

**Um Modelador de Sólidos Experimental: o Sistema MS(E)**

**Vasco Branco, DI.ISEP/INESC.Norte  
F. Nunes Ferreira, DEEC.FEUP/INESC.Norte**

**Primeiro Encontro de Computação Gráfica  
LNEC, Julho de 1988  
Lisboa**



## Sumário

Apresentam-se as várias etapas do desenvolvimento, e a situação actual, de um sistema de modelação de sólidos baseado em representações B-REP.

### 1. Introdução

Ninguém duvida, hoje em dia, da importância dos sistemas CAD como catalizadores do processo de concepção e desenvolvimento de novos produtos industriais.

Por este motivo, a realidade impôs (e impõe) que a nível académico se preparem quadros suficientemente habilitados para compreenderem, utilizarem e desenvolverem este tipo de sistemas.

O ensino da Computação Gráfica, ganha assim uma dimensão relevante como base indispensável à abertura destes horizontes, carecendo de ferramentas de trabalho que permitam precisar e sedimentar conceitos.

O sistema **MS(E)** (**Modelador de Sólidos (Experimental)**) foi realizado com o intuito de servir como banco de ensaio para o ensino e desenvolvimento na área da Computação Gráfica tridimensional.

### 2. Resumo Histórico

Desde 1985 que começaram os esforços no sentido de dispôr de um sistema que suportasse, por um lado, o desenvolvimento de algoritmos para cálculo de visibilidade em cenas tridimensionais [Sou87], por outro, o estudos de processos interactivos de aquisição das descrições de modelos de objectos. O protótipo **GAUDI1.0** (Macintosh 512 + UCSD Pascal) concluído nos finais de 86, foi o primeiro passo nesse sentido e incluía a possibilidade de contruir, facilmente, modelos de sólidos simples, através de um operador de varrimento translacional interactivo e de uma figura plana poligonal desenhada num editor 2D desenvolvido para esse efeito. O núcleo geométrico era pouco fiável e a complexidade dos modelos era muito

limitada (entre outras razões pelo limite a 64K de Data Area imposto pelo compilador utilizado), no entanto, a máquina oferecia a possibilidade de desenhar uma 'interface' com os utilizadores simples e agradável.

A segunda versão deste protótipo realizada em 1986/87 (GAUDI2.0) escrita em TML Pascal para o Macintosh Plus, ultrapassou algumas das limitações da versão anterior, passando o núcleo geométrico a ser bastante fiável e aumentando o número de operadores interactivos para construção de modelos de modo a permitir varrimentos translacionais, rotacionais e ainda a modificação da posição espacial dos vértices de maneira independente. A este protótipo foi ligado um módulo de cálculo de visibilidade ('scan-line') [BSF87].

Finalmente, o sistema MS(E), escrito em C e LISTPACK (Carnegie-Mellon) numa estação gráfica HP9000/320, dispõe de uma versão melhorada do núcleo geométrico GAUDI2.0 e foram acrescentados os módulos para transformações geométricas dos modelos, modelação de movimento [CoB88], cálculo de visibilidade e iluminação ('ray-tracing') [AIB88]. As rotinas gráficas utilizadas pertencem a "Starbase", um pacote gráfico de orientação CGI realizado pela Hewlett-Packard.

