

IV. 6. **A TABELA DAS FACES**

A entidade de base manipulada por HELIOS é a *FACE 3D* codificada pelo número de identificação correspondente – NFACE –.

Após a determinação do NFACE correspondente à face visível (resultado do cálculo de visibilidade ao nível da memória de quadro), recupera-se os valores dos parâmetros que lhe estão associados, a saber; o tipo de material (brilho e apontador de textura) e referencial 3D (matriz de projecção e vector normal) [cf. III. 3.1].

À tabela das faces cabe a responsabilidade de comunicar ao processo de preparação à visualização as informações referidas, devendo, além disso, memorizar dinamicamente todas as modificações que lhes digam respeito.

A realização desta tabela, baseada em memórias TTL S (ciclo da ordem de 50 ns) com uma organização 1K × 1, apresenta uma estrutura que é ilustrada na fig. 4.27.

Fisicamente esta tabela está distribuida por outros módulos, de acordo com as funções específicas respectivas. Assim, no cálculo de iluminação encontram-se as partes correspondentes ao brilho e ao vector normal, enquanto que o apontador e a matriz de projecção estão associadas ao cálculo de texturas.

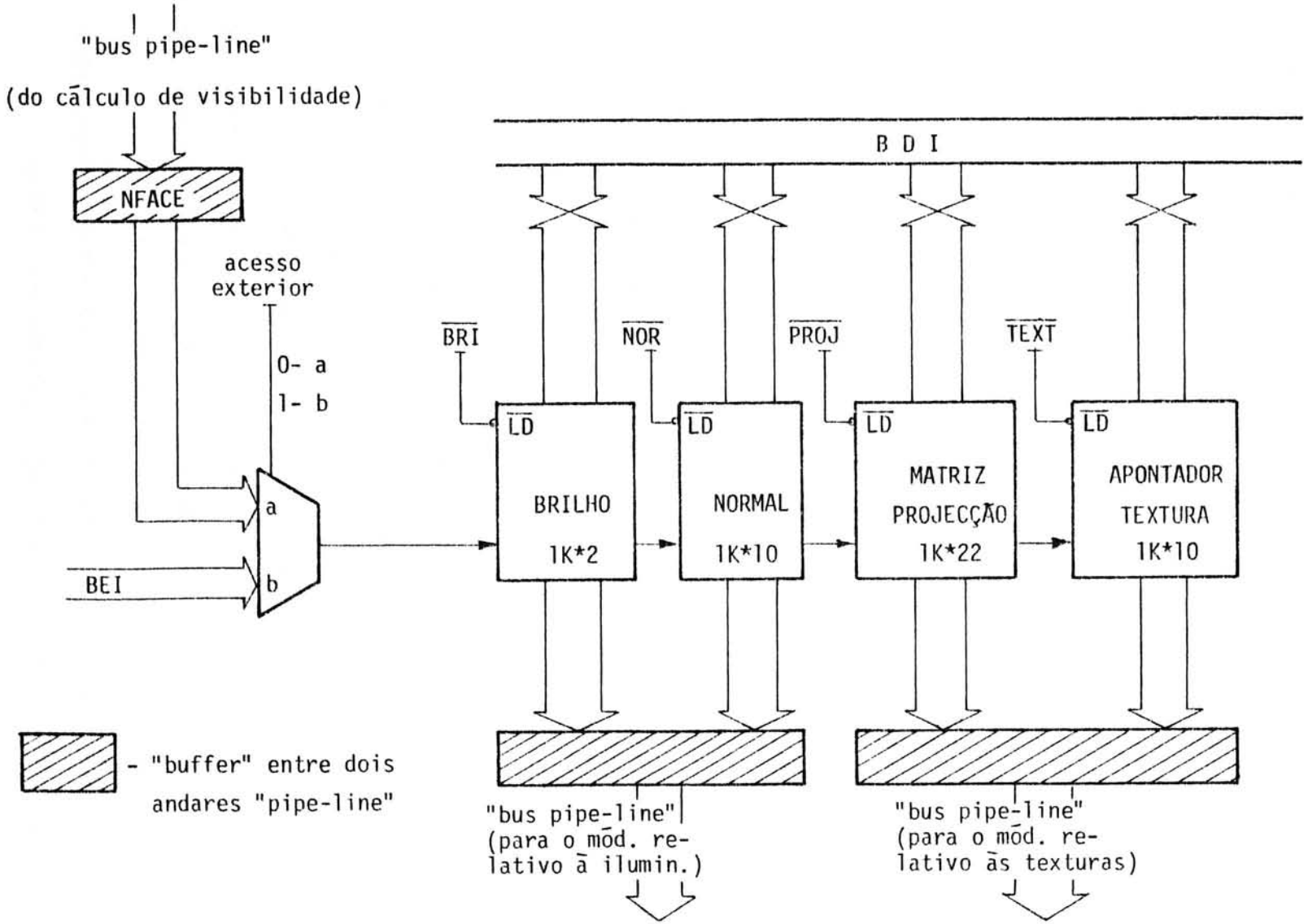


Fig. 4.27 – A arquitectura geral da tabela das faces.

IV. 7. O CÁLCULO DE TEXTURAS

HELIOS admite, ao nível da memória de texturas, a memorização de matrizes coloridas, com dimensões diversas, que simulam o aspecto visual de vários materiais [cf. III. 3.2].

O cálculo de texturas deve determinar, por cada ponto da imagem a visualizar, o endereço da memória de texturas que corresponde à informação colorida desejada.

As informações chegadas ao módulo cálculo de texturas têm origem na tabela das faces (apontador de texturas e matriz de projecção) e no cálculo de visibilidade (endereço de varrimento da memória de quadro) e são tomadas em linha de conta por dois processos que decorrem em paralelo [fig. 4.28]:

— o cálculo da dimensão da textura, definida implicitamente no apontador respectivo [cf. III. 3.1.2],

— o cálculo da projecção [cf. III. 3.1.4].

Estes dois processos não apresentam igual duração (a determinação da dimensão é muito mais curta que o cálculo da projecção) o que justifica a presença de alguns atrasos de sincronização na sub-cadeia "pipe-line" do processo mais rápido. Este procedimento garante que a informação no final das duas sub-cadeias se apresente em correspondência.

Em seguida, o cálculo do endereço toma em consideração a dimensão da textura, o apontador respectivo e o endereço de varrimento do plano de identificação, depois deste último ter sido submetido à operação de projecção [fig. 4.29].

Esta memória apresenta características análogas às de um plano de identificação, pelo que exige também um paralelismo de grau 4 ao nível do andar "pipe-line" no qual está inserida. Este facto traduz-se por duas implicações importantes:

— a realização deste módulo deve incluir uma conversão "série-paralelo"; os endereços correspondentes a quatro pontos consecutivos da imagem são necessários simultaneamente.

— dos 18 bits calculados [fig. 4.29], utilizam-se apenas 16, sendo desprezados os dois menos significativos; de facto, estes dois bits são definidos implicitamente numa leitura em paralelo de quatro pontos.

