

PUC

Série: Monografias em Ciência da Computação, No. 12/91

FORMAS DE REPRESENTAÇÃO PARA OBJETOS TRIDIMENSIONAIS

Rolf Fischer

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP-22453
RIO DE JANEIRO - BRASIL

PUC RIO - DEPARTAMENTO DE INFORMÁTICA

Série: Monografias em Ciência da Computação, No. 12/91

Editor: Carlos J. P. Lucena

Junho, 1991

FORMAS DE REPRESENTAÇÃO PARA OBJETOS TRIDIMENSIONAIS *

Rolf Fischer

* Trabalho patrocinado pela Secretaria de Ciência e Tecnologia da
Presidência da República.

In charge of publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC Rio - Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453 - Rio de Janeiro, RJ
Brasil

Tel.:(021)529-9386 Telex:31078 Fax:(021)511-5645
E-mail:rosane@inf.puc-rio.br

Dissertação apresentada ao
Departamento de Informática da
PUC/RJ como parte dos
requisitos para obtenção do
título de Graduado em
Engenharia de Computação.

Orientador: Bruno Feijó

Curso de Engenharia de Computação
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 21 de Janeiro de 1990

RESUMO

Desde a sua existência, Sistemas de Modelagem Geométrica têm desempenhado um importante papel na integração de sistemas digitais à vida humana. A essência da sua virtude reside no fato de serem capazes de compreender o significado físico de objetos tridimensionais, quando descritos através de uma forma de representação adequada.

Dentro do contexto de Sistemas de Modelagem, o presente trabalho enfoca os principais aspectos relacionados com a análise e o desenvolvimento de formas de representação para Sólidos de Variedade simples. O seu conteúdo é organizado em duas partes, a saber: na primeira, são caracterizados Sistemas de Modelagem, e é definido um arcabouço de critérios para a avaliação de formas de representação; em seguida, faz-se, com base nesses critérios, uma abordagem completa das formas de representação mais conhecidas.

ABSTRACT

The use of Solid Modeling Systems represents a major breakthrough in the integration of digital systems into human living. Their power relies on the ability of understanding the nature of physical objects when described by an appropriate representation scheme.

This work focusses on the development and analysis of representation schemes for manifold objects in the context of Solid Modeling Systems. The content of this work is organized into two parts. The first major part, briefly presents a survey on geometric modeling systems field and introduces a framework for the evaluation of representation schemes. The second part uses that framework to describe and compare all the well-known representations.

SUMÁRIO

LISTA DE TABELAS E FIGURAS	VII
LISTA DE ABREVIATURAS E SIGLAS	XI
1 - INTRODUÇÃO	1
1.1 - Sobre o Presente Trabalho	4
1.2 - Suporte	5
<u>PARTE I - CARACTERIZAÇÃO DE SISTEMAS DE MODELAGEM GEOMÉTRICA</u>	<u>6</u>
2 - ANÁLISE GERAL DE SISTEMAS DE MODELAGEM GEOMÉTRICA	7
2.1 - Evolução Histórica	7
2.2 - Estágios dos Modelos de Representação	12
2.3 - Conceitos e Definições	14
2.4 - Classificação Geral de Sistemas	18
3 - CRITÉRIOS PARA A REPRESENTAÇÃO DE OBJETOS SÓLIDOS	21
3.1 - Motivação para Representações	21
3.2 - Critérios Matemáticos	23
3.3 - Sintaxe e Semântica	27
3.4 - Propriedades Formais e Informais	31
3.4.1 - Domínio	32
3.4.2 - Validade	33
3.4.3 - Completa	34
3.4.4 - Unicidade	36
3.4.5 - Concisão e simplicidade	38
3.4.6 - Eficiência	39
3.5 - Equivalência e Consistência em Representações Múltiplas	40

4 - FORMAS NÃO AVALIADAS DE REPRESENTAÇÃO	44
4.1 - Introdução	44
4.2 - Formas Ambíguas	46
4.3 - Instancias Puramente Primitivas	51
4.4 - Geometria Sólida Construtiva	56
4.4.1 - Álgebra Booleana	57
4.4.2 - Árvores CSG	58
4.4.3 - Propriedades	62
4.5 - Representação por Deslocamento	64
4.5.1 - Definição Matemática	67
4.5.2 - Generalizações	69
4.6 - Representação Analítica	72
4.6.1 - Hipermalhas	73
4.6.2 - Representação Interna	78
4.6.3 - Verificação de Critérios e Conclusões	81
5 - FORMAS AVALIADAS DE REPRESENTAÇÃO	83
5.1 - Enumeração por Ocupação Espacial	83
5.2 - Decomposição em Células Espaciais	88
5.3 - Representação por Fronteira	91
5.3.1 - Integridade Topológica	97
5.3.1.1 - Fórmula de Euler-Poincaré	103
5.3.1.2 - Integridade Geométrica	110
5.3.1.3 - Estruturas de Dados	115
5.3.1.3.1 - Diagrama de Schlegel	116
5.3.1.3.2 - Estrutura Básica	122
5.3.1.3.3 - Estrutura Winged-Edge	127
5.3.1.3.4 - Estrutura Winged-Edge Modificada	134
5.3.1.3.5 - Estrutura Vértice-Aresta	139
5.3.1.3.6 - Estrutura Ciclo-Aresta	141
5.3.1.4 - Classificação	143
6 - FORMAS HÍBRIDAS DE REPRESENTAÇÃO	151
7 - CONCLUSÕES	154
REFERÊNCIAS BIBLIOGRÁFICAS	157

LISTA DE TABELAS E FIGURAS

CAPÍTULO 1

Figura 1.1 - Sub-áreas da Modelagem Geométrica Tridimensional	2
---	---

CAPÍTULO 2

Figura 2.1 - Estágios da Modelagem Tridimensional	11
Figura 2.2 - Estrutura geral de um SMS	16

CAPÍTULO 3

Tabela 3.1 - Propriedades e questões associadas	32
Figura 3.1 - Exemplos de Sólidos de Variedade múltipla	26
Figura 3.2 - Domínio e contradomínio de representações	28
Figura 3.3 - Ambigüidade da representação por <i>Wire-Frame</i> ..	30
Figura 3.4 - Abstração ilusionista de objetos inválidos ...	33
Figura 3.5 - Consistência de múltiplas representações	40

CAPÍTULO 4

Tabela 4.1 - As representações mais conhecidas	44
Figura 4.1 - Representação gráfica através de projeções ...	47
Figura 4.2 - (a) <i>Cubo de Necker</i> , (b) e (c) abstrações possíveis	48
Figura 4.3 - Instâncias Puramente Primitivas	52
Figura 4.4 - Exemplo de Instâncias Puramente Primitivas ...	53
Figura 4.5 - Modelagem utilizando CSG	56
Figura 4.6 - Exemplo de uma árvore do tipo CSG	59
Figura 4.7 - Divisão do espaço por um HSP	61
Figura 4.8 - Sólido representado através de uma técnica de SWEEP	65
Figura 4.9 - Objetos inválidos gerados através de SWEEP ...	67
Figura 4.10 - SWEEP por um deslocamento de translação	68
Figura 4.11 - SWEEP por deslocamento de sólidos	70
Figura 4.12 - Sólido representado através de AREP	73

Figura 4.13	- O mapeamento de um <i>Hyperpatch</i>	74
Figura 4.14	- Curva de <i>Bezier</i> e os seus pontos de controle .	77
Figura 4.15	- Mapeamento extendido de um <i>HYPT</i>	80

CAPÍTULO 5

Tabela 5.1	- Relacionamento natural entre entidades	93
Tabela 5.2	- Conversão entre termos padronizados	97
Tabela 5.3	- Um possível conjunto gerador de <i>EOP</i>	108
Figura 5.1	- Malha espacial de representação	84
Figura 5.2	- Vista lateral de uma <i>SOE</i> baseada em <i>OCD</i>	86
Figura 5.3	- Furo perdido por uma baixa resolução	88
Figura 5.4	- Representação através de <i>SCD</i>	89
Figura 5.5	- Sólido representado através da <i>BREP</i>	92
Figura 5.6	- Operação geométrica local	96
Figura 5.7	- Mesma topologia para diferentes geometrias ...	98
Figura 5.8	- Algumas relações de adjacência	99
Figura 5.9	- Representação topologicamente válida	102
Figura 5.10	- Exemplos de <i>Homeomorfismos</i>	105
Figura 5.11	- <i>Faces</i> geometricamente inválidas	111
Figura 5.12	- <i>Vértice</i> <i>v9</i> geometricamente inválido	112
Figura 5.13	- Perda de informação durante a discretização .	114
Figura 5.14	- <i>Diagrama de Schlegel</i> para um cubo	118
Figura 5.15	- <i>Diagrama de Schlegel</i> para um objeto com <i>Genus</i> 1	120
Figura 5.16	- Descrição em <i>C</i> da estrutura de dados básica .	124
Figura 5.17	- Estrutura <i>Winged-Edge</i>	128
Figura 5.18	- "Percorrimento" de uma <i>Face</i> através da <i>WE</i> ...	130
Figura 5.19	- Impossibilidade de se orientarem <i>Arestas</i>	131
Figura 5.20	- Dupla representação da entidade <i>Aresta</i>	132
Figura 5.21	- Estrutura <i>Winged-Edge</i> modificada	136
Figura 5.22	- <i>Aresta</i> representada através da <i>mWE</i>	138
Figura 5.23	- Estrutura <i>Vértice-Aresta</i>	140
Figura 5.24	- Estrutura <i>Ciclo-Aresta</i>	142
Figura 5.25	- Transformação gerada por um <i>FTop</i>	150

CAPÍTULO 6

Figura 6.1	- Estrutura geral de um <i>HSMS</i>	153
------------	---	-----

LISTA DE ABREVIATURAS E SIGLAS

A

AMR	50
AREP	72

B

BEV	45
BNF	59
BOP	24
BREP	18

C

CA	141
CAD	10
CAM	10
CASE	3
cce	129
ccwe	130
CG	8
COMPA	3
CSG	18
CVIS	127
cw	137

D

DBMS	135
------------	-----

E

EBG	116
EOP	106
E ^a	17

F

FA	142
FEM	91
FTop	148

G

GKS	11
GMA	2
GMS	1
GMSDI	2
GMT	2

H

HSMS	152
HSP	60
HYPT	72

I

IA	4
IGS	3

M

MFLD	25
MI	12
MO	13
MS	13
MSP	12
mWE	135

N

NMFLD	155
-------------	-----

O

OCD	87
-----------	----

P

PPI	51
PROGE	3
PSP	75

R

REDO	109
------------	-----

S

SCD	88
SMS	15
SO	3
SOE	83
SRS	2
SSRF	113
SWEEP	64

U

UI	3
UNDO	109

V

VA	139
VLSIDM	155

W

WFORM	60
WFR	19

N^o

2D	7
2 $\frac{1}{2}$ D	7
3D	7
3DGM	2

CAPÍTULO 1

INTRODUÇÃO

Dentro das tradicionais fases de desenvolvimento de um sistema, encontra-se o projeto da estrutura de dados que aquele irá manipular. Dependendo das características particulares do sistema, essa fase pode ser vista como a mais importante, pois tende a comprometer todos os aspectos do desenvolvimento, da manutenção e das funcionalidades associados ao projeto final.

No contexto de Sistemas de Modelagem Geométrica (*Geometric Modeling System - GMS¹*), a importância atribuída a essa fase toma proporções ainda mais relevantes. Tal afirmação fundamenta-se na simples análise histórica dos principais GMS's, dos quais nenhum, até o momento, obteve sucesso na representação tridimensional de um objeto genérico. Revendo-se conceitos e tentativas, observou-se que o problema residia simplesmente na impossibilidade de uma máquina computacional representar todos os objetos encontrados na natureza, da mesma forma que é incapaz de representar um número irracional. É claro que nem sempre manipulam-se números irracionais, e nem sempre é necessário que se possuam infinitas casas de precisão. Máquinas de calcular, por exemplo, costumam oferecer apenas 7

¹ A definição exata do termo Sistema de Modelagem Geométrica será vista no próximo capítulo.

(sete) ou 8 (oito) casas de precisão e, no entanto, são largamente utilizadas. Analogamente, é satisfatório, em muitos casos, um Modelador Geométrico que seja capaz de modelar, por exemplo, apenas poliedros regulares.