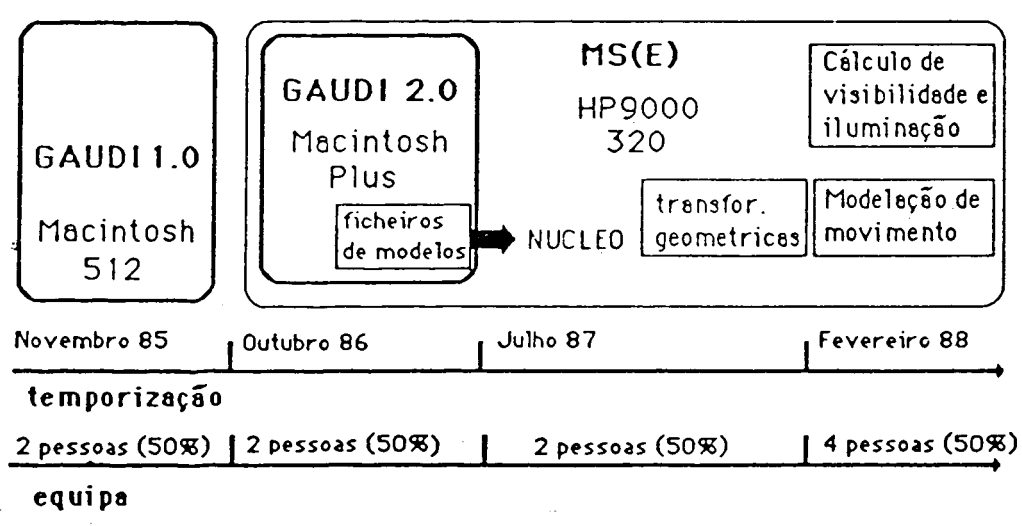
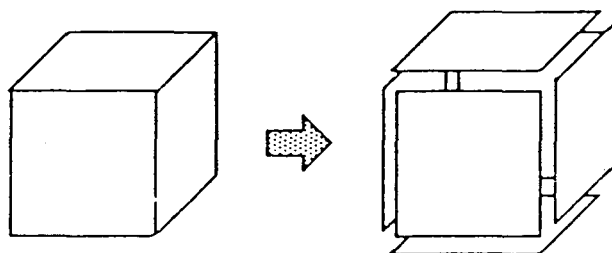


Fig.1 - Evolução do sistema MS(E)

A construção dos modelos dos objectos continua a residir no Macintosh, sendo os ficheiros correspondentes exportados para a estação gráfica HP para aí poderem sofrer outro tipo de tratamento.

### 3. GAUDI1.0/2.0- O Núcleo Geométrico e a Construção de Modelos de Objectos Tridimensionais

O esquema usado para a representação de sólidos é do tipo B-REP (Boundary-REPresentation), ou seja, a descrição é feita através da sua fronteira (fig.2). Os modelos são descritos pelo conjunto das suas faces, arestas e vértices. Este tipo de representação não é ambíguo assegurando, assim, que as descrições dos modelos sejam uma fonte de informação correcta para uma gama variada de aplicações (cálculo de propriedades geométricas, análise de interferências, etc.).



**Fig.2 - Representação B-REP**

As B-REPS são complexas. Objectos com interesse prático podem conduzir a representações com centenas de faces, vértices e arestas.

Os operadores de Euler (definidos inicialmente por Baumgart [Bau75]) são operações elementares que permitem a criação ou a alteração das representações, dividindo-as em acções atómicas, que garantem a validade topológica dos modelos construídos. Estes operadores baseiam-se na fórmula de Euler que relaciona os vários componentes topológicos da descrição de um objecto.

$$v - a + f = 2 * (s - b) + c$$

(v) (f) (a)- número de vértices, faces e arestas

(s)- número de componentes da "fronteira"

(b)- número de buracos através do corpo do sólido

(c)- número total de contornos interiores existentes em todas as faces do sólido.

Num objecto complexo, são comuns as ocorrências múltiplas do mesmo componente elementar ou de outros compostos. Aparece assim a noção de *instância* [Bra85] para economizar na quantidade de memória gasta na representação do modelo, na medida que com este processo um determinado componente só precisa de ser descrito uma vez. A cada instância está associada uma posição e orientação no espaço o que a torna distinta de outras referentes ao mesmo componente.

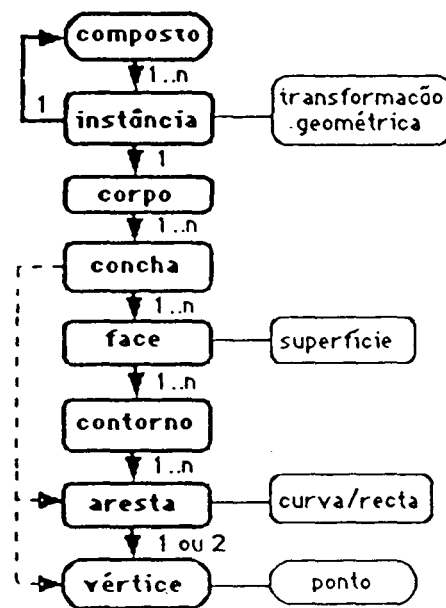


Fig.3 - Hierarquia topológica

Forma-se assim, um grafo acíclico e dirigido com objectos compostos nos nós não terminais, com componentes elementares nos terminais e instâncias nas ligações entre os nós.