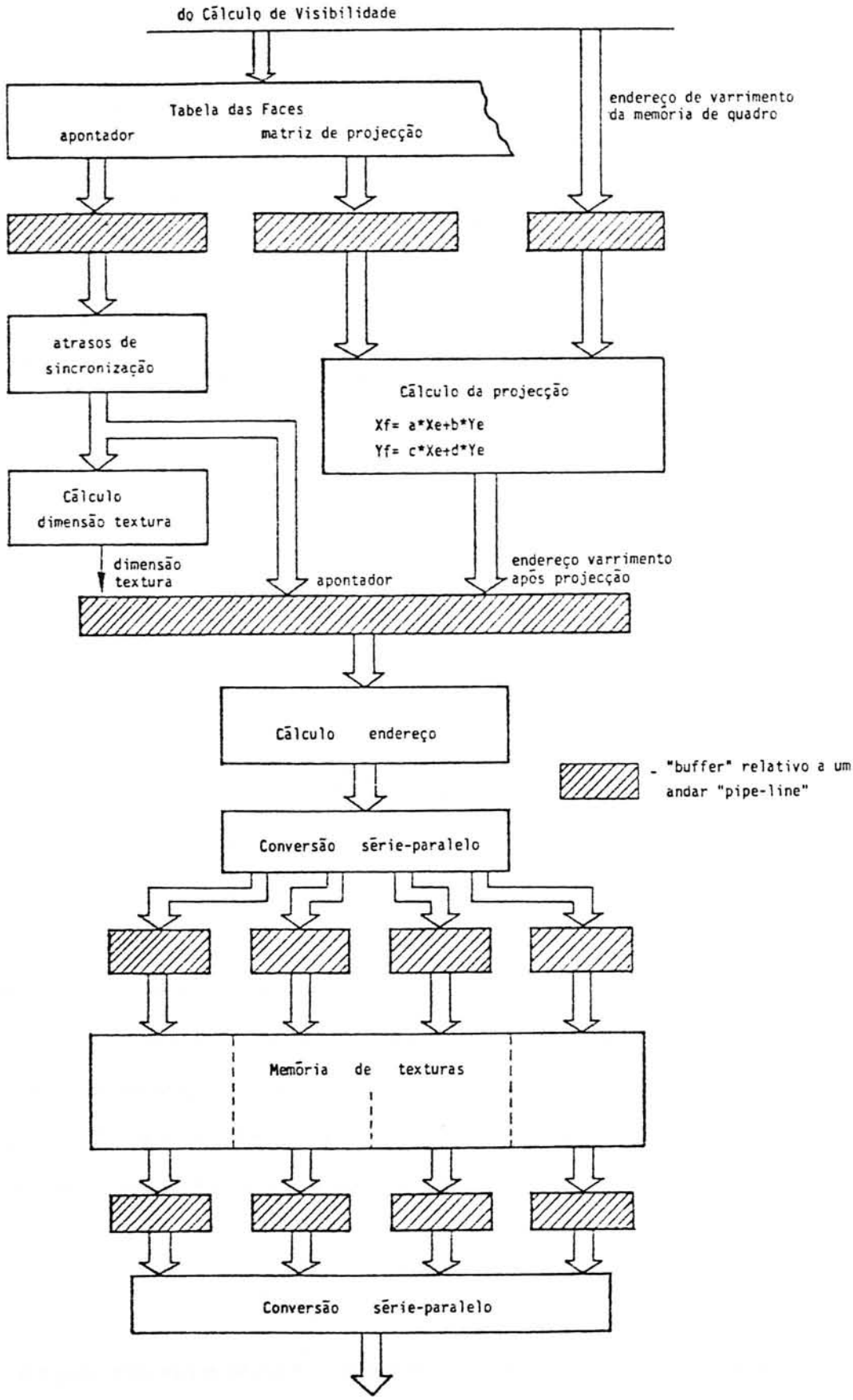


Fig. 4.28 – O cálculo de texturas.

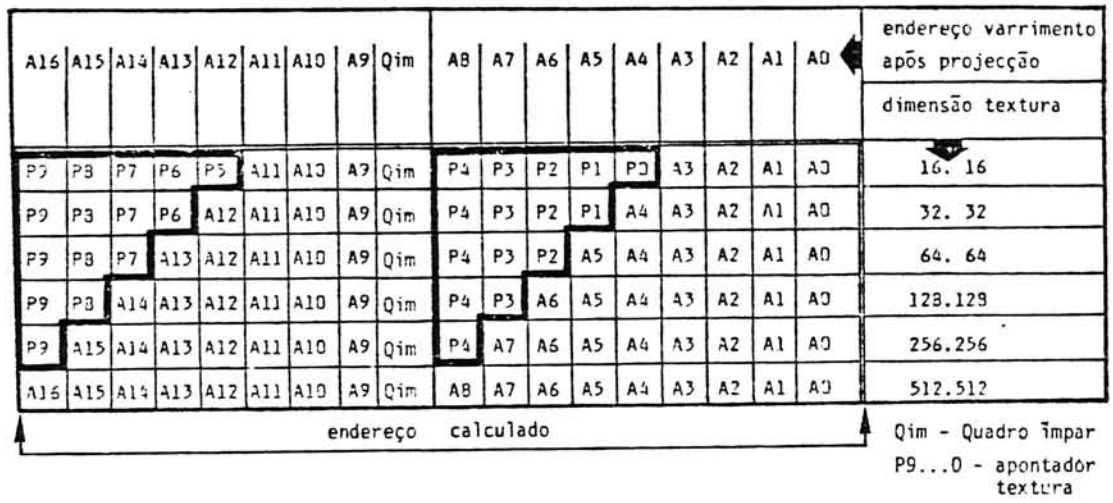


Fig. 4.29 – A definição do endereço a calcular.

Esta maneira de proceder traduz-se por pequenos defeitos na imagem visualizada, quando se utilizam texturas projectadas cujos segmentos unicolores horizontais não apresentem um comprimento múltiplo de 4.

Este problema pode ser minimizado com o emprego de memórias rápidas, como acontecerá com as dinâmicas de 64K [cf. IV. 5.1.1]. Neste caso, a condição imposta ao comprimento dos segmentos unicolores horizontais passa a ser um múltiplo de 2.

Uma outra hipótese que eliminaria por completo este problema, exige que se utilizem memórias de ciclo inferior a 120 ns o que, no quadro da tecnologia actual, limitaria a capacidade da memória de texturas a valores muitíssimo mais pequenos.

A figura 4.30 esquematiza o princípio de realização do módulo em questão.