Quando se aborda Modelagem Geométrica Tridimensional (*Three Dimensional Geometric Modeling - 3DGM*), estruturas de dados, simplesmente, não são relevantes; o são, no entanto, as formas de representação (*Solid Representation Schemes - SRS*) que um determinado sistema utiliza para mapear objetos físicos numa notação simbólica. Todos os aspectos, associados à utilização, avaliação e aplicação prática de GMS's, são, por conseguinte, fortemente dependentes das formas de representação oferecidas pelo mesmo. Sendo assim, quase todos os atuais estudos, relacionados com GMS's, dizem respeito à elaboração e análise de novas formas de representação, visando não somente às técnicas de modelagem, como também, aos sistemas aplicativos.

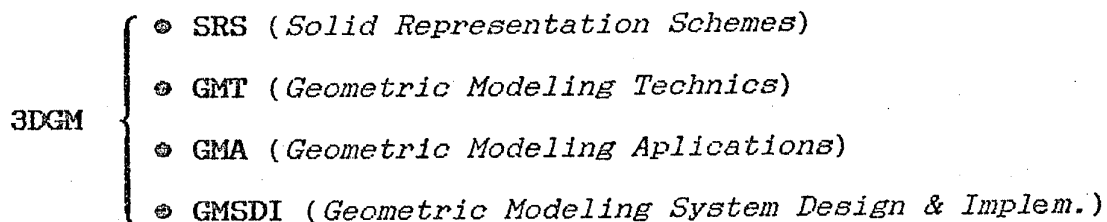


Figura 1.1 - Sub-áreas da Modelagem Geométrica Tridimensional

Não obstante a importância do estudo de formas de representação e das suas respectivas estruturas de dados para a área de Modelagem Geométrica Tridimensional, destacam-se, ainda, três fundamentais sub-áreas de pesquisa. São elas: Técnicas de Modelagem Geométrica (*Geometric Modeling Technics - GMT*), Aplicações de Sistemas de Modelagem Geométrica (*Geometric Modeling Applications - GMA*) e Projeto e Implementação de GMS's (*Geometric Modeling System Design & Implementation - GMSDI*).

A figura 1.1 ilustra a composição estrutural da área de Modelagem Geométrica Tridimensional. A SRS, como já foi explicado, trata do estudo dos esquemas simbólicos de informação que viabilizam a representação de entidades físicas numa máquina computacional. Já a GMT preocupa-se com o problema de se criarem ferramentas e metodologias, por meio de *Software* e *Hardware*, para a geração, edição, manutenção e conversão de formas de representação. Viabilizar, utilizar e integrar GMS's são tarefas da GMA. Por fim, tem-se a GMDI que objetiva a especificação das características do projeto e da implementação de GMS's, abordando outras áreas como as de: Interfaces com Usuário (*User Interfaces - UI*), Sistemas Gráficos Interativos (*Interactive Graphics Systems - IGS*), Ambientes de Programação (*Programming Environments - PROGE*), Sistemas Operacionais (*Operational Systems - OS*), Arquitetura de Computadores (*Computer Architectures - COMPA*), Engenharia de *Software* (*Software Engeneering - CASE*) e, mais recentemente, a área de

Inteligência Artificial (*Artificial Intelligence* - AI).

1.1 - Sobre o Presente Trabalho

Desde os anos 70, a área de Modelagem Geométrica Tridimensional tem sido a protagonista de um acentuado e crescente nível de pesquisa e desenvolvimento de sistemas. No Brasil, entretanto, essa área é ainda uma verdadeira incógnita, com apenas alguns trabalhos científicos já publicados. Dessa forma, abre-se o espaço para a necessidade de uma bibliografia de referência, capaz de abranger os principais conceitos, características, limitações e atuais enfoques na pesquisa da 3DGM.

O presente trabalho, focalizando apenas a *Solid Representation Schemes*, tem por objetivo a futura elaboração de uma dissertação de mestrado. Nesse contexto, faz-se necessária uma cuidadosa análise das principais formas de representação conhecidas e das suas limitações e virtudes.

De uma forma geral, esse trabalho procura abranger, de uma maneira consistente, as principais formas de representação desenvolvidas nos últimos 10 anos, sendo capaz de oferecer um guia de referência para a escolha, análise e implementação de novas representações e, é claro, de novos sistemas.

Com o objetivo de se delinear o contexto desse trabalho, é feita, no capítulo 2, uma breve análise dos principais Sistemas de Modelagem Geométrica conhecidos, das suas características gerais e de seus conceitos relacionados ao processo de Modelagem Tridimensional. No capítulo 3, procede-se à determinação de um preciso arcabouço de definições e critérios para a análise das mais importantes formas de representação, expostas nos capítulos seguintes (4, 5 e 6). Finalizando, resumem-se, no capítulo 7, as principais conclusões desse trabalho.

1.2 - Suporte

Esse trabalho foi desenvolvido no período de 1986 a 1990, suportado pelo *Grupo de Tecnologia em Computação Gráfica* da Pontifícia Universidade Católica do Rio de Janeiro (PUC/RJ), coordenado por *Albino J. Tavares* (Rio Data Centro - RDC), dirigido por *Marcelo Gattass* (Prof. Associado do Departamento de Engenharia Civil - PUC/RJ) e administrado pela Fundação Padre Leonel Franca (FPLF). Nesse período, o financiamento deveu-se à Financiadora de Estudos e Projetos (FINEP, convênio 5/2/85/0283), ao Banco do Brasil (FIPEC, convênio 1.1702-7) e ao Centro de Pesquisa da PETROBRÁS (CENPES).

Parte do conteúdo do presente trabalho foi previamente divulgado em Seminários e Palestras oferecidas dentro e fora da PUC-Rio, sendo também utilizado durante a implementação de um modelador protótipo (NADJA - 1987).

PARTE I

CARACTERIZAÇÃO DE SISTEMAS DE MODELAGEM GEOMÉTRICA

Nessa parte, são caracterizados Sistemas de Modelagem Geométrica e Sistemas de Modelagem Sólida, e é definido um preciso arcabouço de critérios para a avaliação e elaboração de novas formas de representação.

CAPÍTULO 2

ANÁLISE GERAL DE SISTEMAS DE MODELAGEM GEOMÉTRICA

2.1 - Evolução Histórica

Dentro das fases tradicionais do projeto de um produto¹, a especificação da sua forma é considerada, por muitos, como sendo o momento de maior importância. Dependendo das características de tal especificação, uma representação bidimensional (2D) pode conter todas as informações necessárias. Como exemplo, citam-se as aplicações para o projeto de circuitos impressos, plantas de construção, etc. Em alguns casos, utiliza-se uma forma de representação conhecida como semi-tridimensional ($2\frac{1}{2}$ D), onde se insere uma informação de nível, em três dimensões [10]. Aplicações como o projeto de *Chip's* ou de peças mecânicas (com algum tipo de simetria fixa) fazem uso, muitas vezes, desse tipo de representação. Entretanto, de uma forma geral, todas essas tentativas não passam de simplificações da representação tridimensional (3D) completa. Esta, por sua vez, impõe sempre uma grande complexidade à metodologia de representação de objetos reais

¹ Os aspectos, normalmente, de maior relevância durante o projeto de um produto são quanto à função, forma, manufaturabilidade e manutenção e quanto aos fatores econômicos.

3D, ao invés de simples desenhos 2D ou projeções 3D.

Desde o início das pesquisas em Computação Gráfica (*Computer Graphics - CG*), várias tentativas têm sido feitas, a fim de se reunir interação gráfica a programas para o projeto de sistemas físicos, como aviões, carros, construções, navios, peças mecânicas, etc.. Dentre essas tentativas, destaca-se o programa SKETCHPAD, desenvolvido por *Ivan Sutherland* (MIT, 1960's). *Sutherland* foi o primeiro pesquisador a ilustrar a possibilidade de se combinar interação gráfica em tempo real, através da geração de imagens gráficas projetadas num tubo de raios catódicos [42].

Esses esforços têm sido, no entanto, apenas parcialmente recompensados, tendo em vista a utilização, até então, de desenhos como meios para possibilitar a interação homem-máquina. Por serem bidimensionais, os desenhos, quando utilizados na representação de objetos tridimensionais, são, ao mesmo tempo, redundantes e incompletos, gerando, conseqüentemente, ambigüidades de todo tipo.

A razão maior dessa conclusão reside no fato de engenheiros e técnicos possuírem, de uma forma intrínseca, uma grande bagagem de conhecimentos gerais, capacitando-os a extrair dos desenhos toda a informação necessária para a

correta montagem e manufatura dos objetos neles representados².

Apenas recentemente, notou-se que um computador poderia ser utilizado como uma forma de mídia, da mesma maneira que os desenhos. Ao invés de se representarem os desenhos em um computador (segundo a filosofia dos sistemas contemporâneos e incorporando, conseqüentemente, as suas limitações), este poderia reproduzir diretamente o objetivo de tal representação, sendo, assim, o sistema físico tridimensional projetado. Dessa forma, poder-se-ia abstrair, de um único modelo, a informação requerida normalmente por vários desenhos e especificações técnicas.

Tal capacidade, associada a essa nova metodologia de projeção pode ser observada a partir do seguinte exemplo comparativo: uma fábrica de bicicletas deseja construir uma polia nova para rodas, a fim de substituir a peça atual. O projeto dessa polia é feito por um engenheiro, que repassa os dados (provavelmente em forma de desenhos ou rascunhos) para um desenhista. Este, por sua vez, refaz o projeto, seguindo as normas de desenho técnico e o envia para a seção de manufatura. Nessa última fase, entretanto, o torneiro encontra uma incoerência no projeto, pois uma das vistas indica um raio

² Como exemplo desses conhecimentos pode-se citar: a utilidade do dispositivo em questão, princípios gerais de mecânica, formalismos técnicos e convenções da área em questão.

interno diferente do especificado em uma outra vista. Independentemente da causa, o erro será sempre consequência da perda de informação ocorrida entre o mapeamento tridimensional (no qual a polia foi inicialmente projetada pelo engenheiro) e a representação bidimensional (feita pelo desenhista), onde, de fato, existe a troca de informação. Ao se proceder à análise da proposta oferecida pelo modelo tridimensional, conclui-se estarem esses problemas resolvidos. Nessa abordagem, o engenheiro projetaria toda a polia em três dimensões, ou seja, geraria uma representação completa e repassaria para a seção de manufatura apenas as vistas ou cortes que fossem necessários. Caso essa seção já dispusesse de equipamentos de manufatura controláveis por programação numérica, esses poderiam ser programados automaticamente *on-line* a partir da representação tridimensional da polia. Com isso, além de se eliminarem todos os tipos de erros de interpretação, a manufatura seria precisa e consistente, dispensando os desenhos bidimensionais como forma de mídia, tornando-os apenas uma documentação auxiliar.

Atualmente, já existem máquinas de manufatura programáveis para até 6 (seis) eixos e outras ainda para a injeção de materiais plásticos, capazes de integrar processos de CAD (*Computer Aided Design*) e CAM (*Computer Aided Manufacturing*) [37]. Esta é uma tendência atual na linha de *Factory Automation e Design Automation*.

De qualquer forma, é importante ressaltar que tal processo só terá êxito e sentido, caso se possa garantir que o modelo de representação seja realmente completo, não ambíguo, único e conciso (o que será mais profundamente estudado no capítulo 3). Observa-se, portanto, que a evolução natural dos antigos sistemas trouxe a necessidade de se possuir um GMS que fosse capaz de modelar não apenas primitivas bidimensionais (2D), mas também tridimensionais (3D)³ [36].

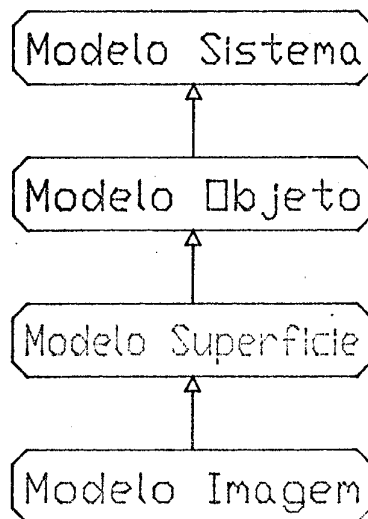


Figura 2.1 - Estágios da Modelagem Tridimensional

³ O conceito de Primitivas Gráficas deriva dos padrões gráficos mais conhecidos como o GKS (Graphical Kernel System); aquelas primitivas identificam entidades gráficas como linhas, pontos, marcas, textos ou o preenchimento de regiões.