A figura (fig.3) mostra a hierarquia de relações topológicas, assim como a ligação dos dados geométricos a cada um dos items topológicos respectivos. Ligada às faces e arestas pode aparecer outro tipo de informação nem topológica nem geométrica como, por exemplo, dados referentes ao aspecto (características superficiais, etc). A este nível foi usada a estrutura de dados conhecida por "winged-edges", desenvolvida por BAUMGART em Stanford (1972) para modelar poliedros de faces planas.

O núcleo, apresentado, está longe de constituir uma via de comunicação natural entre o computador e um utilizador, pelo que se desenvolveu um

conjunto de facilidades que tornassem o sistema viável em ambientes de utilização sem pré-requisitos de formação em Informática.

A estrutura global do conjunto do protótipo é apresentada na figura (fig.4).

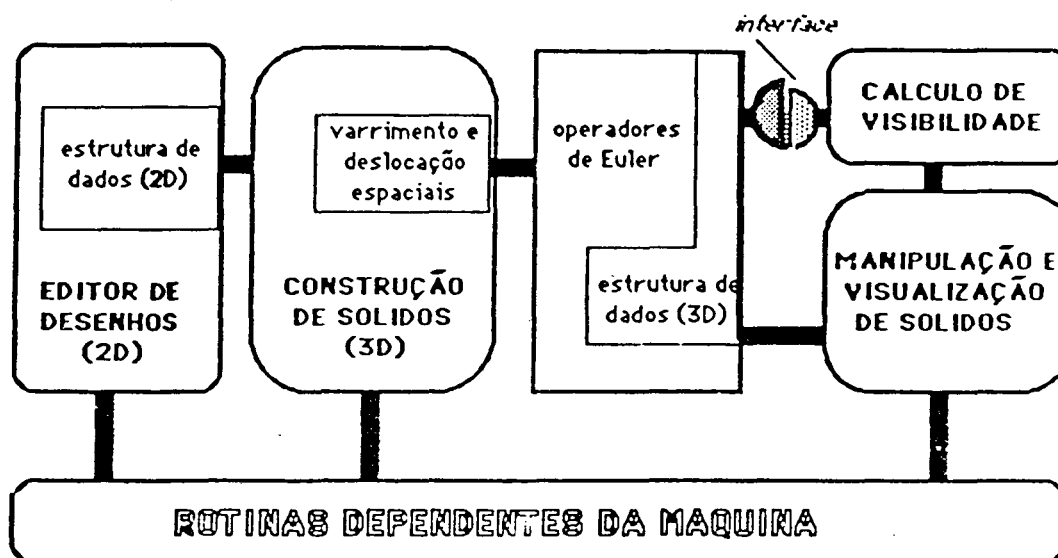


Fig.4 - A arquitectura do protótipo

Os desenhos 2D (formados por segmentos de recta) serão a base da construção "automática" (sem intervenção interactiva do utilizador) de uma "lâmina" (sólido abstracto, sem espessura) com a mesma topologia do desenho, mas no espaço tridimensional.

Um processo interactivo de **varrimento espacial** dará ao objecto a forma desejada. O **varrimento espacial** introduz alterações topológicas no modelo traduzidas (para o caso translacional) pela criação de faces laterais.

#### 4. O Sistema MS(E)

As modificações feitas no núcleo geométrico limitam-se à introdução de mais informação na estrutura de dados (fig.5) de modo a ser possível o desenvolvimento de novos módulos.

