

**PGC**  
**UM PROCESSADOR GRÁFICO CONCORRENTE**

**João Madeiras Pereira (IST/INESC)**

**Mário Rui Gomes (IST/INESC)**

**INESC - Projecto CAD/CAM**

**Rua Alves Redol 9 2º**

**1100 Lisboa**

**Portugal**

**(01)-528163**

**jap@inesc.inesc.pt**

**mcsunlinesc|jap**

**mrg@inesc.inesc.pt**

**mcsunlinesc|mrg**

**RESUMO** - O Sistema PGC é uma proposta de uma arquitectura paralela tanto ao nível dos algoritmos como dos processadores baseado numa rede de transputers.

O Sistema proposto implementará um sub-conjunto da funcionalidade da futura extensão á norma PHIGS, o designado PHIGS+ e tem como objectivo a visualização, em tempo real, de imagens tridimensionais com elevado grau de realismo.

## **1. INTRODUÇÃO**

Em Computação Gráfica a síntese de imagens realistas tornou-se um requisito imprescindível em múltiplas áreas de aplicação que manipulam informação de elevada complexidade tais como arquitectura, mecânica, química, medicina, biologia, CAD/CAM, ensino, entre outras.

A visualização de estrutura em tensão pelo realce com tons de vermelho das superfícies em pré-rotura ou a visualização, no tempo, de ondas propagando-se no espaço sofrendo ou não atenuação, constituem técnicas que utilizam a síntese de imagens realistas, conferindo ao processo de análise um maior grau de compreensão.

Nos últimos anos assistiu-se a duas direcções de evolução no campo de síntese e geração de

imagem:

- aumento do realismo na imagem, sem limitações no seu tempo de cálculo
- construção de sistemas altamente interactivos de modo a produzir imagens com menor realismo mas num período de tempo fixo (entre 1/30 e 1/60 segundos).

O objectivo ideal é a síntese de imagens realistas em tempo real. No entanto, a introdução de realismo em cenas complexas implica a execução de um conjunto de tarefas, cada uma delas requerendo uma grande quantidade de cálculos. A lista dessas tarefas inclui, entre outras: percorrer a base de dados da cena ("display-list"), efectuar transformações geométricas (rotação, translação, escala e perspectiva), aplicar o modelo de iluminação, rasterizar, calcular as superfícies ocultas, sombrear, incluir texturas e gerar sombras. A complexidade computacional envolvida na geração de imagens de objectos tridimensionais com um elevado grau de realismo em tempo "quase real" passa, inevitavelmente, pelo recurso a arquitecturas paralelas. Três questões essenciais se colocam ao assumirmos tal solução para o problema da Geração de Imagem por Computador (GIC):

- Como paralelizar os algoritmos gráficos?
- Como balancear correctamente os vários processos pelos nós da arquitectura paralela?
- Como configurar a arquitectura multiprocessador de modo a maximizar o desempenho do processo GIC?

Neste artigo propõe-se um ambiente HW/SW que permite a implementação eficiente de algoritmos gráficos sobre uma arquitectura paralela de alto desempenho baseada em componentes disponíveis comercialmente.

## 2. IMPLEMENTAÇÃO DO PROCESSO DE VISUALIZAÇÃO

A função básica de um sistema de visualização é a transformação de informação geométrica tridimensional numa imagem bidimensional desenhada num ecrã de um monitor, função que se designa por "rendering". A informação tridimensional constitui o modelo matemático que descreve a cena a ser visualizada. Esse modelo será baseado na norma PHIGS+, e, portanto, estruturado de um modo hierárquico. No seu estabelecimento, assumir-se-ão as seguintes hipóteses:

- 1) a informação gráfica é hierárquicamente organizada em "estruturas". A proposta ISO-PHIGS [ISO88] define estrutura como um conjunto ordenado de primitivas gráficas, matrizes de transformação, selecção de atributos e referências a outras estruturas, formando deste modo uma árvore em que as estruturas "filhas" herdam, por defeito, os atributos da estrutura "pai";
- 2) as primitivas são os elementos geométricos básicos que constituem a cena. Propõe-se dois níveis de primitivas:

- Primitivas gráficas elementares
- Primitivas de alto nível

3) a primitiva gráfica elementar é o polígono, de acordo com as recentes normas ISO [ISO88] [ISO89]. Algoritmos que lidam com sólidos primitivos, usam o triângulo como primitiva elementar [YAM85]. Algoritmos de sombreado requerem a subdivisão de polígonos em quadriláteros e triângulos. Para contemplar ambos os casos, decidiu-se utilizar como primitiva gráfica elementar, o triângulo;

4) as primitivas de alto nível a suportar serão os "poliedros", seguindo, novamente, a terminologia usada no PHIGS+ [VAN88]. Estes poliedros tanto podem corresponder à descrição de fronteira de sólidos primitivos paramétricos (cones, esferas, elipsóides, etc.), como podem corresponder à descrição de fronteira de sólidos resultantes de operações booleanas (diferença, união, intersecção) entre sólidos primitivos;

5) os poliedros serão decompostos num conjunto de primitivas elementares em tempo de "atravessamento" das estruturas.

Como foi referido, o processamento paralelo pode ser aplicado, com sucesso, na resolução do problema GIC (Geração de Imagem por Computador). Isto porque, o paralelismo obtém-se através da decomposição do processo de visualização numa sequência de tarefas que podem ser executadas em paralelo [STR86].

Uma dessas tarefas será implementada pelo denominado "Processo Atravessador" [PAO89] que terá por tarefa fundamental percorrer ("atravessar") a árvore de estruturas, devolvendo um conjunto linear de objectos (primitivas elementares) em coordenadas do Mundo Real. A descrição de cada triângulo contém as coordenadas e valores de intensidade dos respectivos vértices e o vector normal à superfície. Durante o "atravessamento" mantém-se uma "lista de estado" que contém as matrizes de transformação e os valores dos atributos gráficos correntes.

A tarefa principal do PA (Processo Atravessador) consistirá na análise da topologia da estrutura hierárquica a ser visualizada e na construção de uma sua representação interna a que designaremos, simplesmente, por "modelo" - figura 1. Os nós do modelo representam "subestruturas" e/ou o início, ou o fim, de uma estrutura (instruções "OPEN STRUCTURE" e "CLOSE STRUCTURE"). Uma subestrutura é um subconjunto de primitivas gráficas adjacentes que são descritas pela mesma informação de estado. Os arcos orientados não etiquetados representam invocações de outras estruturas (instrução "EXECUTE STRUCTURE"). Os que são etiquetados têm associado os valores de estado - matrizes de transformação e selecção de atributos - presentes na subestrutura.