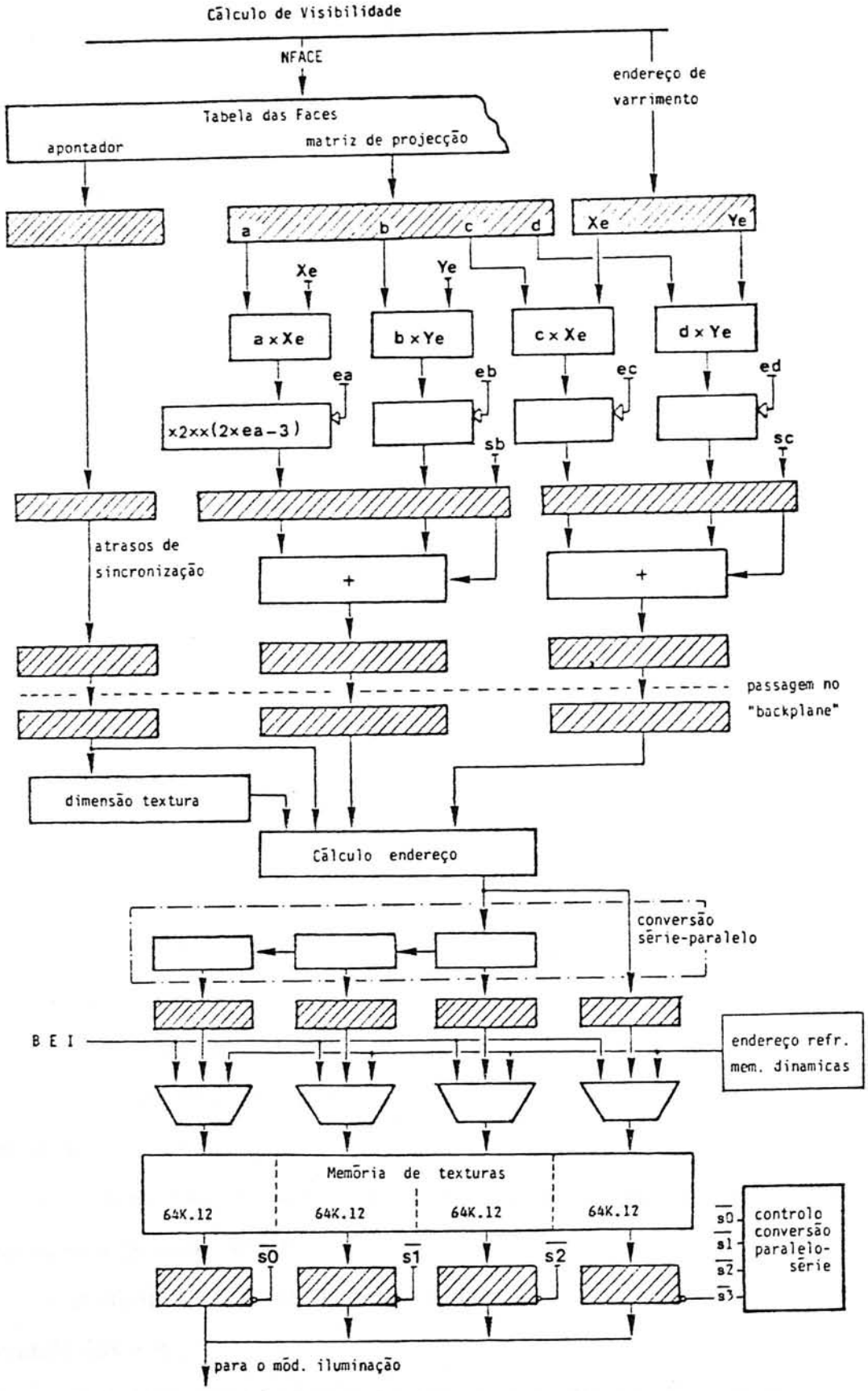


Fig. 4.30 – O princípio de realização do módulo cálculo de texturas.

IV. 8. O CÁLCULO DE ILUMINAÇÃO

A consideração em tempo real dos parâmetros correspondentes a uma fonte luminosa colorida e a uma luz ambiente difusa [cf. III. 3.3] não oferece uma margem de tempo folgada para a execução dos cálculos respectivos. Torna-se assim evidente que a realização deste módulo passa necessariamente pela utilização de tabelas que implementem certas funções.

O problema, ao nível da realização, centrou-se na pesquisa de um modelo de iluminação que se apoiasse em parâmetros relativamente independentes, permitindo minimizar, desta maneira, a dimensão das tabelas a utilizar.

O modelo escolhido [cf. III. 3.3] considera, no que respeita à face a iluminar, o seu vector normal e o brilho do material que a compõe; quanto à fonte luminosa considerada pelo terminal, o modelo toma atenção a sua posição e cor, assim como a intensidade da luz ambiente difusa.

Com a preocupação de se simplificar o processo de cálculo, fica para o utilizador a responsabilidade de comunicar ao terminal a posição mediana entre o observador e a fonte luminosa.

A arquitectura geral do módulo cálculo de iluminação é ilustrada pela figura 4.31, que põe em relevo:

— os parâmetros acessíveis ao utilizador,

— os dois processos que decorrem em paralelo, relativos aos cálculos da reflexão difusa e da reflexão especular,

— os diferentes andares "pipe-line" utilizados,

— os atrasos de sincronização entre os processos de cálculo de texturas (mais lento) e de cálculo de iluminação,

— as tabelas $h(x) = 1 + \cos(x)$ implementados à base de PROM's (ciclo da ordem de 50 ns), com uma organização 32×8,

— as tabelas f(m) e g(i) implementadas à base de RAM's (ciclo de 40 ns), com uma organização 256×4,

— os multiplicadores binários que tiveram uma influência fundamental na precisão adoptada nos cálculos.