2.2 - Estágios dos Modelos de Representação

Podem-se analisar os estágios associados à modelagem gráfica tridimensional, a partir de quatro níveis em ordem crescente de sofisticação e complexidade (figura 2.1).

A primeira tentativa reside na simples reprodução de desenhos, denominada Modelagem de Imagem (MI). Um Modelo Imagem contém a informação necessária para a visualização gráfica, incluindo, possivelmente, a eliminação de linhas e superfícies escondidas. Esse modelo é composto, usualmente, de uma seqüência não estruturada de superfícies.

O segundo nível de sofisticação são os chamados Modelos Superfície (MSP), que representam uma seqüência de superfícies, definindo um sub-espaco fechado e orientado. Um Modelo Superfície tende a ser espacialmente completo e contém, normalmente, toda a informação necessária para a derivação de qualquer propriedade espacial (inclusive informações de tolerância), permitindo que haja um conjunto de operações de manipulação da sua forma geométrica. Técnicas para a manipulação de superfícies são denominadas modelagens geométricas, apesar de atualmente já englobarem também as manipulações topológicas sobre o modelo.

Em seguida, há o nível, considerado até bem pouco tempo como o mais sofisticado, que adiciona ao modelo geométrico a inclusão de atributos topológicos, de material, acabamento, densidade, cor ou qualquer outro atributo ou característica que sejam relevantes às aplicações. Nesse nível, encontra-se o Modelo Objeto (MO) que, como será visto posteriormente, tem toda a sua estrutura fundamentada na utilização de entidades topológicas para a obtenção de uma série de exigências relativas à forma de representação.

Por último, existe o nível mais sofisticado, representando o "estado da arte", no que diz respeito aos estágios da Modelagem Tridimensional. Em tal nível, há uma preocupação crescente com a interação homem-máquina, particularmente no processo de criação dos objetos. Aqui, reside o Modelo Sistema (MS), caracterizado diretamente pelos GMS's paramétricos e por aqueles que possuem estruturas de dados auxiliares, registrando o contexto histórico da modelagem. Alguns trabalhos recentes procuram, também, englobar, dentro desse Modelo, o contexto de Sistemas Inteligentes, relacionados diretamente aos conceitos de Inteligência Artificial, o que provavelmente conduzirá, nos próximos anos, a um novo Modelo na hierarquia da Modelagem Tridimensional.

2.3 - Conceitos e definições

Os termos Sistema de Modelagem Geométrica, Modelador Geométrico, Sistema de CAD/CAM, etc., têm sido utilizados já há muito tempo de uma forma bastante indisciplinada, quando não, contraditória. Tal fato se deve, principalmente, a três razões:

- (i) O desconhecimento puro dos profissionais que, mesmo sendo usuários desses sistemas, não receberam um embasamento teórico sobre o assunto, ou seja, a cultura difundida não foi definida através de uma metodologia formal e precisa;
- (ii) Ausência de uma padronização internacional dos conceitos, da metodologia e das funcionalidades mínimas dos sistemas já em uso comercial ou em desenvolvimento;
- (iii) Em função de uma simples estratégia de *marketing*, grande parte dos fabricantes de *Hardware* e *Software* proclamam que os seus produtos oferecem uma completa solução nessa área, distorcendo terminologias e conceitos, a fim de adequarem os seus produtos às nomenclaturas por eles criadas ou alteradas.

A união desses fatores, aliada a um crescente desenvolvimento de *Hardware* e *Software*, tem originado um verdadeiro caos na conceitualização e avaliação de sistemas.

Sendo assim, determinar-se-á a definição exata dos termos utilizados no presente trabalho.

Os termos Sistemas de Modelagem Geométrica (*Geometric Modeling System - GMS*) e Sistemas de Modelagem Sólida (*Solid Modeling System - SMS*) possuem, geralmente, um significado meio confuso, quando não, equivalente, para a maioria das pessoas. Dentro do contexto dessa dissertação, define-se o termo Sistema de Modelagem Geométrica como:

Sistema que prevê facilidades para a entrada, armazenamento e modificação da representação de objetos tridimensionais [39].

Dentro dessa definição, pode-se afirmar que todos os sistemas contemporâneos de CAD/CAM são, dentro de alguns limites, Sistemas de Modelagem Geométrica, pois todos modelam, de alguma forma, entidades geométricas.

O que distingue os Sistemas de Modelagem Sólida é o fato desses sistemas armazenarem e manipularem representações completas e não ambíguas da geometria e, eventualmente, da topologia de objetos sólidos [35]. Essas representações serão analisadas em detalhes nos capítulos 4,5 e 6.

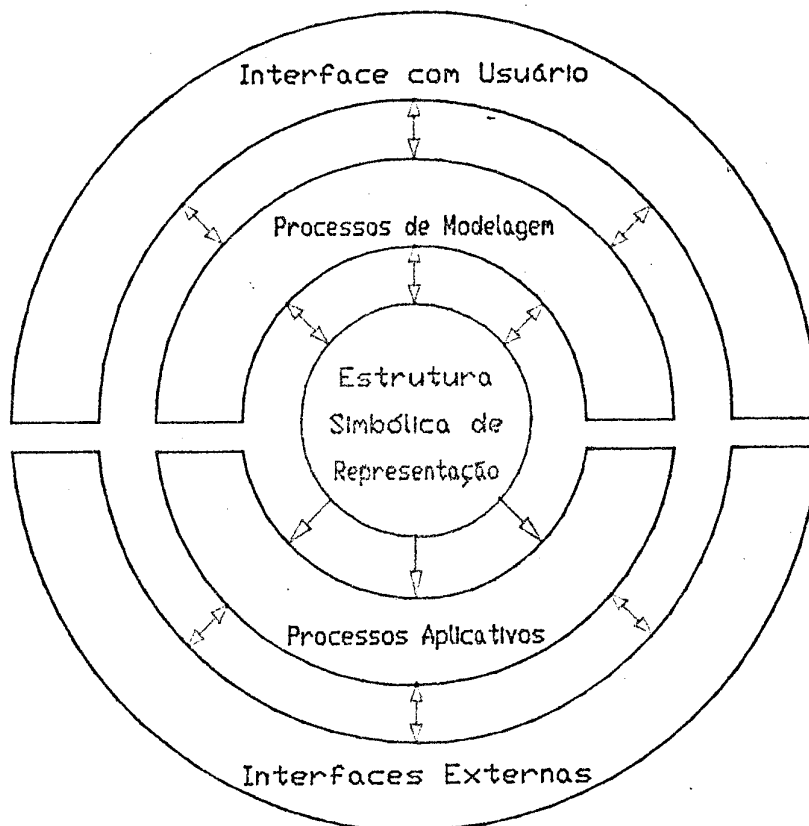


Figura 2.2 - Estrutura geral de um SMS

A estrutura de um SMS (figura 2.2), dentro de uma ampla generalização, o obriga a possuir:

- (i) Estrutura simbólica para a representação de objetos sólidos;
- (ii) Processos que utilizem tal representação para a solução de questões (relativas aos objetos)

geométricas, topológicas e de atributos;

- (iii) Facilidades de entrada, isto é, meios de se criarem e editarem representações de objetos, e de serem invocados processos de manipulação da mesma;
- (iv) Facilidades de saída para a apresentação dos resultados e comunicação com outros sistemas [50].

Fazem-se, aqui, algumas observações esclarecedoras com relação aos conceitos apresentados anteriormente:

- (1) No coração de qualquer SMS, reside uma estrutura de dados simbólica (representação), designando, a "objetos abstratos", a tarefa de modelarem "objetos reais";
- (2) Os sólidos, ou simplesmente objetos (como serão tratados nesse trabalho), estão definidos, exclusivamente, dentro de um subconjunto do espaço euclideo (E^3);
- (3) A definição acerca da estrutura de um SMS, feita no item (1), não especifica, em momento algum, qual a técnica a ser utilizada para a representação interna e para a entrada e saída da geometria dos objetos;
- (4) Não existe nenhuma conexão intrínseca entre SMS e a produção de imagens com textura. Esses dois termos são geralmente alvo de muita confusão, provavelmente em razão do fato de grande parte dos SMS produzirem

imagens, nas quais os objetos possuem uma aparência sólida.

2.4 - Classificação Geral de Sistemas

Existem, atualmente, diversas formas de se mapearem os modelos de representação em um sistema digital. Entre as técnicas mais conhecidas, destacam-se as Representações por Fronteira (*Boundary Representation - BREP*⁴) e as Geometrias Sólidas Construtivas (*Constructive Solid Geometry - CSG*⁵). Ambas têm sido bastante utilizadas e, conseqüentemente, têm se constituído erroneamente no parâmetro de classificação de SMS.

Uma classificação precisa só pode ser obtida ao se analisar um sistema no contexto do Modelo, que aquele enfatiza (Imagem, Superfície, Objeto ou Sistema), e sua representação interna (CSG, BREP, etc.). Tal afirmação fundamenta-se no fato de grande parte dos sistemas mais sofisticados utilizarem duas ou mais formas de representação interna para darem suporte ao mesmo Modelo [5,27].

⁴ Boundary Representation é uma forma de representação, na qual um objeto é descrito completamente através da definição da sua fronteira com o espaço, delimitando, portanto, exatamente a interface entre o seu interior e o seu exterior.

⁵ Constructive Solid Geometry é uma técnica de representação, na qual um objeto é descrito por uma árvore binária, cujos nós são operadores booleanos e cujas folhas contêm primitivas geométricas básicas (blocos, esferas, cilindros, etc.).

Sistemas contemporâneos podiam, outrora, ser facilmente classificados entre os BREP (BUILD, GLIDE, ROMULUS, DESIGN, EUCLID, MEDUSA, DESIGNBASE, etc.) e os CSG (PADL-1, PADL-2, GMSOLIDS, TIPS, GDP, SYNTHAVISION, UNISOLIDS, etc.) [2,11,40,41]. Atualmente, esse panorama sofreu grandes alterações. Como será visto mais adiante, nenhuma forma de representação é uniformemente ideal para todas as aplicações, e, conseqüentemente, muitos sistemas utilizam múltiplas representações. PADL-1, PADL-2, GMSOLIDS, GDP e UNISOLIDS, por exemplo, utilizam tanto BREP como CSG, como formas de representação. Essa estratégia possibilita uma grande flexibilidade no projeto de algoritmos dedicados às aplicações, pois naqueles existe a possibilidade de se utilizar a representação mais adequada. O sistema UNISOLIDS, por exemplo, utiliza a Representação por Fronteira para a geração de imagens simples (do tipo *Wire-Frame* - WFR⁶) e a representação, em árvore booleana para o resto dos algoritmos. O preço que se paga, na utilização de mais de uma forma de representação se apresenta nitidamente nos algoritmos de conversão entre as representações e na conseqüente necessidade de se realizar uma manutenção constante da consistência do Modelo para todo o sistema.

⁶ *Wire-Frame* é uma técnica de visualização, na qual o objeto é apresentado graficamente através de segmentos de reta, representando as suas Arestas.

De qualquer forma, é notória a tendência futura de utilização pelos sistemas de mais de uma forma de representação como meio de possibilitar uma maior sofisticação dos métodos de interação e uma maior generalização dos objetos que efetivamente podem ser modelados.

CAPÍTULO 3

CRITÉRIOS PARA A REPRESENTAÇÃO DE OBJETOS SÓLIDOS

3.1 Motivação para Representações

Como foi visto no capítulo anterior, a maioria dos SMS's atualmente disponíveis para a utilização em instalações industriais de CAD/CAM possuem ferramentas sofisticadas para a interação homem-máquina; sofrem, no entanto, problemas de representação em quase todos os seus aspectos.

Para que um objeto tridimensional possa ser modelado em um SMS, é necessário que este seja capaz de "entender", de alguma maneira, o que tal objeto significa em termos físicos. A realização da tarefa de "entender" o significado físico de um objeto só é possível em uma máquina computacional, através de uma estrutura simbólica do objeto, denominada forma ou esquema de representação. A razão maior para tal fato decorre da incapacidade, dos algoritmos geométricos de manipularem objetos físicos diretamente, mas somente as suas representações simbólicas.