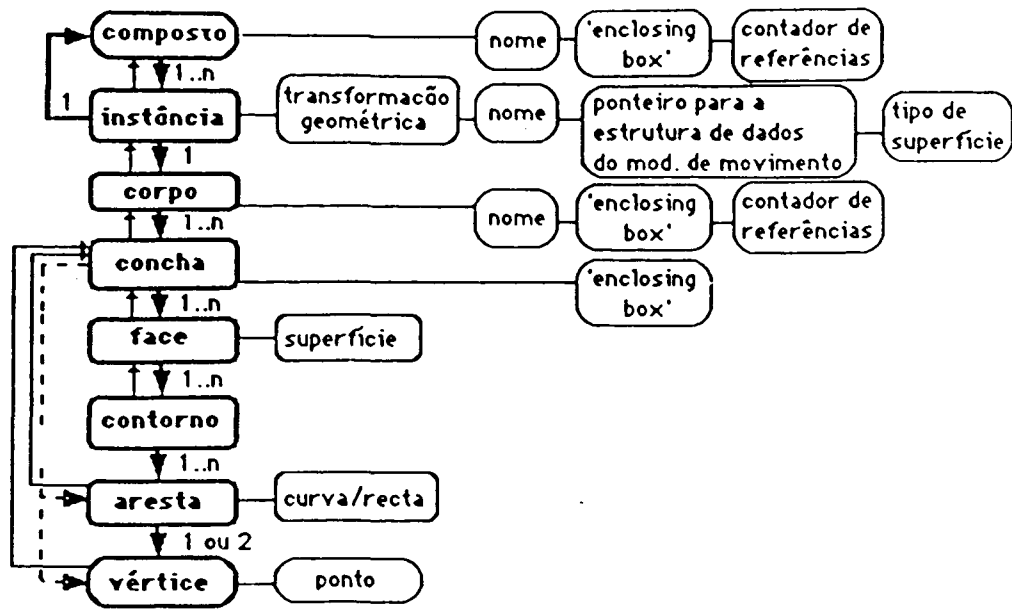


Fig.5 - A 'nova' hierarquia topológica

A "caixa envolvente", agora incluída na estrutura de dados, permite a realização de testes geométricos rápidos, quando não se torna necessário grande rigor a esse nível (por exemplo ao centrar uma figura numa janela). Os ponteiros "para trás" vieram facilitar e tornar mais rápida toda a manipulação da estrutura hierarquica e os "contadores de referências" permitem uma gestão mais eficiente do mecanismo de instâncias.

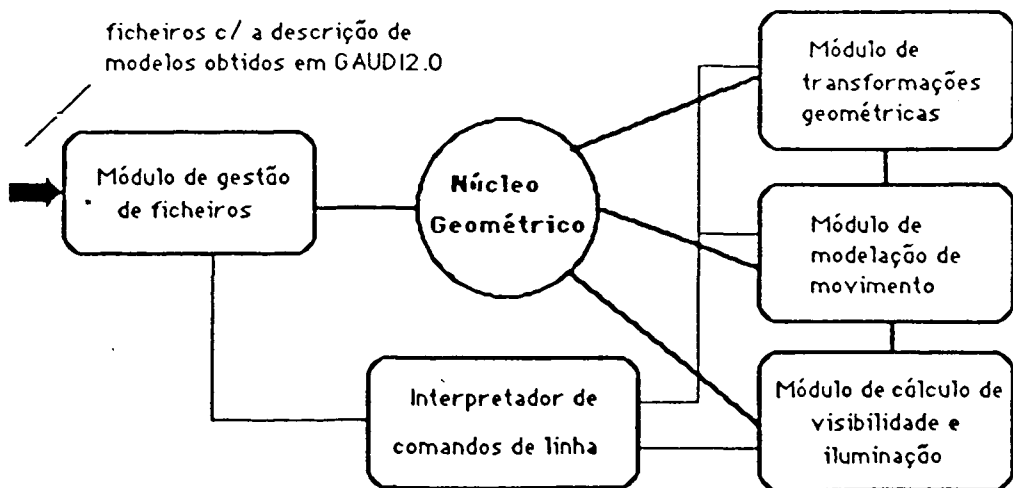


Fig.6 - A arquitectura do sistema MS(E)

O MS(E) só dispõe (neste momento) de um interpretador de comandos de linha (excepto em alguns comandos) para comunicar com o utilizador, tornando-se, deste modo, os nomes dos objectos num atributo fundamental para a sua identificação.