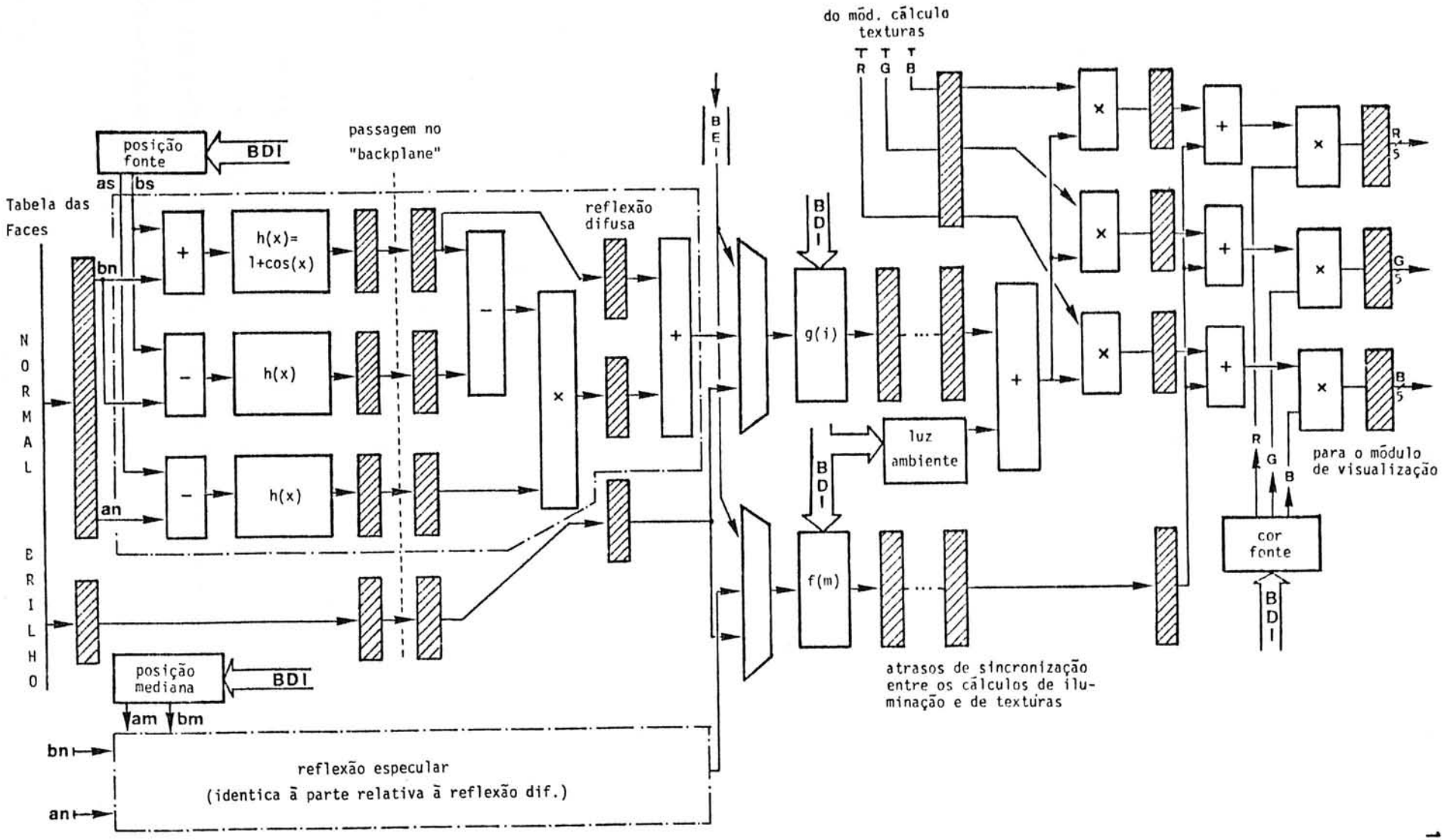


Fig. 4.31 – A arquitectura geral do módulo cálculo de iluminação.

IV. 9. O MÓDULO DE VISUALIZAÇÃO

Este módulo assegura a compatibilidade entre as frequências de trabalho do processo de preparação à visualização (realizado no seio dos módulos já descritos) e o processo de visualização propriamente dito [cf. III. 3.4].

Como já referimos, este último processo apresenta um ciclo (por ponto) da ordem de 70 ns, que lhe é imposto pela frequência de varrimento de um monitor TV. Por seu turno, o ciclo (também por ponto) do processo de preparação à visualização apresenta-se com uma duração de 120 ns [cf. IV, 2.1].

O princípio de realização do módulo de visualização [fig. 4.32] põe em relevo duas memórias, cada uma com uma capacidade 512 × 15, que designaremos por "tampão" (buffer) de linha. Cada um destes "tampões", com uma capacidade suficiente para registar a informação corresponde a uma linha video, funciona alternadamente em escrita e leitura, ao mesmo tempo que o outro funciona em leitura e escrita, respectivamente. Obviamente, o "tampão" em escrita estará associado, num dado instante, ao processo de preparação à visualização, enquanto que o segundo, obrigatoriamente em leitura, estará associado ao processo de visualização.

Na versão actual consideram-se apenas 4 bits por cor (12 bits no total) ao nível deste módulo; à saída do módulo cálculo de iluminação dispõe-se de 5 bits por cor. A cada "tampão" corresponde portanto uma capacidade 512 × 12, implementada à base de memórias RAM TTL, com uma organização 256 × 4 e um ciclo da ordem de 40 ns.

O sinal FIMLINHA, que inverte as funções dos dois "tampões", é submetido a um atraso de sincronização igual ao número de andares da cadeia "pipe-line" anteriormente descrita.

A visualização do cursor [cf. IV. 10] é também da responsabilidade deste módulo que, ao receber a ordem CURSOR = 1, força à entrada dos conversores D/A a cor associada a este dispositivo de comunicação.

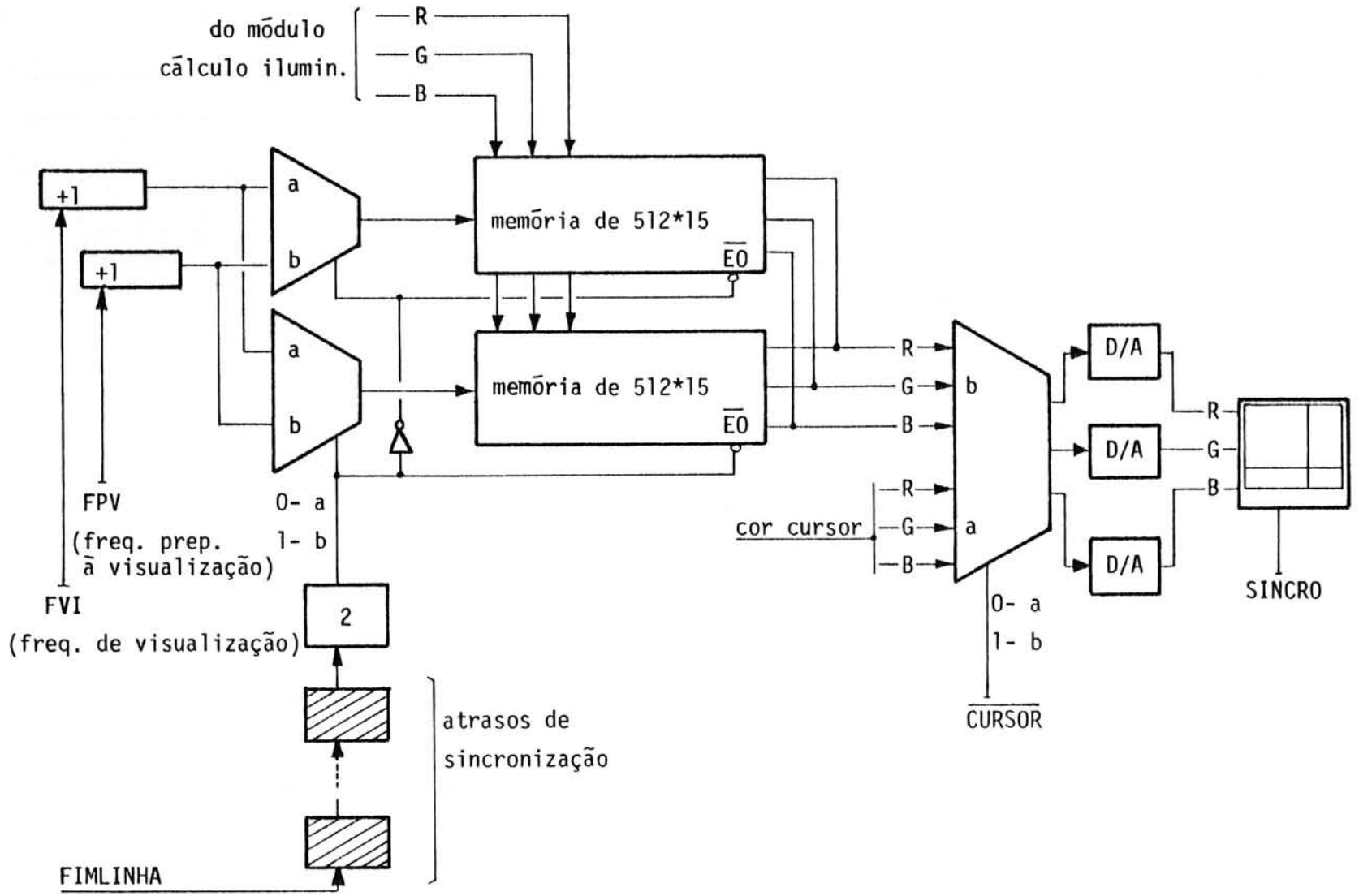


Fig. 4.32 – O princípio de realização do módulo de visualização.