A complexidade envolvida na busca de uma forma de representação adequada é de difícil avaliação. Na tentativa de se formalizar essa análise, necessita-se de um completo e preciso arcabouço de critérios para a caracterização e avaliação das formas de representação disponíveis.

Um outro problema imediato, com o qual se confronta, ao serem estudadas novas formas de representação, é a impossibilidade, até o momento, de se fazer uso de qualquer abstração dos sentidos humanos para a formalização de uma estrutura de representação com estes compatível. Em outras palavras, o problema recai sobre o fato da máquina computacional ser incapaz de representar, de forma simbólica e com plenitude e exatidão, os objetos, alvo da experiência, que o ser humano adquire durante toda a sua vida, objetiva e subjetivamente.

Uma arcabouço de critérios foi desenvolvido por *Arestides A.G. Requicha* [39], para permitir uma precisa avaliação das formas de representação conhecidas e das novas a serem projetadas. Tais exigências correspondem ao conjunto mínimo de pontos que devem ser observados, seja durante o desenvolvimento e implementação de um SMS, seja quando da fase de utilização do mesmo. Em muitos casos, o conhecimento prévio das limitações exatas de um sistema permite ao usuário contornar os problemas, adaptando-se a elas, e,

conseqüentemente, explorando ao máximo a potencialidade do dado sistema.

É objetivo desse conjunto de definições que se estabeleça, com exatidão, o que se entende matematicamente por um objeto sólido, e o que se entende formalmente por representações completas e equivalentes. Este último ponto é de fundamental importância para a elaboração de novas estruturas de dados que agreguem, em si, características e particularidades comuns a outras já existentes.

3.2 Critérios Matemáticos

Como já foi mencionado anteriormente, a primeira premissa a ser adotada ao se abordar uma forma particular de representação, consiste em se garantir que esta seja realmente capaz de modelar objetos físicos e de produzir representações inteiramente contidas no E^3 (espaço euclidiano). As limitações e capacidades associadas à geometria euclidiana já são bem conhecidas, mas apenas alguns subconjuntos do E^3 são adequados à modelagem de objetos físicos. São estabelecidos, então, alguns critérios matemáticos acerca de um sólido genérico (ou abstrato), para que este possa ser representado e manufaturado através de um SMS [31].

- (i) **Rigidez** - Um sólido genérico deve possuir uma forma invariante, independentemente da sua localização e orientação no espaço;
- (ii) **Homogeneidade** - Um sólido, para ser considerado homogêneo tridimensionalmente, não deve possuir partes (ou entidades) isoladas ou "penduradas" na sua fronteira com o espaço. Além disso, exige-se que possua um interior bem definido;
- (iii) **Finito** - Um sólido deve ocupar um volume finito do espaço, no qual se encontra inserido;
- (iv) **Fechado** - Este critério é aplicado a todas as Operações Booleanas (*Boolean Operations* - BOP¹) e de transformação que possam ser efetuadas sobre um sólido abstrato, exigindo que o seu resultado (ou uma composição deste) também produza objetos sólidos;
- (v) **Discretizável** - Independentemente das entidades (3D), que se utilizem para a representação de um sólido, aquelas devem ser finitas (em seu sentido contável). Por exemplo, um sólido deve possuir um número finito de vértices e arestas;
- (vi) **Fronteira determinável** - A fronteira de um sólido deve descrever, sem ambigüidades, o que é interior e exterior ao mesmo.

¹ Operações booleanas serão estudadas em detalhes na seção 4.4.1.

Matematicamente, tais critérios restringem o domínio dos sólidos que podem ser representados. Tal domínio equivale a um subconjunto do E^3 , denominado *Conjunto-R*, e que forma uma classe de congruência que obriga os sólidos, a ela pertencentes, a serem fechados, regulares, com fronteira determinável e semianalíticos. É importante notar que, sendo o *Conjunto-R* uma classe de congruência, este pode ser obtido de maneira única, a partir de uma coleção de conjuntos, nos quais são realizadas operações de translação, rotação e escala. Essa definição sugere, portanto, que um sólido seja modelado a partir de um único *Conjunto-R*.

De uma forma geral, pode-se considerar o *Conjunto-R* como sendo formado pelos poliedros regulares (curvos ou não), com uma topologia bem comportada, não excluindo-se, no entanto, a modelagem de sólidos com furos ou passagens. Além disso, será observado o fato do conjunto garantir a sua invariância com relação à aplicação de operadores booleanos regulares sobre o mesmo. O critério (ii), referente à homogeneidade de um sólido, merece uma atenção especial. A homogeneidade de um objeto tridimensional pressupõe, acima de tudo, que este não esteja unido a si próprio ou a outros objetos através de *Arestas* ou *Vértices* comuns. Objetos, que respeitam tal condição, são denominados Sólidos de Variedade simples (*Manifold Solids - MFLD*).

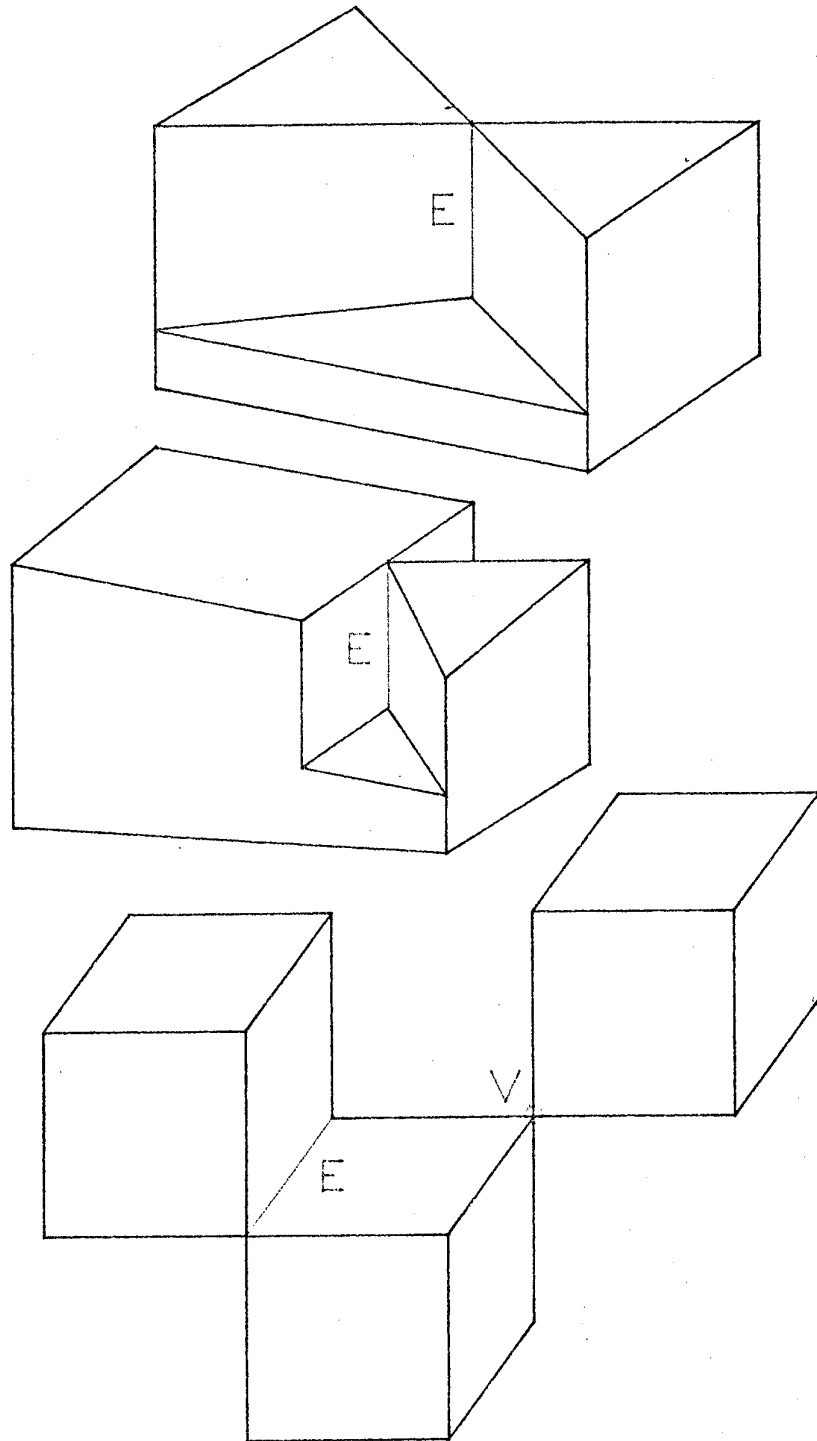


Figura 3.1 - Exemplos de Sólidos de Variedade múltipla

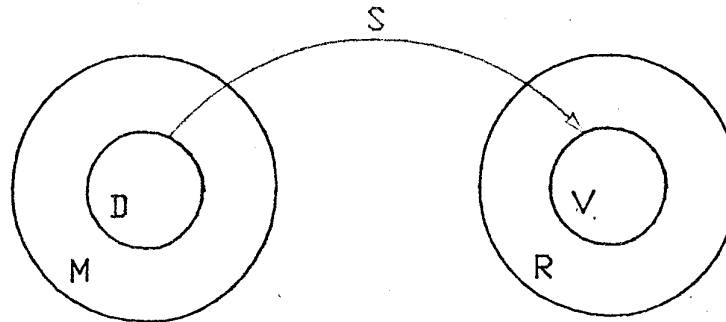
Exemplos de Sólidos de Variedade múltipla podem ser visualizados na figura 3.1. A exigência mencionada é normalmente garantida pelas estruturas simbólicas que implementam as principais formas de representação conhecidas. No entanto, é importante notar que tal restrição equivale a se exigir que apenas os sólidos, que efetivamente possam ser manufaturados, devam ser representados. Dessa forma, mesmo que objetos não múltiplos fossem representados, esses não poderiam ser realizados fisicamente. A razão para isso advém do fato de que *Arestas* e *Vértices* não podem ser modelados fisicamente de uma maneira infinitamente fina e precisa.

3.3 Sintaxe e Semântica

Uma forma de representação pode ser analisada, com precisão, a partir de sua sintaxe e semântica. É útil, conseqüentemente, que sejam estabelecidos critérios exatos para a determinação sintática e semântica de uma estrutura simbólica de dados.

Representações sintaticamente corretas são estruturas simbólicas, construídas a partir de símbolos, extraídos de um alfabeto, segundo algumas regras de sintaxe. Define-se, ainda, a coleção de todas as formas de representações sintaticamente corretas como um *Espaço de Representação R*. Um espaço de representação pode ser visto como uma linguagem gerada por

alguma gramática. É importante salientar, que não é feita nenhuma restrição com relação à forma sintática, pela qual uma representação deve ser implementada. Esta pode ser mapeada para um conjunto de caracteres, um grafo, uma árvore ou outra estrutura qualquer, sem, no entanto, estar comprometendo o seu espaço de representação.



- M** : Espaço de Modelagem
R : Espaço das representações sintaticamente corretas produzidas por uma gramática
S : Forma de representação
D : Conjunto dos elementos de M representáveis via S, caracterizando, portanto, o domínio de S
V : Conjunto de representações sintaticamente corretas, que são imagens dos elementos de D, ou seja, identifica o contradomínio de S

Figura 3.2 - Domínio e contradomínio de representações

A semântica de uma representação é, sem dúvida alguma, a parte mais importante relacionada à sua estrutura propriamente dita. Esta é obtida ao se associarem entidades

geométricas a uma representação. A qualidade e a quantidade da informação geométrica adicionada irão classificar uma forma de representação, determinando o seu domínio e contradomínio. A partir das definições e restrições com relação aos sólidos válidos, anteriormente estabelecidos, pode-se determinar um *Espaço de Modelagem M* como sendo aquele, cujos elementos são sólidos abstratos, levados ao *Conjunto-R* através de uma forma de representação (figura 3.2).