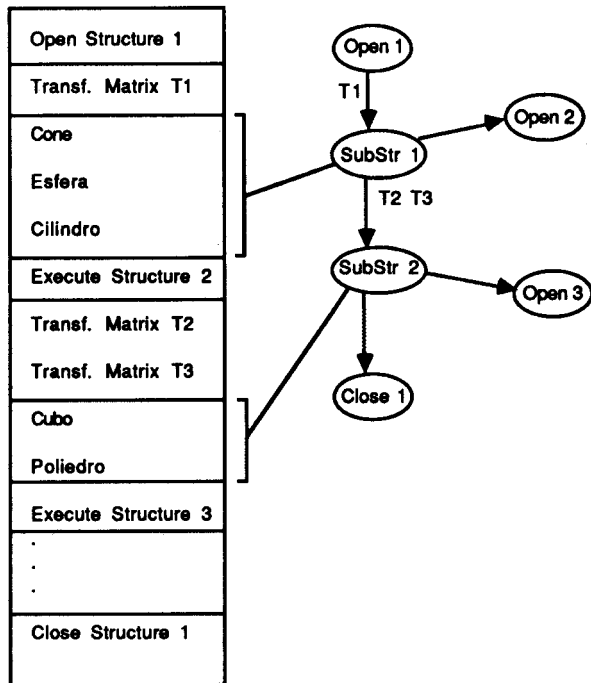


Figura 1 - Estrutura hierárquica tipo PHIGS e modelo associado

Na fase de inicialização do sistema, o processador central ("host") carregará a estrutura hierárquica numa zona de memória da placa, a que se designa por "display list". O PA divide essa estrutura hierárquica em subestruturas, ou seja constrói o "modelo" e percorre essa representação interna calculando o estado corrente relativo a cada subestrutura. Essa informação (subestrutura + estado corrente) será enviada para o Processo Transformador, PT, o qual responsabilizar-se-á pelas operações geométricas, ou seja, pela transformação das coordenadas das primitivas definidas no Mundo Real em coordenadas do Espaço Imagem (coordenadas de visualização) de acordo com uma posição e uma direcção de visualização. Esta transformação consiste na execução da matriz de transformação (em coordenadas homogéneas), recorte ("clipping") e transformação perspectiva para obter a ilusão de profundidade. Para facilitar o recorte é aplicada, em primeiro lugar, a transformação perspectiva seguida da transformação de normalização. O recorte é efectuado sobre o volume normalizado, tornando, normalmente, o processo mais eficiente. O aspecto das primitivas é, igualmente, alterado de acordo com os atributos correntes.

A informação gerada pelo PT será transferida para o "Processo Desenho" que implementa a secção do "pipeline de visualização" dedicada ao processamento sobre o pixel.

Em vez de um único PT, optou-se por utilizar Processos Transformadores adicionais de modo a aumentar a velocidade de desenho dos objectos. Na realidade, N Processos Transformadores

poderão processar informação a um ritmo até N vezes superior ao de um único PT. Este factor de rapidez depende não só do ritmo com que o Processo Atravessador será capaz de fornecer informação, mas também, da velocidade com que o Processo Desenho, PD, calcula os pixels a desenhar no ecrã. Esta velocidade de desenho poderá ser aumentada se considerarmos a introdução de paralelismo no PD. Existem duas estratégias alternativas para obtenção de tal objectivo: subdivisão (ou partição) do Espaço Imagem ou subdivisão do Espaço Objecto.

No projecto, implementar-se-á partição no Espaço Imagem. A opção alternativa, partição no Espaço Objecto, impõe dependência entre os processadores da rede, o que, a nosso ver, deve ser evitado.

A decisão tomada implicará a existência vários processos, cada um dos quais responsável por uma região do ecrã. Denominar-se-ão cada um destes processos por Processo Imagem, PI. Este tipo de partição é, como se sabe, adequado à implementação do algoritmo Z-buffer [FOL84] como solução do problema da eliminação das superfícies ocultas. Este algoritmo elimina as áreas ocultas através da comparação de profundidade pixel a pixel. De realçar que o algoritmo Z-buffer trabalha separadamente sobre cada um dos pixels e não é dependente de nenhuma relação geométrica ou temporal entre eles, podendo ser, portanto, facilmente paralelizado. Utilizar-se-á um Z-buffer com uma profundidade de 16 bits.

### 3. IMPLEMENTAÇÃO DO PROCESSADOR GRÁFICO CONCORRENTE

Para a realização de uma placa gráfica com arquitectura paralela que suporte a funcionalidade descrita e com uma resolução de 1024 x 1024 pontos com 24 planos de cor (cor real), optou-se pelos Transputers IMS-T800 que implementam o Processador Gráfico Concorrente, PGC. A concorrência do Processador Gráfico ocorre a dois níveis: do software (existência de processos concorrentes em cada transputer) e do hardware (vários transputers processando concorrentemente).

O T800 é um microcomputador CMOS que integra num único "chip" um CPU de 10 MIPS (versão 20 MHz), um processador de vírgula flutuante 32/64 bits de 1.5 MFLOPS conforme a norma IEEE754, 4 Kbytes de RAM interna estática, interface de memória com controlador de DRAM capaz de endereçar 4 Gbytes de memória externa a um ritmo máximo de 26.6 Mbytes/s e quatro "links" de comunicação interprocessador "full-duplex".

O processador suporta instruções do tipo "block move" microprogramadas, que proporcionam um suporte eficaz para a implementação de operações gráficas como "windowing", "zooming", manipulação de texto e actualização de ecrã.