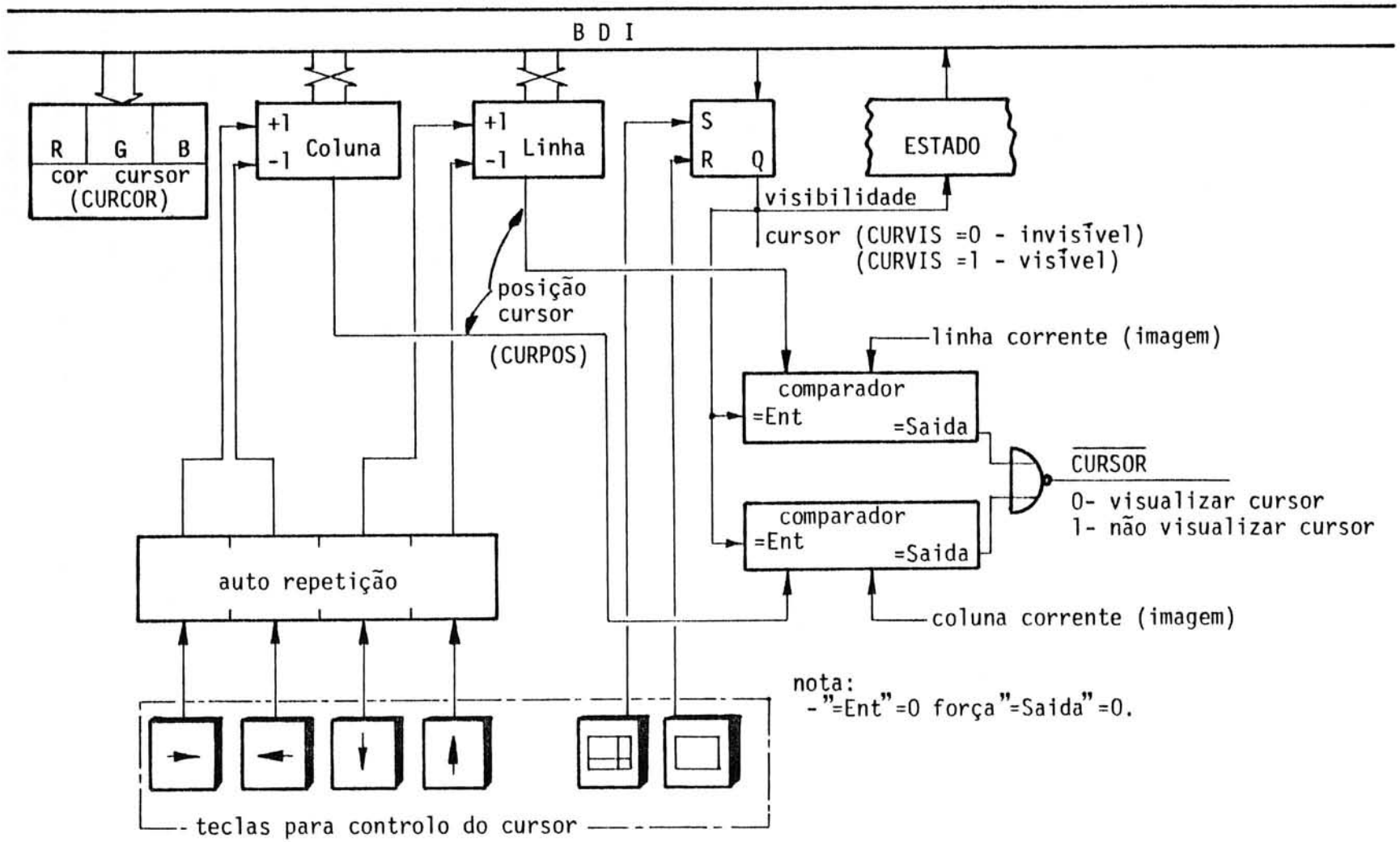


Fig. 4.33 – O princípio de realização do módulo cursor.

IV. 10. O CURSOR

As funções associadas a este dispositivo de comunicação e que podem ser activadas por "software" [cf. III. 4.5.6] são as seguintes:

— a escolha de uma cor entre 2××12 possíveis,

— a modificação ou a leitura do seu estado de visibilidade,

— a modificação ou a leitura da sua posição no espaço 512 × 512 da imagem.

Em particular, a modificação por "software" da posição do cursor permite, por exemplo, o seu controlo através de uma mesa de digitalização.

É ainda o próprio terminal que controla directamente um conjunto de teclas [cf. III. 4.5.6] com as quais é possível:

— deslocar o cursor no espaço imagem,

— modificar o seu estado de visibilidade.

A figura 4.33 ilustra o princípio de realização das funções citadas.

IV. 11. O MÓDULO DE CONTROLO

A partir de uma frequência fundamental controlada por cristal de quartzo, este módulo, verdadeiro "coração" do sistema, produz os sinais de controlo necessários ao funcionamento dos restantes módulos [fig. 4.34].

O princípio de realização deste módulo põe em evidência [fig. 4.35]:

— o sinal FPV (Frequência de Preparação à Visualização) que marca a cadência do processo de preparação à visualização e, consequentemente, de toda a estrutura "pipe-line".

— os sinais que têm um interesse especial para o varrimento da memória de quadro (FIMLINHA, por exemplo),

— o sinal FVI (Frequência de VIsualização) que, no módulo de visualização, controla a

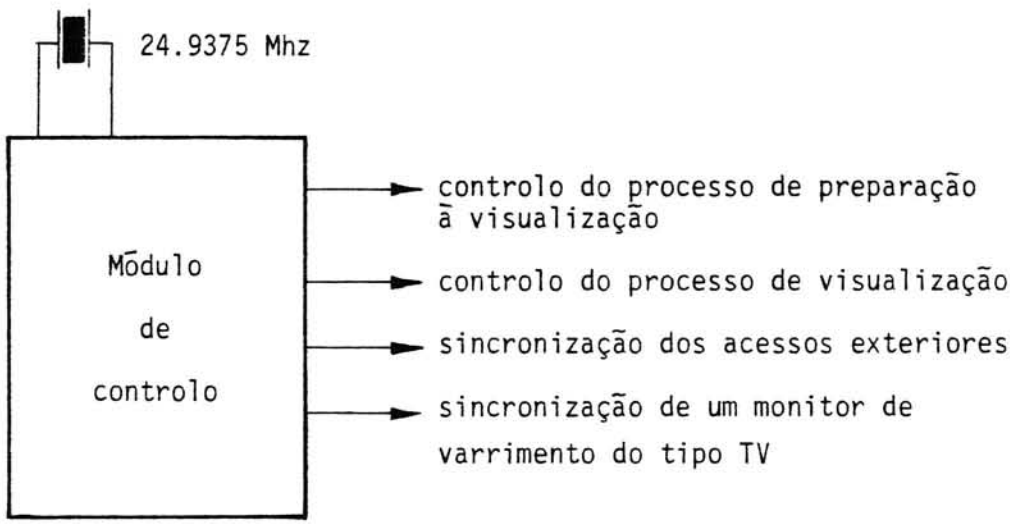


Fig. 4.34 – Os sinais gerados pelo módulo de controlo.