É possível, portanto, definir formalmente uma forma de representação *S* como sendo uma relação:

$$S : M \rightarrow R .$$

O domínio e o contradomínio de *S* são denotados por *D* e *V*, respectivamente. Qualquer representação que estiver no contradomínio *V* é dita válida, pois esta será sintática e semanticamente correta, ou seja, ela pertencerá a *R* e possuirá elementos em *D*. É importante ressaltar que, a partir da definição acima, não é assumido que todos os objetos são efetivamente representáveis e nem que todas as representações sintaticamente corretas são, por isso, naturalmente válidas. Matematicamente, tal fato é simples de ser demonstrado, pois:

$$(D \supseteq M) \text{ e } (V \supseteq R) .$$

Aproveitando ainda a definição acima, pode-se afirmar que uma representação W em V não é ambígua (completa), se esta corresponde a um único objeto, ou seja, se o conjunto $S^{-1}(W)$ só possuir um elemento em D (figura 3.3).

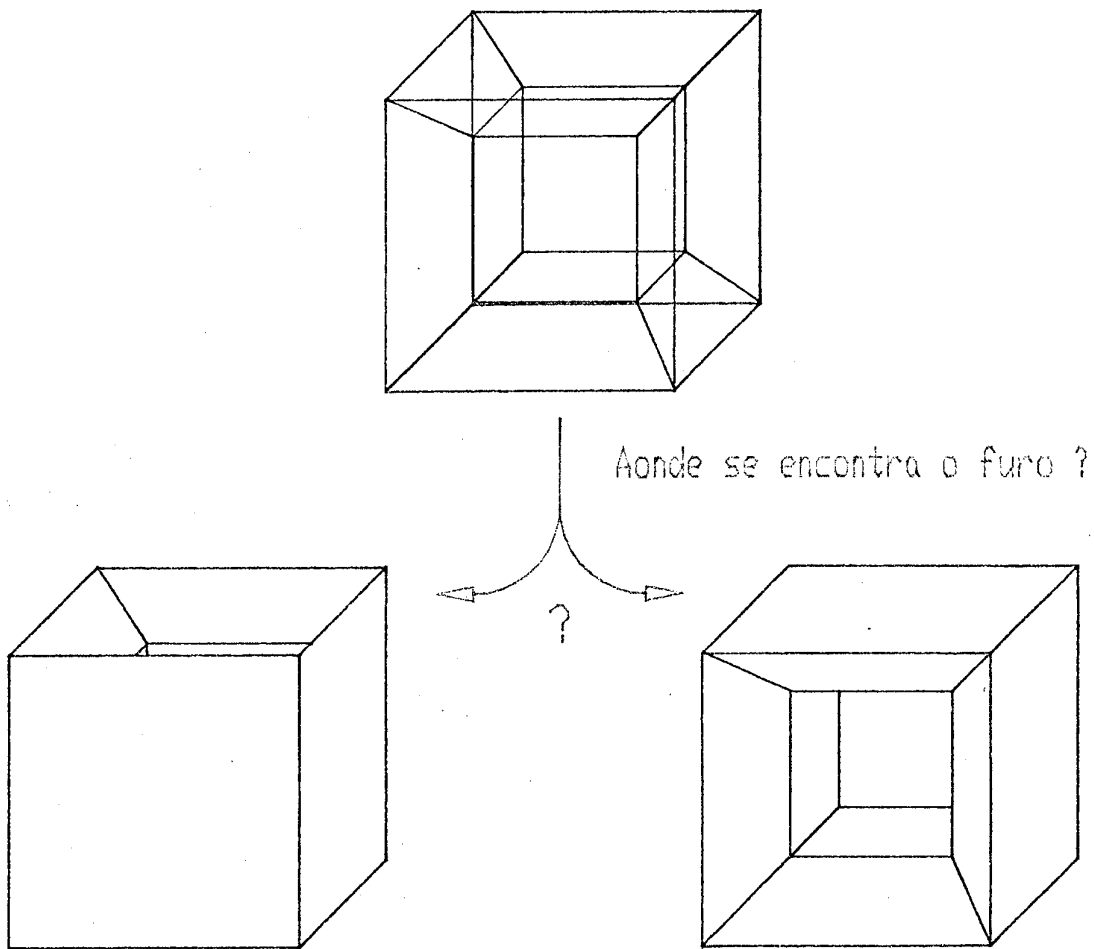


Figura 3.3 - Ambigüidade de uma representação por *Wire-Frame*

Além disso, esta será única, se os seus objetos não possuírem outra representação em M . Isso equivale a se exigir que:

$$S(S^{-1}(W)) = \{ W \} .$$

Uma forma de representação será completa e única, somente se todas as suas representações também o forem. Entretanto, não existe nenhuma implicação mútua entre as duas propriedades. Intuitivamente, pode-se afirmar que uma forma de representação é uma relação (função) entre sólidos abstratos e representações. Uma representação será inválida, se não corresponder a pelo menos um sólido válido, e será dita ambígua, se corresponder a vários. Além disso, constata-se que um sólido terá uma representação não única, se puder ser representado de diversas formas distintas [1,2].

3.4 Propriedades Formais e Informais

A partir das definições apresentadas, podem-se abstrair algumas propriedades importantes, relativas às formas de representação. Tais propriedades procuram oferecer critérios simples e práticos para a análise de formas de representação no contexto geral de um SMS e de suas aplicações (tabela 3.1).

Propriedades de Formas de Representação	
Domínio	<i>Quais objetos podem ser modelados ?</i>
Validade	<i>Quais possuem sentido físico ?</i>
Completa	<i>Quais podem ser manipulados ?</i>
Unicidade	<i>Quais são apenas cópias ?</i>
Concisão	<i>Quais repr. são práticas ?</i>
Eficiência	<i>Quais repr. são viáveis ?</i>
Consistência	<i>Quais repr. são contraditórias ?</i>

Tabela 3.1 - Propriedades e questões associadas

3.4.1 Domínio

O domínio de uma forma de representação caracteriza o seu poder em representar objetos. Essa propriedade explica, por si só, o fato de SMS's serem comumente (e erroneamente) analisados e comparados, em função da forma de representação que adotam. Sabe-se, por exemplo, que SMS's, baseados em uma forma de representação do tipo CSG, permitem facilmente a composição de objetos complexos a partir de sólidos primitivos, mas são, no entanto, bastante limitados no que diz respeito a modelagens locais. A razão maior dessa limitação não se refere especificamente ao SMS, mas sim à sua forma de representação.

3.4.2 Validade

O contradomínio de uma forma de representação especifica o conjunto de representações válidas que aquela pode assumir. Essa propriedade é de vital importância para que se possa garantir a integridade de uma estrutura de dados. Uma forma de representação válida garante, intrinsecamente, que toda a sua informação simbólica esteja sempre mapeando apenas objetos válidos. Um exemplo típico de objetos tridimensionais, que não obedecem a esse critério de validade, pode ser dado através dos conhecidos desenhos de *M.C.Escher* (figura 3.4).

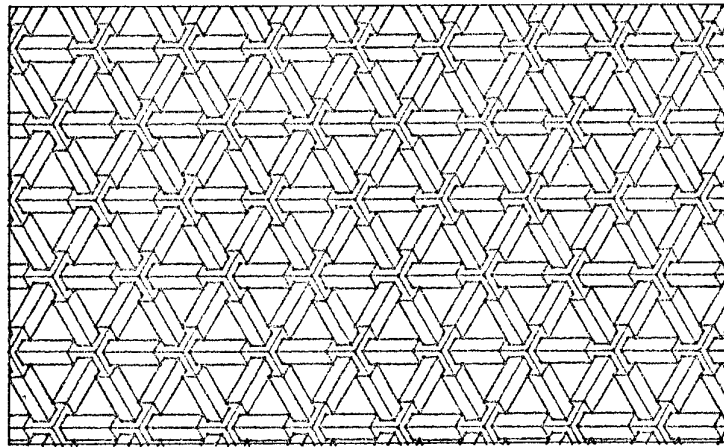


Figura 3.4 - Abstração ilusionista de objetos inválidos

Garantir a integridade de uma estrutura de dados não é uma tarefa fácil, se tal propriedade não for levada em consideração na sua fase de projeto. Observando-se, ainda, que algoritmos geométricos já são, por si só, altamente complexos, querer que estes garantam a validade dos objetos manipulados é praticamente impossível, tendo em vista o *overhead*, que tal fato iria acarretar. Na prática, essa exigência é simplesmente desconsiderada ou, então, transferida para o próprio usuário, causando, com frequência, o bloqueio da máquina ou, no pior caso, os estados, em que os resultados "parecem" consistentes, mas não o são realmente.

Independentemente da técnica usada para se obter a validade de uma estrutura simbólica de dados, a maneira mais limpa e segura é, sem dúvida alguma, a de se garantir que a forma de representação utilizada seja sintática e semanticamente correta, segundo os critérios já mencionados.

3.4.3 Completa

A propriedade de uma forma de representação poder ser dita completa ou não advém, principalmente, do tipo ou característica das aplicações que irão manipulá-la. Por definição, uma representação completa e não ambígua deveria ser capaz de conter toda a informação necessária, para que se pudesse distinguir quaisquer propriedades ou atributos

relacionados às suas entidades no domínio da modelagem, oferecendo recursos para que todo tipo de funções matemáticas, a elas pertinentes, fossem computadas.

Tal propriedade assume uma importância elevada quando o número de aplicações distintas, e conseqüentemente independentes, se torna grande. É comum observarem-se SMS's, que utilizam uma forma de representação central bem simples e compacta e deixam, a cargo das aplicações, a incumbência de geração de estruturas auxiliares, de modo a se suprir a falta de informação do núcleo através de dados relevantes apenas a elas. Esse procedimento tem se tornado, no entanto, cada vez mais desvantajoso em virtude dos seguintes pontos:

- (i) Os próprios SMS's têm necessitado, cada vez mais, de informações completas e precisas com relação aos objetos, não sendo mais viável a utilização de uma estrutura simples e robusta para satisfazer as novas e sofisticadas técnicas de modelagem;
- (ii) O *overhead*, associado à redundância de informação gerada pela duplicação de segmentos de estruturas, torna os procedimentos complexos e burocráticos;
- (iii) A geração de estruturas auxiliares exige uma constante atualização e verificação de consistência relativa ao modelo central de representação.

Formas de representação incompletas podem ser perfeitamente adequadas a algumas aplicações específicas. Uma análise global demonstra que apenas no contexto de SMS's, que objetivem uma grande variedade de aplicações, é que a propriedade de uma forma de representação ser completa ou não toma uma importância relevante. Serão vistas, mais adiante, formas de representação que foram desenvolvidas, objetivando-se uma certa classe restrita de aplicações.

3.4.4 Unicidade

A razão maior de se desejar uma forma de representação única está relacionada com a necessidade comum de se identificarem cópias de objetos ou de entidades dos mesmos. Dentro do contexto das aplicações e mesmo dos próprios SMS's, é freqüente a exigência de se determinar e, eventualmente, eliminar sólidos iguais. Sendo assim, se for possível garantir que uma forma de representação seja ao mesmo tempo não ambígua e única², pode-se concluir que objetos distintos possuirão representações distintas, reduzindo, portanto, a tarefa de identificação de igualdades a algoritmos que comparem apenas sintaticamente os objetos.

² Uma forma de representação não ambígua e única permite um mapeamento "um para um" entre objetos e representações.

Para se ter uma idéia da diferença entre uma comparação semântica com uma sintática, apresenta-se o seguinte exemplo de uma representação não ambígua e não única, que utilize conjuntos de pontos (3D) como forma de representação básica. Se se desejar testar dois conjuntos com relação à sua igualdade, deve-se testar, semanticamente, o conjunto que corresponde a sua diferença simétrica com o conjunto vazio.

Dentre as principais razões para a não unicidade de formas de representação, encontram-se as seguintes:

- (i) Dentro de uma mesma representação, é comum a permutação de sub-estruturas (*não unicidade por permutação*);
- (ii) Representações distintas correspondem, muitas vezes, ao mesmo objeto ou entidades congruentes, deslocadas apenas espacialmente (*não unicidade por posicionamento*).

Apesar de simples as razões para a não unicidade de uma representação, identificá-las pode ser bastante dispendioso computacionalmente para estruturas grandes, principalmente em se tratando da *não unicidade por posicionamento*, na qual complexos algoritmos geométricos são necessários.