A cada um dos "links" está associado dois dispositivos DMA internos (um para entrada e outro para saída) que proporcionam ritmos de comunicação até 20 Mbits/s, (2.35 Mbytes/s), e funcionam independentemente dos outros "links" ou do processador. Por isso, os quatro "links" podem operar simultaneamente, proporcionando transferência de informação máxima de 9.4 Mbytes/s.

A existência destes "links" juntamente com um sequenciador implementado por hardware extremamente rápido (a comutação de processos é da ordem de 1  $\mu$ s) faz com que os Transputers apresentem a capacidade de serem facilmente configurados em redes extremamente eficientes.

Numa primeira versão, o Processador Gráfico constará de uma rede em anel de quatro Transputers. Mas, devido à possibilidade de esta arquitectura poder vir a exibir uma elevada frequência de acesso à memória de imagem através do bus de sistema, cada processador da rede disporá de memória própria local, com o objectivo de evitar a saturação do bus - figura 2.

Na primeira versão do sistema, o ecrã será dividido em dezasseis regiões. Portanto, em cada Transputer serão implementados quatro PIs, cada um deles com os seus Z-buffer (128 Kbytes) e "frame-buffer" (192 Kbytes) locais, ocupando um total 1280 Kbytes de memória local. Os restantes 768 Kbytes (ver figura 1) serão dedicados, fundamentalmente, às transformações geométricas e aos algoritmos de cálculo (rasterização, eliminação das superfícies ocultas e sombreamento).

Em termos de arquitectura funcional, e tirando partido da estrutura "pipeline" do processo de visualização, implementar-se-ão quatro Processos Transformadores cujas saídas serão enviadas para um "pipeline" de dezasseis Processos Imagem - figura 3.

Com o objectivo de melhorar o desempenho da arquitectura acima descrita, introduzir-se-á paralelismo intra-andar.

Assim, cada PT pode ser distribuído sobre uma estrutura pipeline, como facilmente se reconhece da figura 4.

Inicialmente, optar-se-á por um algoritmo simples de sombreamento, o de Gouraud. Este método de sombreamento será incorporado no algoritmo de rasterização, dado que envolve, meramente, a interpolação linear entre intensidades definidas nos vértices dos triângulos. Detalhando as operações a serem executadas por um Processo Imagem, chegar-se-á à conclusão que, em primeiro lugar haverá que calcular as posições das arestas do polígono em relação a cada pixel com o objectivo de determinar se este último se localiza no interior ou no exterior do primeiro. Apenas os pixels que se encontram no interior do triângulo serão utilizados na posterior comparação de profundidade. Como já foi mencionado, cada PI terá o seu próprio Z-buffer, pelo que após o cálculo da profundidade do triângulo nesse pixel, esse valor só será armazenado se for menor que o valor anterior. Nessa situação, o PI calculará a intensidade desse pixel através da interpolação entre valores armazenados nos vértices do polígono, guardando-o, de seguida, na respectiva posição do "frame-buffer" local.

Novamente, a questão do paralelismo intra-andar pode ser utilizada. De facto, cada PI pode ser dividido em três tarefas e distribuídas segundo um "pipeline" - figura 5.

Após um triângulo ter sido processado por um PI, será novamente introduzido no próximo PI, que poderá correr no mesmo Transputer ou num outro, e assim sucessivamente. Quando o último polígono tiver sido processado pelo último PI do "pipeline" de imagem, gerar-se-á um Processo Display que moverá os dezasseis "frame-buffers" locais para o "frame-buffer" principal. Esta

transferência envolverá o acesso dos quatro Transputers ao bus do sistema, pelo que se tornará necessária a arbitragem deste último. Esta questão será solucionada através da utilização clássica de um "token" que viajará ao longo do anel. O Processo Display, apenas, iniciará a sua