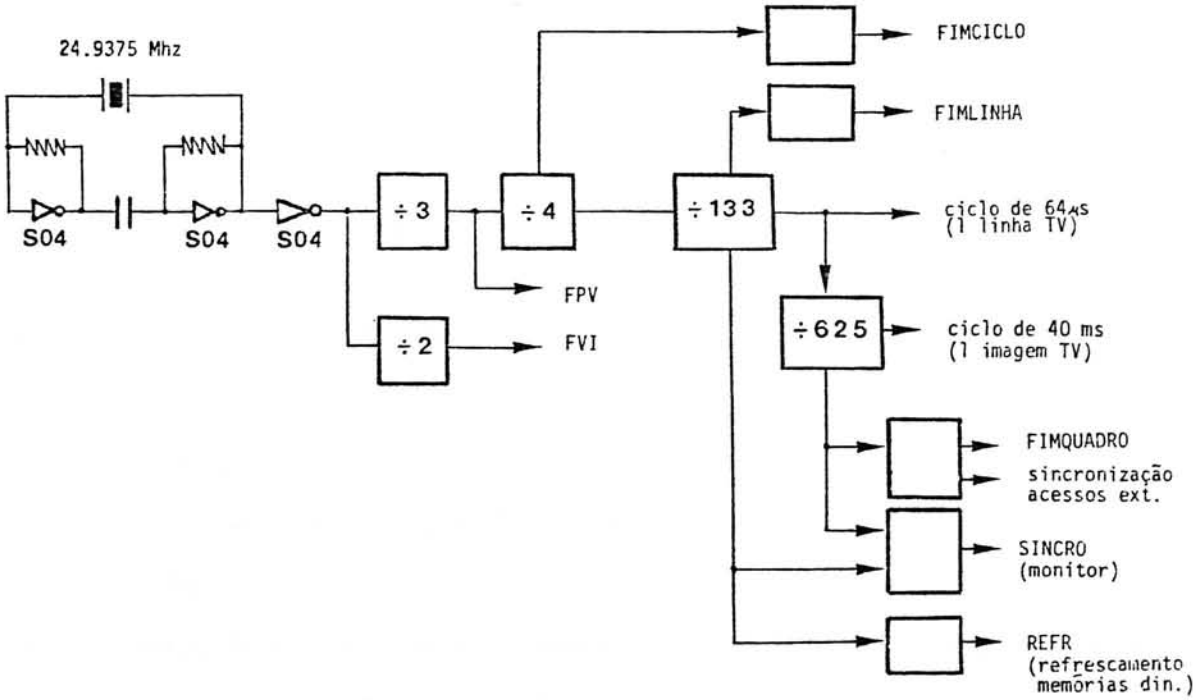


Fig. 4.35 – O princípio de realização do módulo de controlo.

cadência do processo respectivo,

— os sinais utilizados na sincronização dos acessos vindos do exterior, ao nível do módulo de comunicação,

— o sinal SINCRO exigido pelos monitores de varrimento TV,

— o sinal REFR (REFRescamento) que indica o período reservado ao refrescamento dos memórias dinâmicas,

Na versão actual, o módulo de controlo utiliza um cristal de quartzo de 24,5760 Mhz (o valor próprio deveria ser 24,9375 Mhz), facto que limita a 131 o número de ciclos em que uma linha video é decomposta; haviamos previstos 133. Desta maneira, foram eliminados dois ciclos do processo de preparação à visualização (o que corresponde a 8 pontos de uma linha com a resolução 512), por forma a conservar o número ciclos de refrescamento das memórias dinâmicas. Isto reflete-se, nas imagens de resolução 512 × 512, por uma coluna de 8 pontos sem significado próprio.

IV. 12 – **CONCLUSÃO**

No decurso do presente capítulo apresentámos os princípios de realização dos módulos que entram na composição de HELIOS. Foram postas em evidência as soluções mais originais, nomeadamente:

— *Soluções gerais*

- utilização de uma estrutura baseada nos conceitos de "pipe-line" e paralelismo [fig. 4.36],
- utilização de dois processos paralelos e assincronos, um de preparação à visualização e outro referente à visualização propriamente dita, o que permitiu duplicar o tempo de base disponível para os cálculos.

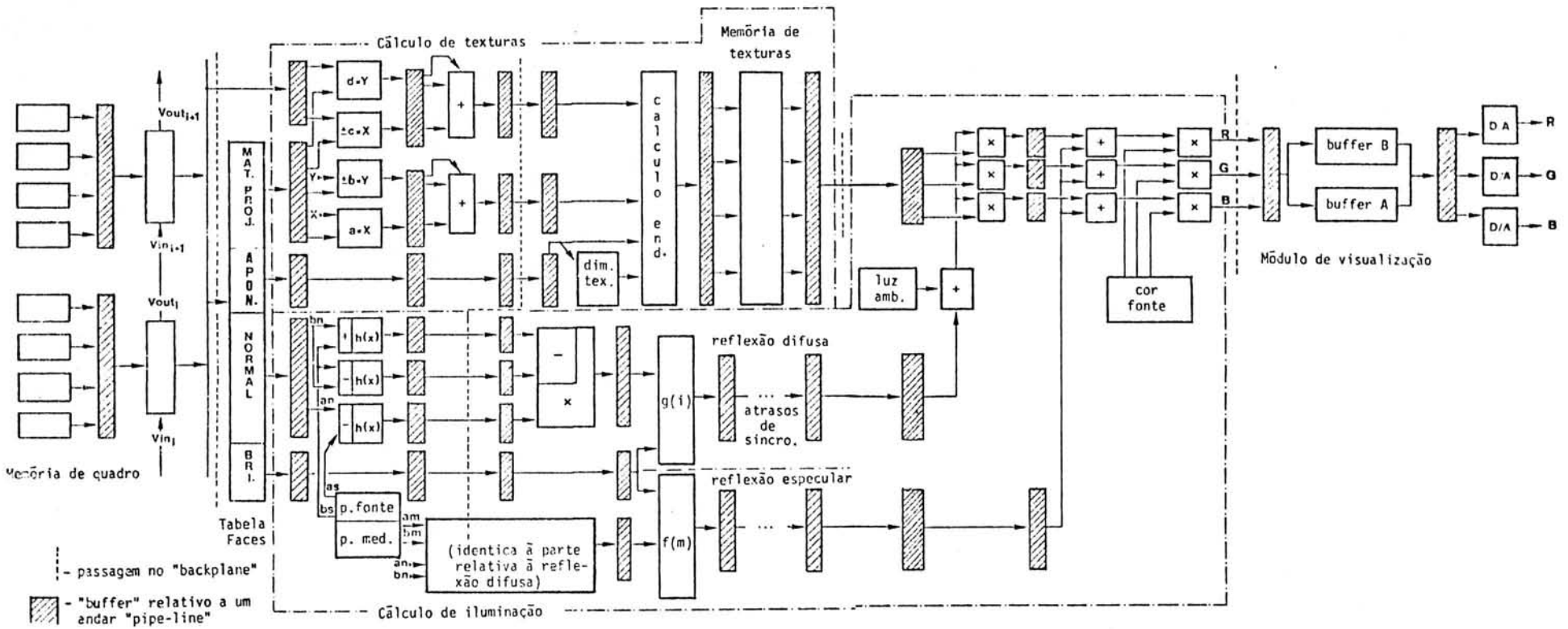


Fig. 4.36 – Panorâmica da arquitectura original do terminal HELIOS.

— Soluções específicas ao nível de cada módulo.

· *módulo de comunicação*

— tratamento da informação a um ritmo compatível com a visualização de sequências animadas em tempo real.

— possibilidade de comprimir consideravelmente a informação circulante (factor de compressão máxima igual a 255).