3.4.5 Concisão e simplicidade

A propriedade de concisão diz respeito ao tamanho de que uma determinada representação necessita para representar os seus objetos. Uma representação concisa costuma ser facilmente transportável e de simples avaliação e manutenção, pois contém pouca informação redundante³.

Levando-se em consideração a tendência atual dos SMS's de fazerem uso cada vez mais constante de ferramentas geométricas e topológicas de alta sofisticação, necessitando de estruturas simbólicas poderosas para o seu suporte, prevê-se, conseqüentemente, que essa propriedade informal tende a ser um tanto quanto desprezada. Na verdade, essas novas técnicas de modelagem já têm sido desenvolvidas, considerando-se os recursos disponíveis na forma de representação adotada, modificando-se o enfoque, até então observado, para o desenvolvimento de SMS. Atualmente, a influência das necessidades, decorrentes de avançadas técnicas de modelagem, estão cada vez mais presentes na busca de uma "boa" forma de representação.

³ A não redundância de uma estrutura de representação implica, de uma forma geral, em que as suas entidades sejam pouco interdependentes, ocasionando, em consequência, o fato de poucas relações de restrição serem satisfeitas.

A simplicidade, com que representações válidas são geradas em SMS's, é de clara importância, principalmente por serem seres humanos os seus usuários. Representações concisas costumam ser mais fáceis de serem criadas, mas são, contudo, mais limitadas nas suas aplicações. Um dos objetivos maiores dentro da fase de projeto de uma estrutura de dados é encontrar um compromisso entre a complexidade da representação e o seu meio de aplicação (do ponto de vista das técnicas de modelagem).

3.4.6 Eficiência

O objetivo prático maior de um SMS é o de fornecer uma estrutura de dados capaz de possibilitar uma grande variedade de aplicações. Estas podem fazer uso de tal representação de diversas maneiras. Dependendo da estrutura utilizada, essa poderá ser bastante eficiente para algumas aplicações e pouco, para outras. Números naturais, por exemplo, quando representados através da simbologia romana, não são práticos para serem utilizados em manipulações numéricas, tendo no entanto, a sua serventia em outros contextos. Não obstante tal fato, a forma de representação utilizada deverá ser capaz de permitir a utilização de algoritmos confiáveis, isto é, algoritmos corretos, eficientes e robustos, na presença de erros numéricos ou de grande quantidade de dados.

3.5 Equivalência e Consistência em Representações Múltiplas

A experiência acumulada até o presente momento tem demonstrado que nenhuma forma de representação é uniformemente adequada a todas as aplicações ou mesmo às tarefas mais importantes, como a visualização tridimensional (*Rendering*). A técnica mais utilizada, para se aumentar a eficiência geral de SMS, tem sido a de se inserir informação redundante (normalmente sobre a forma de múltiplas representações), direcionada para uma determinada classe de aplicações.

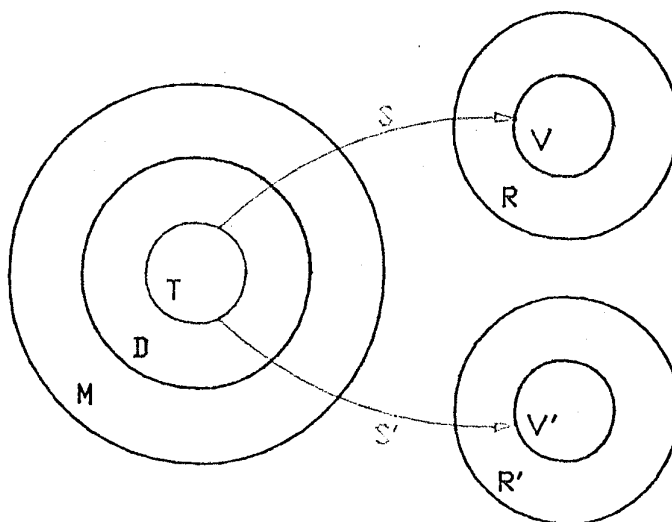


Figura 3.5 - Consistência de múltiplas representações

Através dessa técnica, pode-se verificar mais de uma forma de representação a mapear um mesmo objeto, introduzindo uma forte redundância em SMS. Sendo assim, há de se garantir que, a qualquer estágio da modelagem, as estruturas de dados estejam num estado consistente, isto é, que formas diferentes de representação não possuam nenhum tipo de informação contraditória entre si. Formalmente, pode-se afirmar que as formas de representação $V1$ em R e $V2$ em R' são consistentes, se existir ao menos um objeto T em M , mapeável através das formas de representação S e S' escolhidas na elaboração do sistema (figura 3.5).

Como já foi visto, consistência não implica, necessariamente, em uma relação de equivalência entre formas de representação. Formalmente, diz-se que duas formas de representação S e S' são equivalentes entre si, se estas se associam através do mesmo conjunto de objetos, isto é, se:

$$S^{-1}(R) = S'^{-1}(R') .$$

Em outras palavras, é possível se afirmar que duas representações R e R' são equivalentes, se, para cada representação de um, encontramos um equivalente na outra, e vice-versa. Observe-se que tal definição obriga ambas as representações a possuírem o mesmo domínio, sendo, então, consideradas como equivalentes e não ambíguas.

Todavia, dentro do contexto específico de SMS's com múltiplas representações, é comum o desprezo ao critério de equivalência. O importante, por outro lado, é o fato de se assegurar a consistência geral do sistema, sem, com isso, comprometer a sua estrutura e finalidade maior, que é a de possibilitar uma representação completa e confiável para objetos tridimensionais.

PARTE II

FORMAS DE REPRESENTAÇÃO

PARA

OBJETOS SÓLIDOS

Com base nos critérios e conceitos estabelecidos nos capítulos anteriores, faz-se, nessa parte, uma abordagem completa das formas de representação mais conhecidas, bem como uma rápida análise de formas híbridas de representação.

CAPÍTULO 4

FORMAS NÃO AVALIADAS DE REPRESENTAÇÃO

4.1 - Introdução

Pelo pouco que o arcabouço de critérios e propriedades, discutidos no capítulo anterior, sugere, não é simples a tarefa de se descobrir uma nova forma de representação. A história tem demonstrado que as boas soluções estão sempre condicionadas a ciclos de estudos e tentativas,

FORMAS DE REPRESENTAÇÃO	Sigla	Avaliada
Formas Ambíguas	<i>AMR</i>	<i>não</i>
Instâncias Puramente Primitivas	<i>PPI</i>	<i>não</i>
Geometria Sólida Construtiva	<i>CSG</i>	<i>não</i>
Representação por Deslocamento	<i>SWEEP</i>	<i>não</i>
Representação Analítica	<i>AREP</i>	<i>não</i>
Enumeração por Ocupação Espacial	<i>SOE</i>	<i>sim</i>
Decomposição em Células Espaciais	<i>SCD</i>	<i>sim</i>
Representação por Fronteira	<i>BREP</i>	<i>sim</i>

Tabela 4.1 - As representações mais conhecidas

resultando em pequenos avanços. A procura de uma solução para o problema aqui apresentado não se apresenta como uma exceção; muito pelo contrário, várias propostas têm sido elaboradas nos últimos anos.

Dentre as formas de representação mais conhecidas, destacam-se algumas que serão analisadas em detalhes durante as próximas seções e capítulos (tabela 4.1). Essas formas de representação podem ser classificadas segundo um processo conhecido como "Avaliação de Fronteira" (*Boundary Evaluation - BEV*), realizado constantemente em SMS's. O processo de avaliar a fronteira de um sólido consiste, resumidamente, em descrevê-lo em termos da sua fronteira com o espaço, no qual este se encontra inserido. Tal procedimento é de fundamental importância para a viabilização de uma série de técnicas para a modelagem de sólidos. Já o termo isolado "avaliação" é geralmente utilizado, para especificar o processo de se extraírem propriedades geométricas de uma determinada representação.

Como já foi dito anteriormente, nenhuma dessas propostas foi capaz de preencher a todos os requisitos exigidos, muito menos de representar todos os objetos encontrados na natureza. Não obstante tal fato, tais tentativas podem ser bastante úteis para a elaboração de novas representações.

Analisando-se, de uma forma bem abstrata, o problema, poder-se-ia afirmar que tais representações serviriam como "cercas individuais", delimitando o conjunto dos objetos efetivamente representáveis. A composição dessas "cercas" poderia dar origem a uma forte e poderosa "fronteira", englobando, possivelmente, todas as aplicações práticas, baseadas em uma estrutura simbólica para a representação de sólidos tridimensionais¹.

4.2 Formas Ambíguas

Sendo a Engenharia uma ciência aplicada, a comunicação de fatos físicos deve ser completa e precisa. Relações quantitativas são expressas matematicamente. A palavra escrita completa muitas descrições, mas, sempre que máquinas e estruturas são projetadas e construídas, a representação gráfica é necessária. Embora as pinturas de artistas (ou fotografia e outros métodos de reprodução) ofereçam representação em perspectiva, não podem servir como descrições formais para uma representação não ambígua de objetos. Traçados em perspectiva, sombreados e fotografias são usados para determinados fins, mas a grande maioria dos desenhos de Engenharia é feita apenas com linhas, com vistas diferentes,

¹ Na verdade, existem teorias ainda não comprovadas, que proclamam que, se os critérios e propriedades mencionados no capítulo anterior forem todos satisfeitos, a representação utilizada seria, por conseguinte, capaz de descrever todos os objetos físicos encontrados na natureza.

dispostas em um sistema lógico de projeção. A estas vistas, acrescentam-se dimensões e lembretes especiais, dando direções e outras instruções (figura 4.1). Esta é a linguagem da expressão gráfica, que pode ser definida como a representação gráfica de objetos sólidos e suas relações.

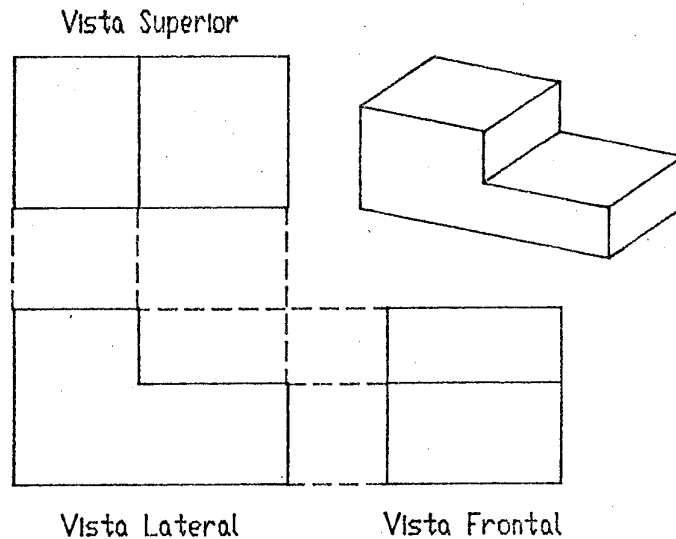


Figura 4.1 - Representação gráfica através de projeções

Essa forma de representação é melhor vista como uma maneira informal de comunicação entre os seres humanos, pois estes se utilizam, livremente, de um senso comum para interpretar os desenhos. Comumente, ocorrem erros de interpretação, e, vez por outra, desenhos errados são interpretados corretamente, sem que seja notada a existência

daqueles. Tal fato ocorre, exatamente por não haver nenhuma definição formal do desenho técnico como uma forma de representação. As entidades, que podem ser observadas nas vistas, não são definidas de uma forma precisa. Um exemplo clássico pode ser constatado no *Cubo de Necker*, para o qual, dependendo da forma com a qual o observador está mais familiarizado, este gera abstrações ambíguas do objeto real (figura 4.2).