**Os novos comandos**

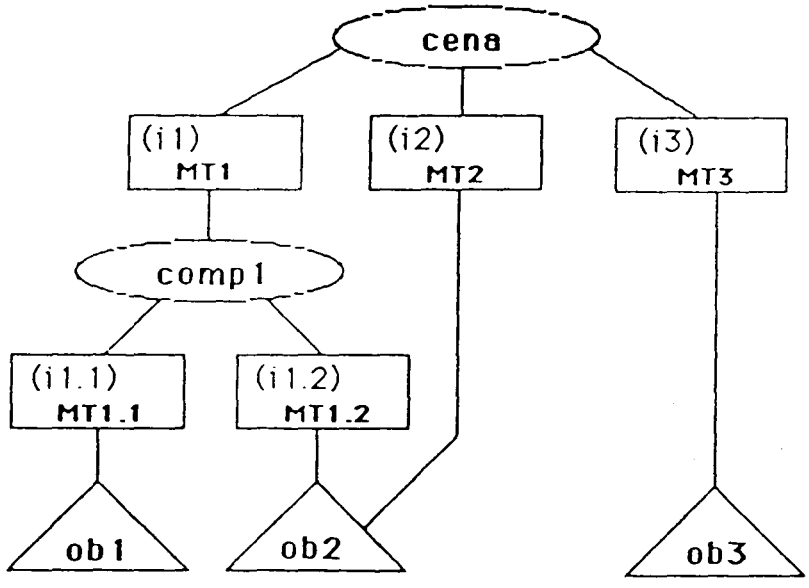
Todos os comandos obedecem a uma sintaxe comum:

<comando> [<nome do objecto>] [<parâmetros>]

**• Transformações geométricas**

- **translação, rotação, escalamento** - De acordo com a hierarquia apresentada a aplicação das transformações faz-se ao nível da instância, requerendo uma estrutura de dados adicional ('stack' de matrizes de transformação) para o processo se tornar flexível.

Para a cena apresentada na figura (fig.7), ao desenhar o objecto **ob2** (instância **i1.2**), todos os pontos correspondentes aos seus vértices deverão ser transformados pela matriz **MT1.2\*MT1**. No entanto, para a instância **i2** (do mesmo objecto) essa transformação deverá ser feita pela matriz **MT2**.



**Fig.7 - Estrutura de uma cena em MS(E)**

O 'stack' de matrizes é gerido (o pacote 'Starbase' inclui rotinas que facilitam esta tarefa) de modo a apresentar no topo a matriz resultante correspondente ao objecto a tratar (fig.8).



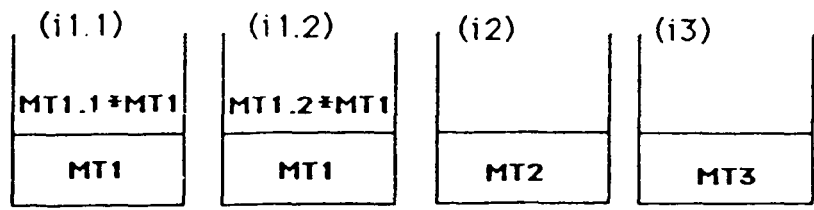


Fig.8 - O 'stack' de matrizes de transformação

Para que isto aconteça é feita, por cada nível da hierarquia visitado, a preconcatenação da matriz de instância com o topo do 'stack' com o consequente 'push' da matriz resultante. No regresso, à saída desse nó, é feito um 'pop', deixando no topo a matriz correspondente ao nível anterior.