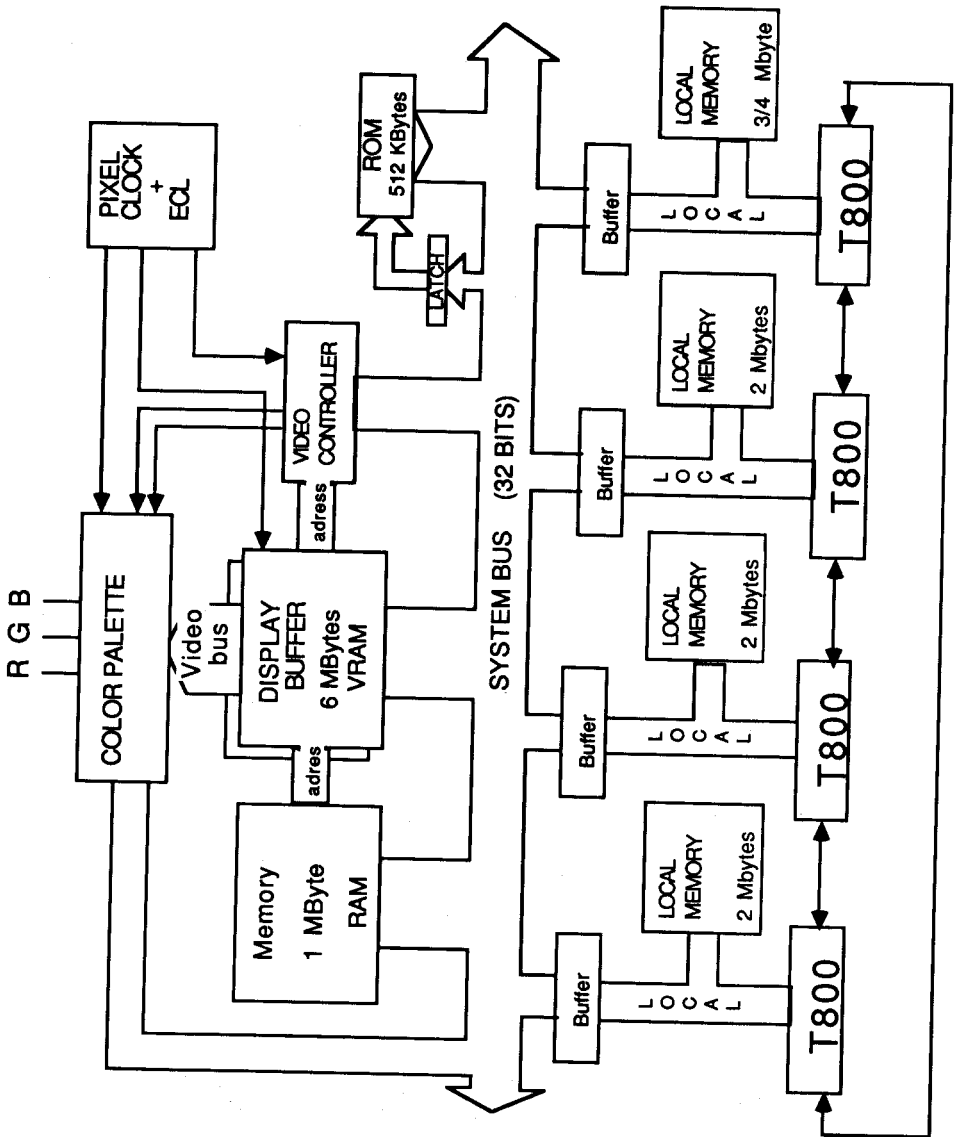


Figura 2 - Diagrama de blocos da primeira versão do sistema gráfico

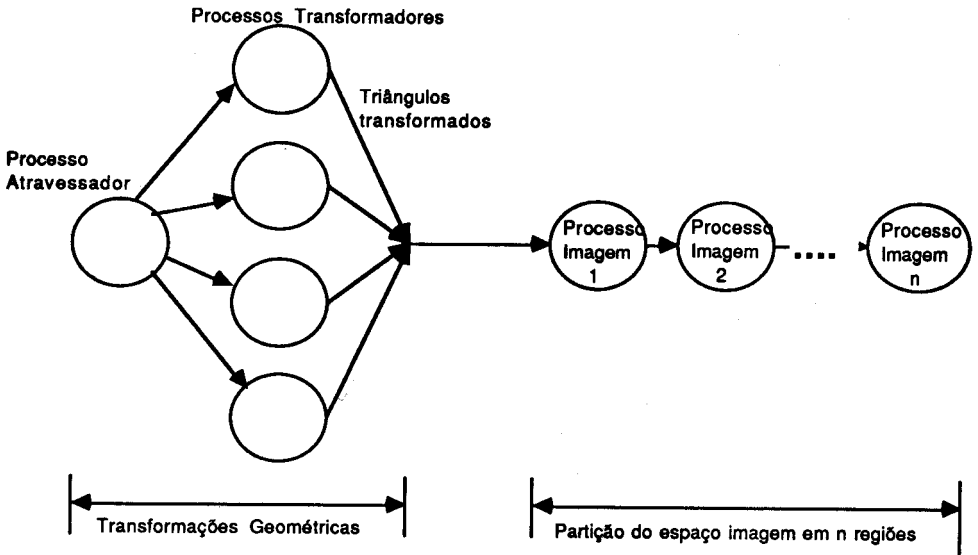


Figura 3 - Arquitectura funcional do processo de visualização

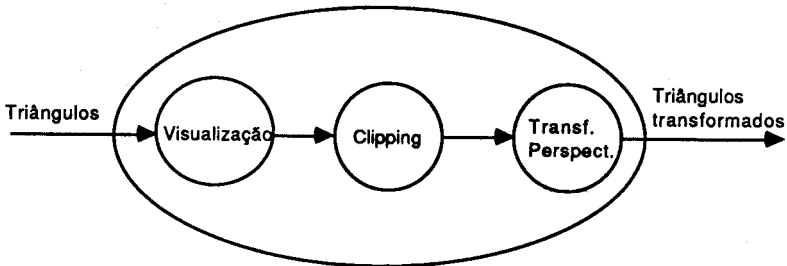


Figura 4 - Processo Transformador

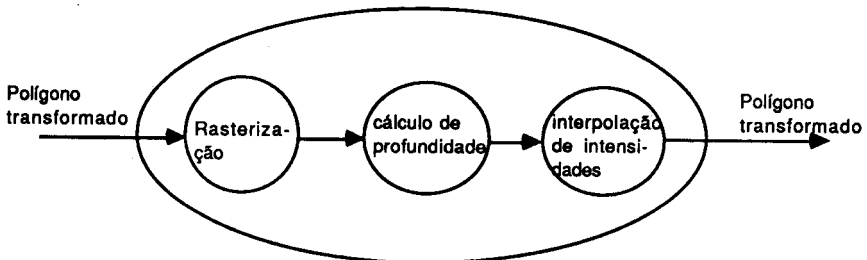


Figura 5 - Processo Imagem



acção relativamente a um Transputer quando este possui o "token". Os outros não poderão, entretanto, aceder ao bus do sistema. Após a transferência, o Processo Display passará o "token" para o próximo processador dando início à correspondente actividade.

Um ponto importantíssimo a realçar nesta implementação do processo de visualização será o uso intensivo de "buffering" inter-andares que decorre da concorrência comunicação/processamento proporcionada pelo Transputer. Por exemplo, entre as saídas dos quatro Processos Transformadores e a entrada do "pipeline" dos dezasseis PIs existirá um Processo Multiplexer. Este processo aceitará mensagens que estejam disponíveis de qualquer um dos PTs e as encaminhará para um buffer. Está aqui implícito a utilização de um mecanismo tipo "semáforo". Quando um dos PTs está a transferir informação para o buffer, os outros três PTs, eventualmente, terão que aguardar pela sua vez, através da sinalização do semáforo.