· *memória de quadro*

— acessos exteriores todos os 480 ns na versão que utiliza as memórias dinâmicas de 64K.

— escrita em paralelo de até quatro pontos em 480 ns.

— identificação das entidades visualizadas (faces).

· *cálculo de texturas*

— memorização e visualização de texturas coloridas de dimensões diferentes.

— projecção de texturas memorizadas de acordo com o referencial 3D das faces visualizadas.

· *Cálculo de iluminação*

— implementação "hardware" de um modelo que considera uma fonte luminosa colorida, que pode ser deslocada no espaço 3D, e uma luz ambiente difusa.

· *módulo de visualização*

— realização de um duplo "tampão" da linha video que permite separar os processos de preparação à visualização e de visualização propriamente dita.

CAPÍTULO V

CONCLUSÃO

CAPÍTULO V: CONCLUSÃO

HELIOS, um protótipo de um sistema gráfico para a visualização interactiva de imagens realistas, encontra-se completamente realizado. Foi assim possível demonstrar a validade e a viabilidade de realização da ideia de base que se traduz pela *separação dos tratamentos dos diferentes tipos de informação contidas numa cena;* a vantagem mais evidente relaciona-se com a possibilidade de se modificar localmente um tipo qualquer de informação independentemente dos outros.

Conservou-se esta separação até ao terminal e equipou-se o processador gráfico, na medida do possível, com certas funções "hardware"; isto veio permitir o cálculo em tempo real das modificações que incidirem sobre os tipos de informação considerados directamente por aquelas funções.

A arquitectura de HELIOS, inspirada nestas ideias originais, apresenta várias vantagens das quais se salientam:

— *interacção em tempo real ao nível das informações de aspecto e de iluminação.*

— *identificação das entidades visualizadas pela designação directa sobre o ecran, através de um cursor.*

Agora, a questão que se põe é a seguinte:

— Como realizar o tratamento das informações que não são directamente consideradas por HELIOS?

Para esta questão duas soluções podem ser apresentadas:

— uma relaciona-se com os desenvolvimentos de natureza "software".

— outra engloba os futuros desenvolvimentos "hardware".

Quanto aos primeiros, os testes necessários à realização do protótipo permitiram verificar que se torna difícil explorar correctamente um terminal evoluido na ausência de um "software" específico.

Pensamos portanto que novos temas de investigação podem (e devem) ser considerados, em particular ao nível do "software" de visualização, onde se trata de resolver os problemas com origem nas impossibilidades manifestadas por um terminal evoluído, mas que não é ainda o ideal, aproveitando da melhor maneira as facilidades que ele oferece.

No caso particular de HELIOS pode-se citar: modelização da transparência utilizando a facilidade das texturas, ou ainda a simulação de sombras "modulando" o vector normal das faces de maneira adequada.

Quanto aos futuros desenvolvimentos de natureza "hardware", pensamos que devem visar dois objectivos:

— interação em tempo real ao nível das informações morfológicas e geométricas, pela realização "hardware" ou microprogramada do "preenchimento das faces", ou até mesmo de um algoritmo de visibilidade (por exemplo, algoritmo de Newell, Newell e Sancha [cf. I. 4.3.3])

— aumento do nível de realismo.

Ao darmos prioridade à interação em tempo real, pudémos constatar certos condicionalismos que limitam o nível de realismo das imagens como seja, por exemplo, a utilização de faces unicamente planas, excluindo-se portanto as não planas.

Um desenvolvimento viável seria, por exemplo, considerar um dos planos de identificação como sendo um "plano de normais", onde cada posição conteria o vector normal da face não plana, em cada ponto.

Este desenvolvimento, relativamente simples de realizar, implicaria a distinção de faces planas/não planas ao nível da tabela das faces; se a face for plana o processo descrito nos capítulos precedentes é realizado, caso contrário, o valor da normal a considerar deve ser procurado no plano das normais.

Esta facilidade permitiria sintetizar imagens compostas de faces planas e faces não planas.

O conjunto de questões evocadas anteriormente podem ser inseridas numa ideia muito mais geral, baseada em duas constatações:

— A primeira mostra-nos que uma cena é, em geral, composta por objectos que se deslocam constantemente, os "objectos dinâmicos", exigindo consequentemente um cálculo muito rápido, e outros que, pelo contrário, são "estáticos" e para os quais o tempo de tratamento se torna menos crítico.

Evidentemente, a classificação de objectos em "dinâmicos" e "estáticos" é relativa à imagem final e não à cena inicial; assim um objecto "imóvel" pode ser classificado como objecto "dinâmico" se o observador se desloca ...

— A segunda constatação diz-nos que os níveis de realismo exigidos pelos objectos dinâmicos e estáticos não são os mesmos. Com efeito, o observador é muito menos sensível à falta de pormenores de um objecto dinâmico do que de um outro que é estático.

Um sistema completo, baseado em HELIOS, poderia assim propor simultaneamente duas aproximações:

— uma primeira daria prioridade ao critério tempo real, consideraria apenas faces planas desde a fase de descrição até à fase de visualização, permitindo a síntese dos objectos dinâmicos.

— uma segunda daria prioridade ao realismo, tomaria em linha de conta a descrição e a síntese de faces não planas, as sombras, a transparência, etc. conseguindo desta maneira um nível de realismo importante e consequentemente uma apresentação satisfatória dos objectos estáticos.

Estas duas formas de encarar a síntese de imagens poderiam ser aplicadas conjuntamente aos mesmos objectos, possibilitando a obtenção de imagens "semi-realistas" instantaneamente, enquanto decorresse o cálculo da imagem "realista".

Estamos convencidos que HELIOS traduz um primeiro passo na direcção de uma nova forma de sintetizar e visualizar imagens realistas e que o seu interesse não reside unicamente nas possibilidades importantes que oferece, mas também nos numerosos temas de investigação e novas aplicações que inspira.

REFERÊNCIAS BIBLIOGRÁFICAS

REFERÊNCIAS BIBLIOGRÁFICAS

(AtW 78) P. ATHERTON, K. WEILER

Hidden Surface Removal Using Polygon Area Sorting

Computer Graphics, SIGGRAPH-ACM, 1978.

(Bae 79) R. BAECKER

Digital Video Display Systems and Dynamic Graphics

Computer Graphics, SIGGRAPH-ACM, 1979.

(BaS 80) P. BAUDELAIRE, M. STONE

Techniques for Interactive Raster Graphics

Computer Graphics, SIGGRAPH-ACM, 1980.

(BeB 80) A. BECHTOLSHEIM, F. BASKETT

High Performance Raster Graphics for Microcomputer Systems

Computer Graphics, SIGGRAPH-ACM, 1980.

(Bli 78) J. F. BLINN.

Simulating of Wrinkled Surfaces

Computer Graphics, SIGGRAPH-ACM, 1978.

(BoK 78) J. BOUKNIGHT, K. KELLEY

An Algorithm for Producing Half-Tone Computer Graphics

Presentation with Shadows and Movable Light Sources

Spring Joint Computer Conference, 1970.

(Bou 80) P. BOULLE

Etude et Réalisation d'Algorithmes pour la Visualisation de

Scènes Composées de Facettes Planes

Thèrese de Docteur Ingénieur, Grenoble, 1980.

(Bra 75) I. C. BRAID

The Synthesis of Solids Bounded by many Faces

CACM, abril 1975.