Um problema surge ao aplicar sucessivas transformações a um composto e a um dos objectos simples seus constituintes, e isto devido não só ao facto de não ser arbitraria a ordem de aplicação das transformações, mas também porque o utilizador ao definir uma transformação para impôr a um objecto referencia-se sempre pelo sistema de eixos global (da cena). Se rodarmos em torno do eixo dos XX ( $R_x$ ) o composto **comp1** e seguidamente o objecto **ob1** em torno do eixo dos ZZ ( $R_z$ ), ao visualizar este último objecto teríamos no topo do 'stack':

$$MT1.1 * R_z * MT1 * R_x$$

o que não corresponde à ordem correcta definida para a aplicação das transformações, ou seja:

$$MT1.1 * MT1 * R_x * R_z \quad (1)$$

O esquema seguido para resolver este problema parte da obtenção de uma matrix resultante ( $MR = MT1 * R_x$ ) da preconcatenação de todas as matrizes de instância encontradas ao longo do percurso na hierarquia até ao objecto, sendo calculada a sua inversa  $MR^{-1}$ . A nova matriz de instância do objecto é então obtida da forma:

$$MT1.1 * MR * R_z * MR^{-1}$$

o que se traduzirá em (1) pela acção do mecanismo do 'stack' descrito atrás.

Excepto o escalamento, as outras transformações dispõem de uma versão interactiva.



- **edição de modelos**

- **nomes** - apresentação dos nomes dos vários objectos (simples e compostos) presentes na cena, juntamente com o desenho das suas 'caixas envolventes'.

- **duplicar** - constrói uma nova instância de um objecto, atribuindo a essa instância o nome do objecto original concatenado com o sufixo **\_in** (sendo **n** o valor do contador de referências) e actualizando o contador de referências desse objecto.

- **mudar** - transfere uma instância de um componente de um objecto composto para outro.

- **apagar** - permite apagar uma instância de um objecto. A remoção da descrição do objecto só será realizada se o respectivo contador de referências atingir zero.

- **animação** - invocação do módulo de modelação de movimento.

- **visualização**

- **ver** - este comando oferece como opções: a posição do observador (CIMA,FRENTE,LADO (fixas) e AXIO (variável)) , o tipo de vista (paralela ou em perspectiva) e a distância a que o observador se coloca (NORMAL,PERTO,LONGE).

- camera** - permite fixar a posição do observador no espaço.

- zoom** - para definir, de modo interactivo, a distância a que o observador se coloca, relativamente à cena.

- visib** - invocação do módulo de cálculo de visibilidade e iluminação ('ray-tracing' ).

- cores** - ajuste interactivo da paleta de cores (16) por variação dos parâmetros RGB separadamente com a possibilidade de definir uma escala monotónica. **Fundo** permite a selecção da cor de fundo para cada uma das janelas presentes.

- **ficheiros de objectos**

A descrição dos objectos é guardada em ficheiros de texto. Os comandos implementados possibilitam a leitura e gravação desses ficheiros e a consulta ao directório corrente.

## 5. Conclusão

O desenvolvimento do sistema MS(E) vai continuar no sentido de englobar novas hipóteses de construção e edição de modelos: operadores booleanos, levantamento a partir de vistas planas, reconhecimento de esboços são algumas das pistas possíveis.

Outra direcção a seguir na expansão deste sistema é a passagem para modelos de envolvente não fraccionada em faces planas e a inclusão no quadro da hierarquia topológica de representações de objectos baseadas em outros esquemas (por exemplo, definições analíticas). Neste caso serão possíveis duas vias: ou a uniformização das descrições geométricas se realiza de modo a ser possível ter algoritmos únicos para tratamento de qualquer modelo; ou a construção de algoritmos diferenciados por tipo de representação, que formem com estas entidades únicas.

## BIBLIOGRAFIA

[Bau75] B. Baumgart

A polyhedron Representation for Computer Vision  
AFIPS Proc., Vol.44, 1975

[Bra85] I. C. Braid

Geometric modelling

Notas preparadas para um tutorial da conferência EUROGRAPHIC'S 85

[Bra87] V. A. Branco

Modelação de sólidos: síntese de conceitos, realização de um protótipo  
Dissertação de Mestrado, Universidade do Porto, 1987

[Sou87] A. A. Sousa

Realização de um Algoritmo de Cálculo de Visibilidade

Síntese para provas de avaliação científica e pedagógica, FEUP  
Universidade do Porto, 1987