#### 4. MEMÓRIA DE IMAGEM E ANDAR DE VIDEO

O projecto de sistemas com desempenho gráfico em tempo real é caro e difícil pois implica a geração de imagens no ecrã em tempo real, ou seja, o sistema deve possuir a capacidade de recalculer e redesenhar a imagem 30 a 60 vezes por segundo de modo a dar a ilusão de movimento contínuo. Uma análise do sistema gráfico revela que o maior obstáculo ao seu desempenho é, não só a saturação do bus do sistema cuja resolução já foi descrita, mas também, o método de organização utilizado na implementação da memória de imagem ("frame-buffer") [WHI84]. Na realidade, o Processador Gráfico Concorrente, PGC, altera a imagem visualizada através da escrita de novos dados na memória de imagem, o que significa que, em sistemas "real-time", são escritos milhões de pixels em cada trama exigindo que o processador tenha ao seu dispor um grande número de ciclos de memória. Por outro lado, para que a imagem seja estável, livre de "flicker", o controlador de vídeo ou VRC ("video refresh controller") deve enviar dados para o monitor de acordo com parâmetros de temporização muito precisos, implicando, conseqüentemente, que o controlador tenha um acesso adequado à memória de imagem. Como resultado, o PGC e o VRC encontram-se em contenção por um número finito dos ciclos de memória disponíveis.

Portanto, no projecto do "frame-buffer" três questões devem ser contempladas: a contenção por ciclos de memória, a geração de uma imagem estável no ecrã (largura de banda exigida pelo VRC) e a modificação da imagem visualizada (largura de banda requisitada pelo PGC). Em caso de contenção, é o último parâmetro que é afectado não sendo alcançado o propósito de visualização em tempo real.

A contenção da memória de imagem é totalmente eliminada com o uso da técnica de "double-buffering" e quase eliminada com o uso das vídeo RAMs (VRAMs).

No projecto usar-se-á "double-buffering" com VRAMs.

O "double-buffering" elimina a visualização de uma imagem ainda com porções da imagem anterior uma vez que existem dois "frame-buffers" distintos: um para a imagem a ser

visualizada no ecrã e outro para a formação da nova imagem. A opção por VRAMs foi, unicamente, imposta pela necessidade de garantir a largura de banda exigida pelo controlador de vídeo. De facto, os registos internos das VRAMs proporcionam um meio eficiente de obter o necessário ritmo dos pixels. Uma outra solução seria a da utilização de memórias mais rápidas que conduziria a custos mais elevados.

O "frame-buffer" utilizará as VRAMs de 1 Megabit (256K x 4) TMS44C251 da Texas Instruments. Dada a resolução pretendida, 1024 x 1024 x 24, serão necessários 24 circuitos integrados para implementar um buffer (no total 48 integrados serão necessários para os dois buffers).

É essencial conhecer o ritmo com que um novo pixel deve ser injectado no andar de saída de vídeo de modo a determinar a velocidade de acesso ao "frame-buffer" para efeitos de refrescamento do ecrã. No nosso caso, um "display" de 1024 x 1024 e uma taxa de refrescamento de 60 Hz implica um tempo de pixel da ordem dos 11 ns, ou seja uma largura de banda de 90 Mpixeis/s (24 bits por pixel). Ora, uma operação de leitura da memória de imagem por parte do VRC, tirando partido dos registos internos das VRAMs, ocorre apenas de duas em duas linhas; portanto o controlador necessitará de um ciclo de memória para aceder a 2048 pixels (a referida operação de leitura consiste numa transferência de dados da memória para os respectivos registos). É de realçar que, entre duas operações de leitura desencadeadas pelo VRC, os dados são deslocados em série nos registos a uma velocidade máxima de 45 MHz (o deslocamento é temporizado pelo controlador). Como quatro pixels são acedidos por cada operação de deslocamento, o "frame-buffer" garantirá para o VRC uma largura de banda máxima de 180 Mpixeis/s, portanto acima da largura de banda necessária.

O VRC a ser utilizado consitui um dos blocos funcionais do integrado GSP (Graphics System Processor) TMS34020 da Texas Instruments e será responsável pelo controle da imagem no ecrã, da memória de imagem e das VRAMs. De referir que, de momento, o TMS34020 é o único processador que aproveita directamente as potencialidades inovadoras proporcionadas pelas vídeo RAMs TMS44C251: escrita de blocos, modo de acesso "enhanced page-mode", entre outras.