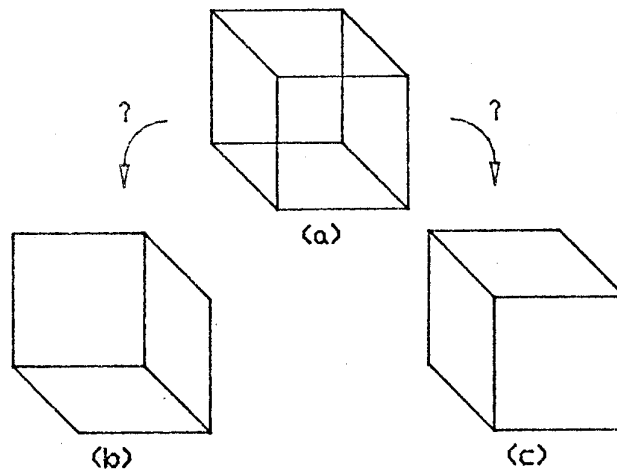


Figura 4.2 - (a) *Cubo de Necker*,

(b) e (c) abstrações possíveis

Uma pesquisa, realizada com os principais textos sobre desenho técnico, mostra que a restrição, com relação aos desenhos, é muito fraca. Consta apenas, que entidades

desenhadas através de vistas, vistas auxiliares, detalhes e notas paralelas devem ser fornecido em uma quantidade suficiente, de forma a evitar ambigüidades. Conclui-se que definições literais são muito informais para serem tomadas como critérios para a especificação de uma forma de representação.

Não obstante os problemas de representação expostos acima, desenhos são uma forma bastante eficiente de comunicação entre os homens e têm sido, até o presente momento, a principal forma de mídia para a especificação geométrica utilizada pela indústria. Os primeiros GMS's representavam, na verdade, entidades bidimensionais ao invés de objetos em três dimensões. Nesses sistemas, modificações feitas a uma vista de um objeto não podiam se propagar dentre as outras ali disponíveis, pois o sistema desconhecia o fato de todas as vistas estarem se referindo a um mesmo objeto, visto de diferentes posições.

Uma tentativa de formalizar a representação de objetos através do desenho técnico é proposta por *Arestides A.G Requicha* [39] da seguinte forma:

- (i) Objetos sólidos seriam representados por uma coleção de projeções planas. A fim de se construir uma forma não ambígua de representação a partir de tal noção, deveria ser fornecido, dentro do seu domínio, um preciso mapeamento entre objetos e desenhos. Definir

esse mapeamento é bastante difícil e pode ser comparado com o problema, ainda não resolvido, de se abstrair uma informação sobre a composição da forma de um objeto em três dimensões, a partir de suas projeções em duas dimensões;

- (ii) Abandonar a utilização de projeções e procurar uma coleção adequada de entidades 3D para a representação. Nesse caso, poder-se-iam utilizar *Arestas* como entidade primária dessa representação. Tal fato levaria a uma representação do tipo *Wire-Frame*, que também conduziria a uma representação ambígua, mesmo para o domínio dos objetos de faces planas.

Observa-se, ainda, uma outra forma de representação ambígua (*Ambiguous Representations - AMR*) que, como pode ser constatado a partir da sua própria definição, também conduz a várias inconsistências. Essa representação se origina do problema prático de se encontrar uma forma para representar um determinado objeto, dentro de uma máquina computacional. A técnica consiste em se coletar um número suficiente de medidas relevantes do objeto em um conjunto, definindo-o geometricamente.

Como exemplo, pode-se considerar um conjunto de pontos pertencentes à carroceria de um automóvel. Observa-se

claramente que essa coleção de pontos forma uma representação sem qualquer relação com a estrutura topológica do modelo, mesmo que a resolução dos pontos selecionados seja elevada, pois o problema reside, puramente, numa falta de informação topológica associada ao modelo e não na sua geometria. Além disso, existe a ambigüidade gerada pela não especificação da forma, pela qual tais pontos seriam considerados como relevantes à representação do objeto. Na verdade, essa informação pendente deveria estar encoberta pelos algoritmos que a manipulam, tornando-os não apenas dependentes da estrutura interna de representação, como também do tipo de objeto a ser representado.

4.3 Instâncias Puramente Primitivas

A representação de objetos, que visam exclusivamente o processo de manufatura, foi uma das primeiras justificativas para a utilização de uma forma de representação. Uma técnica bem original foi adotada em 1973 por ingleses [2,11], através de Instâncias Puramente Primitivas (*Pure Primitive Instancing - PPI*). Essa forma de representação se baseia na noção de famílias de objetos, subdivididas em membros (figura 4.3). Uma família é composta de membros que se distinguem através do valor de alguns parâmetros que os definem. Cada objeto-família é chamado de uma primitiva genérica, e membros individuais são chamados de instâncias primitivas.

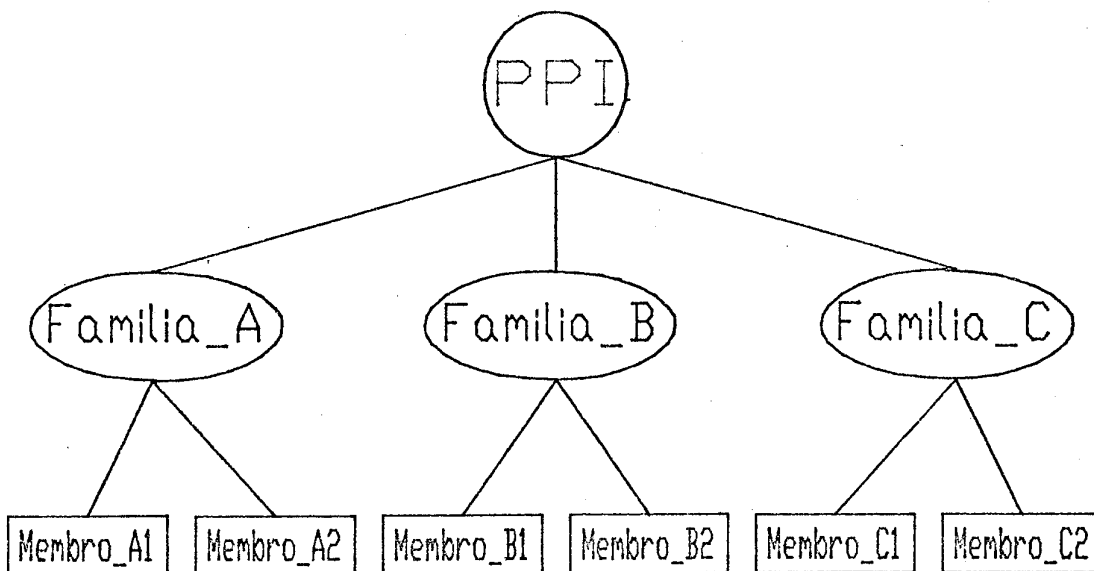


Figura 4.3 - Instâncias Puramente Primitivas

Instâncias primitivas são representadas internamente por uma seqüência de parâmetros, que identificam a primitiva genérica (família) a qual aquelas irão pertencer, e suas características (normalmente, atributos de dimensão). Como exemplo, cita-se uma primitiva genérica *Bloco_Tipo_L*, que poderia ser representada por uma instância primitiva da seguinte forma:

("Bloco_Tipo_L", L, Hb, Ht, Ct, Cb) ,

onde "Bloco_Tipo_L" é um conjunto de caracteres, identificando a família; "L", "Hb", "Ht", "Ct" e "Cb", são parâmetros de dimensão (figura 4.4).

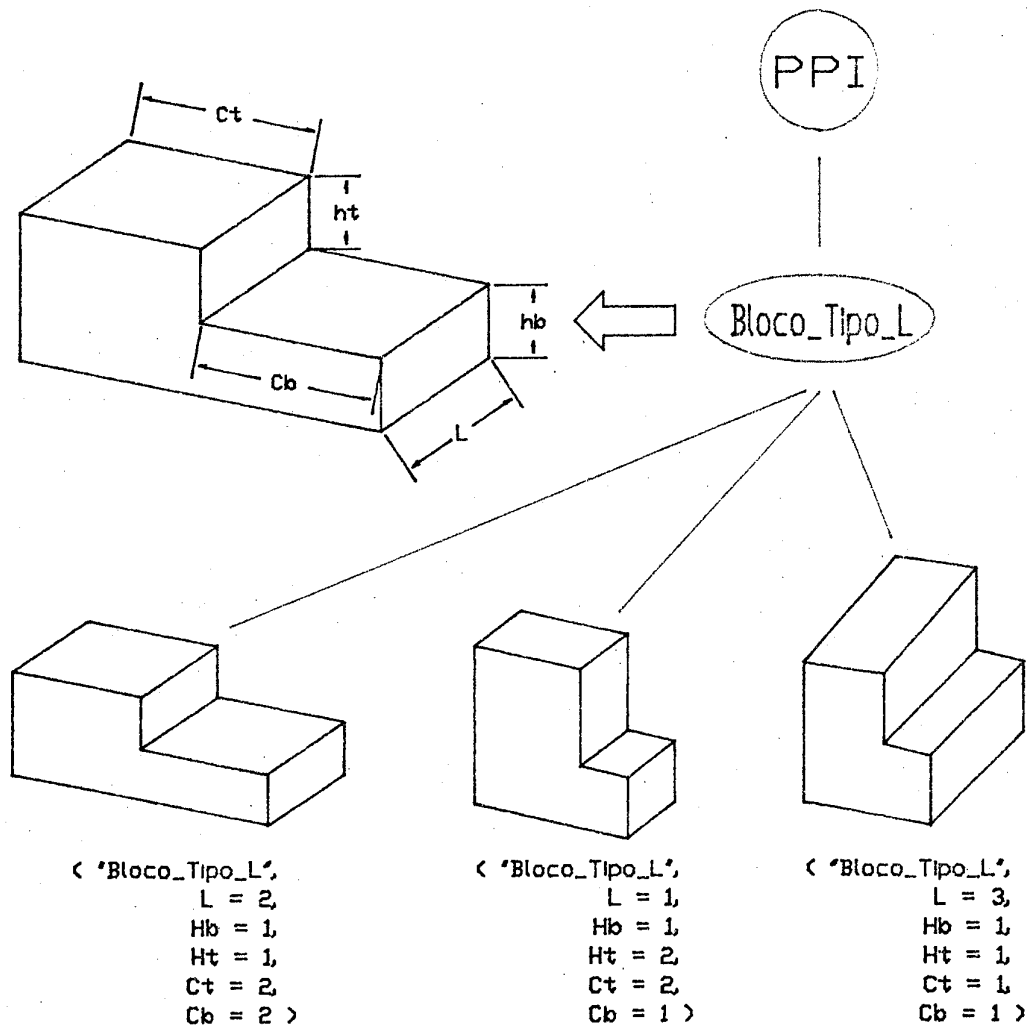


Figura 4.4 - Exemplo de Instância Puramente Primitiva

É importante notar a partir do exemplo acima, que apenas os cinco últimos parâmetros foram suficientes para instanciar completamente a primitiva genérica dentro da sua família "*Bloco_Tipo_L*". Dependendo da complexidade da primitiva genérica, mais parâmetros poderiam ser necessários de modo a permitir a sua completa instanciação.

Bastante marcante em tal forma de representação é a impossibilidade de se combinarem instâncias, visando à criação de estruturas que representem objetos mais complexos ou ainda não definidos na biblioteca-padrão do sistema. *Arestides A.G. Requicha* [39] faz uma comparação bastante sutil, quando compara um sistema, que utiliza essa forma de representação, a uma linguagem definida a partir de uma gramática, na qual não é possível combinarem-se palavras para se formarem frases ou sentenças.

Em princípio, Instâncias Puramente Primitivas não são ambíguas, são únicas, facilmente validáveis, concisas e simples de serem utilizadas, além, é claro, de promoverem a padronização de componentes e objetos. Na prática, entretanto, tais vantagens são pouco notadas, tendo em vista o restrito domínio, sobre o qual uma representação desse tipo pode ser utilizada, a não ser que uma vasta biblioteca de famílias esteja disponível, com cada instância sendo caracterizada por um número reduzido de parâmetros.

Nota-se, aqui, a forte relação dessa forma de representação com o processo de manufatura, onde apenas alguns objetos com características bem marcantes são de interesse específico a uma determinada aplicação. Em tais circunstâncias, essa forma de representação se justificaria em virtude da sua restrita aplicação.

Uma desvantagem bastante marcante dessa representação recai sobre os algoritmos que a manipulam. Estes são, quase sempre, fortemente dependentes das primitivas genéricas que estão disponíveis, visto que cada família é tratada como um caso particular, resultando, conseqüentemente, em algoritmos lentos, grandes e de pouca eficiência. Conclui-se, portanto, não ser possível um tratamento uniforme para diferentes primitivas genéricas.