(CaS 80) E. CATMULL, A. R. SMITH

3-D Transformations of Images in Scan-Line Order

Computer Graphics, SIGGRAPH-ACM, 1980.

(Cat 75) E. CATMULL

Computer Display of Curved Surfaces

Proc. IEEE Conf. on Computer Graphics, Pattern Recognition

and Data Structures, Maio 1975.

(Cla 80) J. H. CLARK

A VLSI Geometry Processor for Graphics

COMPUTER, Julho 1980.

(CHP 79) C. CSURI, R. HACKATHORN, R. PARENT, W. CARLSON

Towards an Interactive High Visual Complexity Animation System

Computer Graphics, SIGGRAPH-ACM, 1979.

(Cra 77) F. C. CROW

Shadow Algorithmes for Computer Graphics

Computer Graphics, SIGGRAPH-ACM, 1977.

(EaS 75) J. EASTMAN, J. STAUDHAMMER

Computer Display of Colored Three-Dimensional Objects

Proc. 2nd Annual Symposium on Computer Architecture, Janeiro 1975.

(Eng 75) J. M. ENGLAND e outros

A Display-Optimized Processor

Proc. 2nd Annual Symposium on Computer Architecture, Janeiro 1975.

(Eve 52) R. R. EVERETT

The Whirlwind I Computer

Joint AIEEE-IRE Conf., Rev. Electron. Digital Comput., 1952.

(FeG 80) E. FEIBUSCH, D. P. GREENBERG

Texture Rendering System for Architectural Design

Computer-Aided Design, Março 1980.

(FLC 80) E. FEIBUSCH, M. LEVOY, R. L. COOK

Synthetic Texturing Using Digital Filters

Computer Graphics, SIGGRAPH-ACM, 1980

(Gou 71) H. GOURAND

Continuous Shading of Curved Surfaces

IEEE Transactions on Comp., Junho 1971.

(Gra 80) M. GRAVE

Etude d'un Noyau de Système de Synthèse d'Images. Aplication

à la Visualisation de Scènes Tridimensionnelles.

Thèse de Docteur-Ingénieur, Lille, Dezembro 1980.

(Gre 77) D. P. GREENBERG

An Interdisciplinary Laboratory for Graphics Research

and Applications

Computer Graphics, SIGGRAPH-ACM, 1977.

(GrJ 78) D. P. GREENBERG, G. H. JOBLOVE

Color Spaces for Computer Graphics

Computer Graphics, SIGGRAPH-ACM, 1978.

(Han 77) G. HANBERG

Advanced CGI VISUAL Technology Reshapes Pilot Training

Possibilities

ICAO Bulletin, abril 1977.

(Her 80) D. HERBISON-EVANS

Rapid Raster-Ellipsoid Shading

Computer Graphics, Spring 1980.

(Ler 80) P. LERAY

Réalisation d'un Système de Génération Synthetique d'Images en Temps Réel (G. S. I.)

Congrès AFCET, Nancy, Novembro 1980.

(Luc 77) M. LUCAS

Contribuition à l'Etude des Techniques de Communication Graphiques avec un ordinateur. Elements de Base des Logiciels Graphiques Interactifs.

Thèse de Docteur Es-Sciences, Grenoble, 1977.

(Luc 79) M. LUCAS

Techniques de Modelisation de Scènes Tridimensionnelles sans utilisation de Facettes Polygonales Planes

Rapport Bibliographique, IMAG, Grenoble, Fevereiro 1979.

(MaF 81) F. MARTINEZ, F. FERREIRA

HELIOS: Termical Interactif pour la Synthèse d'Images Réalistes

Congrès AFCET TTI Informatique, Gif sur Yvette, Novembro 1981.

(Mar 79) F. MARTINEZ

La Synthèse d'Image

Convention LMT-MICADO, Grenoble, Janeiro 1979.

(Mar 79.1)F. MARTINEZ

An Approach to the Modelling and Display of Landscapes

EUROGRAPHICS 79, Bologna, 1979.

(Mar 80) F. MARTINEZ

CLOVIS: Complexe Logiciel pour la Visualisation Interactive Structurée

Congrès AFCET TTI Informatique, Nancy, Novembro 1980.

(Mar 81) F. Martinez

Rapport Scientifique IMAG

Fevereiro 1981.

(Mat 78) P. MATHERAT

A Chip for Low-Cost Raster-Scan Graphic Display

Computer graphics, SIGGRAPH-ACM, 1978.

(Mer 79) M. MERIAUX

Etude et Réalisation d'un Terminal Graphique Couleur

Tridimensionnel Fonctionnant par Tâches

Thèse de Docteur-Ingénieur, Lille, Janeiro 1979.

(Mye 75) A. J. MYERS

An Efficient Algorithme for Computer Generated Pictures

Ohio State Univ., Computer Graphics Res. Group, 1975

(NeS 73) W. M. NEWMAN, R. F. SPROULL

Principles of Interactive Computer Graphics, 1st edition

McGraw-Hill, New York, 1973.

(NeS 79) W. M. NEWMAN, R. P. SPROULL

Principles of Interactive Computer Graphics, 2nd edition

McGraw-Hill, New York, 1979.

(NNS 72) M. E. NEWELL, R. G. NEWELL, T. L. SANCHA

A New Approach to the Shaded Picture Problem

Proc. ACM National Conference, 1972.

(OgS 75) D. J. OGDEN, J. STAUDHAMMER

Computer Graphics for 3D Object Images

Computer and Graphics, Junho 1975.

(Par 80) F. I. PARKE

Simulation and Expected Performance Analysis of Multiple Processor Z-Buffer Systems

Computer Graphics, SIGGRAPH-ACM, 1980.

(Pho 75) B. T. PHONG

Illumination for Computer Generated Pictures

CACM, Junho 1975.

(PiW 80) E. PILLER, H. WIDNER

Real-Time Scan Unit with Improved Picture Quality

Computer Graphics, Junho 1980.

(Sho 79) R. C. SHOUP

Color Table Animation

Computer Graphics, SIGGRAPH-ACM, 1979.

(Smi 78) A. R. SMITH

Color Gamut Transform Pairs

Computer Graphics, SIGGRAPH-ACM, 1978.

(SSS 74) I. E. SUTHERLAND, R. F. SPROULL, R. A. SHUMACKER

A Characterization of ten Hidden-Surface Algorithmes

ACM Computing Surveys, Março 1974.

(Sta 78) J. STAUDHAMMER

On Display of Space Filling Atomic Models in Real-Time

Computer Graphics, SIGGRAPH-ACM, 1978.

(StT 79) J. STAUDHAMMER, J. TARTAR

Shaded Graphics Hardware

Eurographics, 1979.

(The 72) F. THERON

Sur le programme EUCLID: création, manipulation et visualisation de formes tridimensionnelles dans un langage géométrique à génération dynamique

Thèse de 3ème cycle, Paris, 1972.

(Wat 70) G. S. WATKINS

A Real Time Visible Surface Algorithme

University of Utah, UTEC, CSs 70101, Junho 1970.

(Wei 78) R. WEINBERG

Computer Graphics in Support of Space Shuttle Simulation

Computer Graphics, SIGGRAPH-ACM, 1978.

(Wil 78) L. WILLIAMS

Casting Curved Shadows on Curved Surfaces

Computer Graphics, SIGGRAPH-ACM, 1978.

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000006212