O bloco controlador de vídeo do TMS34020, além de gerar os sinais de temporização de vídeo necessários para a realização da interface com um monitor raster, liberta o Processador Gráfico Concorrente das tarefas de endereçamento e de refrescamento da memória dinâmica, incluindo o carregamento e deslocamento dos registos internos das VRAMs.

Os registos internos das 24 VRAMs comportam-se como um grande buffer de vídeo com uma organização de 512 x 4 pixels, pelo que numa operação de deslocamento, quatro pixels serão acedidos simultaneamente, sendo, portanto, necessário, realizar uma conversão paralelo-série. Esta função será realizada por três RAMDACs BT457/170 MHz da Brooktree. Cada uma destas RAMDACs possui, essencialmente, uma tabela de cor de 256 x 8 e um conversor D/A de oito bits. Devido às elevadas frequências de relógio envolvidas na operação das RAMDACs, estas lidam com sinais de níveis eléctricos ECL. Como se sabe, a utilização de lógica ECL pressupõe um manuseamento muito cuidadoso que se traduz, fundamentalmente, no recurso a um ambiente de

linhas de transmissão com as correspondentes terminações. Na realidade, tal ambiente é necessário devido à grande propensão dos circuitos ECL para a ocorrência de reflexões e "crosstalk", situações de ruído que devem ser eliminadas.

## 5. CONCLUSÕES

A arquitectura paralela foi especificada com o objectivo de obter um desempenho "real-time". No entanto, tal propósito depende do software gráfico de suporte utilizado na implementação do "pipeline de visualização". De facto, o desempenho do Processador Gráfico Concorrente está intimamente dependente do grau de paralelismo que é possível introduzir nos algoritmos gráficos (programação concorrente) e da distribuição mais eficiente dos vários processos pelos quatro Transputers. Para efeitos de estimativa far-se-á uma avaliação do desempenho da referida arquitectura paralela HW/SW. Nesta fase, há que ter em conta a utilização ou a geração de ferramentas que permitam medir ou monitorar o desempenho de um algoritmo ao qual foi aplicado um particular esquema de paralelização:

- monitoração do tempo gasto em determinadas rotinas
- medição do tempo dedicado à comunicação e ao I/O
- medição das interacções entre processos e monitoração de acontecimentos especificados por um utilizador

Outra ferramenta importante, a implementar, permitirá concluir qual o correcto balanceamento dos vários processos pelos processadores da rede.

Em alternativa ao "pipeline" de visualização, desenvolver-se-á um outro software gráfico de suporte que implementará a futura norma ISO-CGI (Computer Graphics Interface) sobre um outro processador que se encontra na placa: o TMS34020. Este processador dispõe, para além de um conjunto de instruções gráficas poderosas, de um vasto "set" de instruções de uso geral.

## REFERÊNCIAS

- [FOL84] J.D. Foley, A. Van Dam, *Fundamentals of Computer Graphics*, Addison-Wesley Publishing Company, 1984.
- [ISO88] ISO: Information Processing Systems; "Programmer's Hierarchical Interactive Graphics System (PHIGS) - Functional Description", Part 1 ISO/IEC 9592, 1988
- [ISO89] ISO: Information Processing Systems; "Graphics Kernel System for Three Dimensions (GKS 3-D) - Functional Description", IS 8805, 1989

- [PAO89] Paoluzzi, A.; Rosina, M.; "A Parallel HW/SW Environment for Image Synthesis", Computer Graphics Forum 8, 1989
- [STR86] Strasser, W.; "VLSI-Oriented Graphics System Design", Tutorial B1, Eurographics 1986
- [VAN88] Van DAM A., "PHIGS+ Functional Description Revision 3.0", ACM Computer Graphics 22, Julho 1988
- [WHI84] Whitton, M. C.; "Memory Design for Raster Graphics Displays", IEEE CG&A, Março 1984
- [YAM85] Yamaguchi F., Tokieda T.; "Bridge edge and triangulation approach in solid modelling", Frontiers in Computer Graphics, Springer-Verlag, 1985