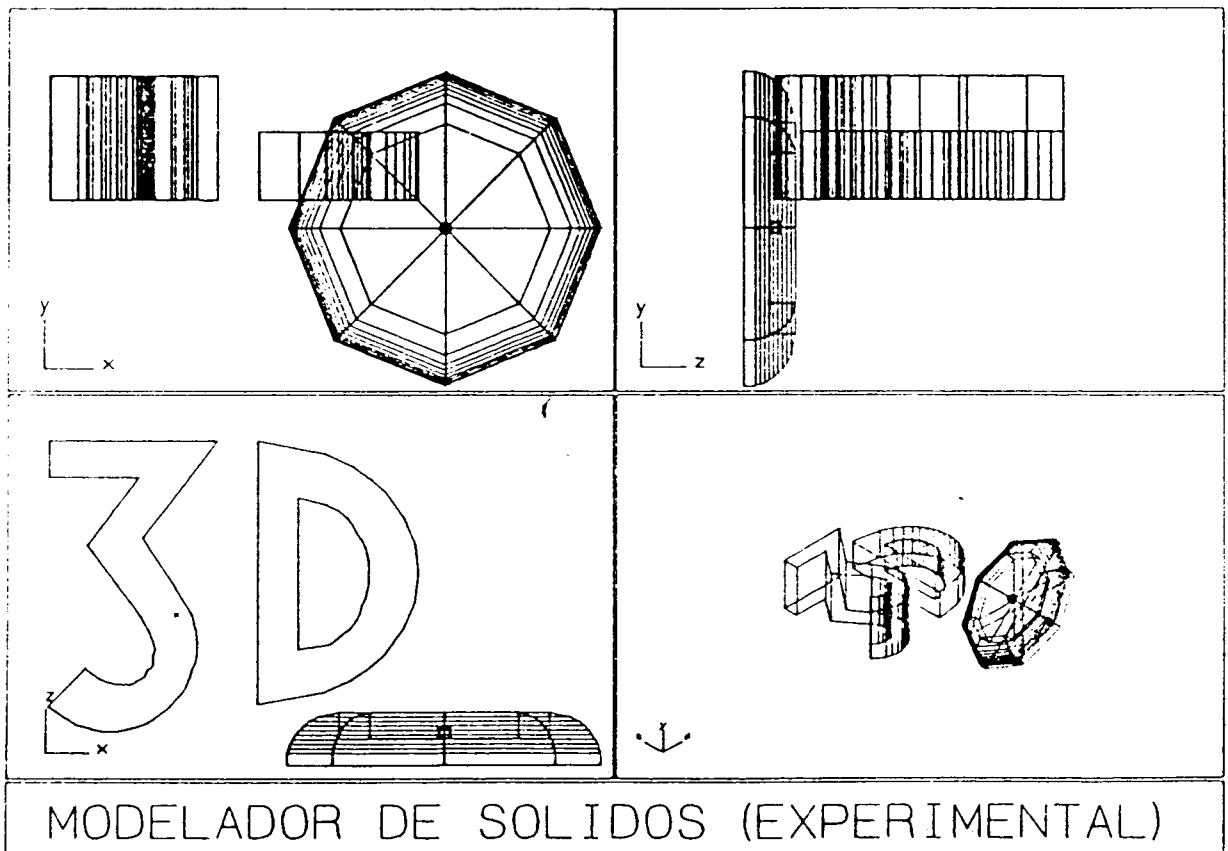
[BSF87] V. A. Branco, A. A. Sousa, F. N. Ferreira



Modelação e visualização de objectos tridimensionais  
1<sup>as</sup> Jornadas de PPPAC da Ordem dos Engenheiros, Lisboa, 1987

[CoB88] R. F. Costa, V. A. Branco  
Modelação de movimento no sistema MS(E)  
(a apresentar neste Encontro)

[AlB88] J. E. Almeida, V. A. Branco  
Adaptação de um algoritmo de 'ray-tracing' ao sistema MS(E)  
(a apresentar neste Encontro)



ISEP / INESC -- 1987/88