A principal virtude de uma representação do tipo PPI recai sobre a entidade família, que agrega em si uma forma estruturada e hierárquica atribuída aos objetos nela representados. Essa característica, no entanto, tem sido desvalorizada em função do fato de outras formas de representação terem surgido (o que será abordado posteriormente), nas quais as instâncias não são parametrizadas, e as famílias não são restritas, mas onde a noção de hierarquia também está presente.

4.4 Geometria Sólida Construtiva

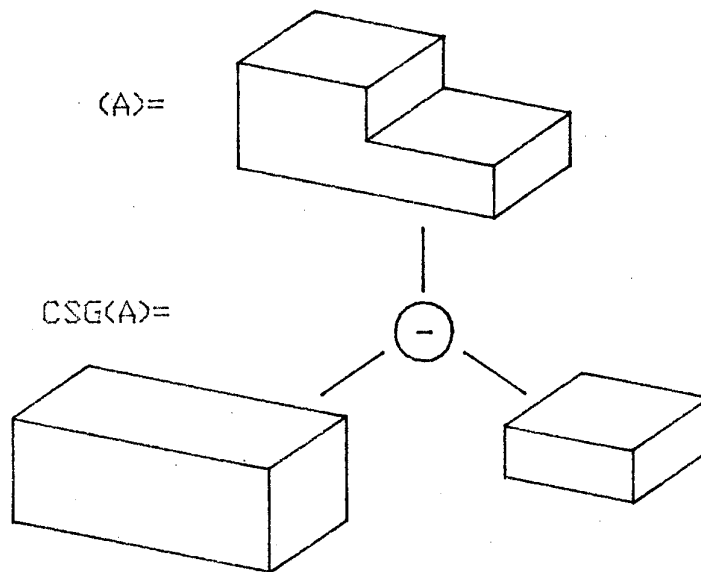


Figura 4.5 - Modelagem utilizando CSG

Uma forma de representação bastante intuitiva e elegante para objetos tridimensionais pôde ser alcançada a partir da técnica conhecida como Geometria Sólida Construtiva (*Constructive Solid Geometry - CSG*). Tal forma de representação foi idealizada, tendo como base a simples observação de como um artesão ou um torneiro modelam os seus objetos. Estes fazem uso da velha técnica de "dividir para conquistar", no sentido de que ao manufaturarem os seus objetos, utilizam apenas operações simples, como a extração de um volume do sólido inicial ou,

ainda, a adição, colando ou fundindo sólidos individualmente. Observaram-se, então, duas coisas: primeiro, o fato dos volumes retirados também serem sólidos, e, em segundo, que as operações utilizadas eram, na verdade, operações booleanas, entre as quais inclui-se a operação de interseção de sólidos, não observada até então.

Feita uma analogia com o processo de modelagem descrito acima, define-se a forma de representação CSG como aquela, na qual sólidos são representados a partir de composições booleanas de sólidos primitivos (figura 4.5).

4.4.1 Álgebra Booleana

As operações booleanas mencionadas estão inseridas dentro do contexto de uma álgebra booleana definida conforme explicitado a seguir.

Seja W um conjunto (o universo) e T uma topologia sobre W , isto é, uma coleção de todos os subconjuntos de W . No espaço topológico (W, T) , um subconjunto X de W é dito um conjunto regular fechado, se este é fechado com relação ao seu interior, isto é:

$$X = \text{cl}X,$$

onde k e i denotam o seu fechamento e o seu interior respectivamente [8]. O conjunto regular fechado, composto pela união (\cup), interseção (\cap), diferença ($-$) e pelo complemento (C) de dois subconjuntos X e Y de W é definido da seguinte forma:

$$X \cup Y = ki (X \cup Y);$$

$$X \cap Y = ki (X \cap Y);$$

$$X - Y = ki (X - Y);$$

$$C X = ki (C X);$$

onde C representa o complemento usual com relação a W . Define-se, portanto, uma álgebra booleana a partir dos conjuntos regulares de W , adicionados ao conjunto dos operadores regulares definidos acima [43].

4.4.2 Árvores CSG

A implementação prática de uma representação do tipo CSG é alcançada através da utilização de uma árvore binária ordenada. Nessa topologia, definem-se os nós não terminais como operadores que, por sua vez, podem ser classificados como movimentos rígidos ou operações regulares, como união, interseção ou diferença. Os nós terminais dessa árvore (folhas) serão, ou sólidos primitivos representando subconjuntos do E^3 (*Conjunto-R*), ou argumentos de transformações de movimentos rígidos (figura 4.6) [38].

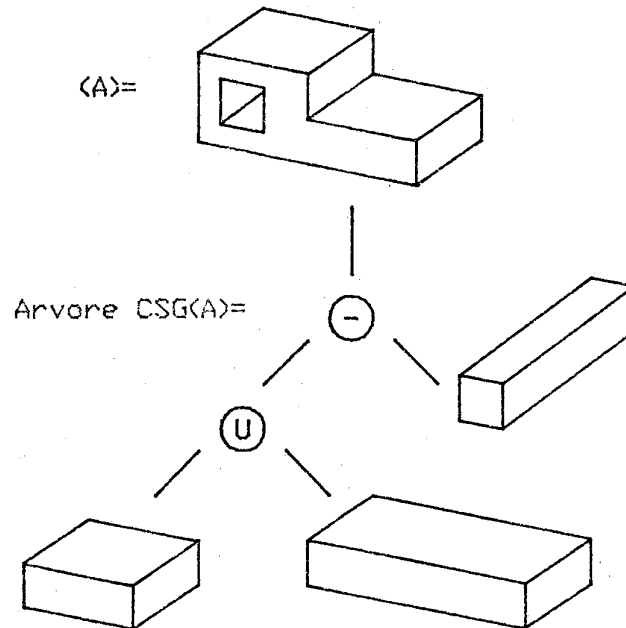


Figura 4.6 - Exemplo de uma árvore do tipo CSG

Formalmente, pode-se definir um árvore do tipo CSG, com o auxílio da linguagem BNF (*Backus-Naur Form*) da seguinte maneira;

```

<Árvore CSG> ::= <Primitiva> |
               <Árvore CSG> <Operador> <Árvore CSG> |
               <Árvore CSG> <Transformação> <Argumentos>
<Primitiva>  ::= Sólido_Válido
<Operador>   ::=  $\cap$  |  $U$  |  $-$ 
<Transformação> ::= Escala | Translação | Rotação
<Argumentos> ::= Número_Real { Número_Real }

```

Em muitas implementações, é permitido o compartilhamento de sub-árvores para objetos distintos, resultando, portanto, em uma representação não mais em árvore, mas sim em um grafo dirigido. Além de tal detalhe, existem algumas ambiguidades associadas às primitivas mencionadas acima. Normalmente, as primitivas utilizadas devem ser fechadas e únicas, como, por exemplo, blocos, cilindros, esferas, etc., ou seja, sólidos regulares válidos. Entretanto, existem algumas implementações que permitem também a existência de primitivas não fechadas, dando origem a árvores baseadas em Sub-Espaços (*Half-Spaces - HSP*).

Árvores CSG, baseadas em HSP, têm sido bastante utilizadas, oferecendo uma maior flexibilidade na representação de primitivas básicas. Qualquer superfície "Bem-Formada" (*Well-Formed - WFORM¹*), capaz de dividir o espaço tridimensional em duas regiões disjuntas, pode ser definida como um Sub-Espaço. Exemplificando (figura 4.7), tem-se que o plano, definido pela equação:

$$\alpha : x = 0,$$

divide o espaço tridimensional em dois Sub-Espaços, a saber:

¹ Critérios precisos para a classificação de superfícies "Bem-Formadas", serão abordados mais adiante (no estudo da Representação por Fronteira).

$$\beta : x < 0 \text{ e}$$

$$\gamma : x > 0 .$$

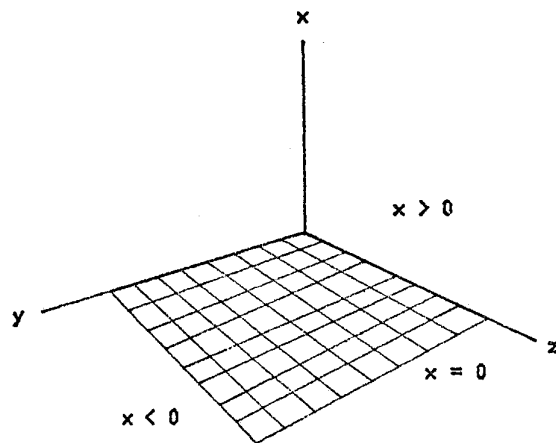


Figura 4.7 - Divisão do espaço por um HSP

Similarmente, tem-se que o cilindro:

$$\alpha : x^2 + y^2 = 1$$

é definido pelos Sub-Espaços:

$$\beta : x^2 + y^2 < 1 \text{ e}$$

$$\gamma : x^2 + y^2 > 1 .$$

Observa-se, ainda, que a noção de Sub-Espaços não impõe que estes sejam finitos em extensão, como demonstra o

Sub-Espaço que descreve uma esfera do tipo:

$$\alpha : x^2 + y^2 + z^2 = 1 .$$

4.4.3 Propriedades

Esquemas de representação do tipo CSG não são ambíguos, mas também não são únicos. O domínio dessa representação está diretamente vinculado (dependente) ao conjunto de sub-espacos (representáveis), encobertos pelas suas primitivas básicas, e através dos tipos de transformações e operadores disponíveis.

Uma característica bastante valorizada de uma árvore do tipo CSG é o fato de, ao se garantir que os sólidos primitivos utilizados sejam realmente elementos de um *Conjunto-R*, se poder conclusivamente deduzir que qualquer árvore CSG é uma representação válida para o mesmo. Para tanto, deve-se exigir que o operador regular de complemento (C) não seja utilizado, pois este pode facilmente destruir a exigência de os sólidos serem fechados (imposição feita na definição de um sólido válido). Além disso, pode-se garantir, também, que se, em uma linguagem CSG, um objeto for representado sintaticamente de uma forma correta, este poderá ou não ter os seus parâmetros instanciados, mantendo, ainda assim, uma representação não ambígua. Tal fato abre um campo de aplicação

bastante vasto para SMS's do tipo paramétricos ou para aqueles orientados a macros ou procedimentos, que se utilizem de uma representação do tipo CSG.

Pela abordagem anterior, pode-se observar a importância de que se estabeleçam critérios precisos para a determinação do fato, de uma primitiva sólida ser ou não fechada com relação ao seu espaço de representação. Essa prova é de difícil avaliação, existindo, apenas, alguns procedimentos conhecidos (e computacionalmente caros), que partem de uma re-representação do objeto inicial, via uma representação por fronteira, testando, então, faces individualmente. Além disso, deve-se garantir, também, que os operadores utilizados façam parte daqueles definidos como regulares.

Um ponto bastante positivo dessa representação decorre da facilidade comprovada, com a qual os seres humanos manipulam as primitivas, a fim de modelarem os objetos. Pesquisas, realizadas com sistemas que oferecem essas técnicas, têm demonstrado, entretanto, que os usuários conseguem modelar rapidamente apenas uma aproximação dos objetos desejados. A razão para tal fato advém, simplesmente, do pouco suporte oferecido pelos sistemas para a realização de operações locais sobre os objetos. Fora isso, existem sérios problemas associados à tarefa de se posicionarem os objetos tridimensionalmente, antes que operações booleanas sejam

invocadas pelos usuários. Resultados pouco formais proclamam que aproximadamente 65% de todos os objetos encontrados na natureza podem, de alguma forma, ser modelados em sistemas que utilizem uma representação do tipo CSG.

São conhecidas várias formas restritas de representações, baseadas no modelo CSG. Algumas delas serão vistas, como a SOE e a SCD, nas quais apenas um operador do tipo união, mais conhecido como um operador de junção, é utilizado. Tal operador é, entretanto, aplicável apenas para conjuntos formados por sólidos de interiores disjuntos.

4.5 Representação por Deslocamento

A Representação por Deslocamento (*Sweep Representation - SWEEP*) oferece mais uma opção para a representação da forma de objetos tridimensionais. Essa técnica consiste, basicamente, no princípio de que, ao se escolher um conjunto de entidades tridimensionais (ou bidimensionais) e se estas forem deslocadas no espaço sobre uma trajetória pré-determinada, tais entidades irão "cortar" um volume do espaço. Esse volume de espaço é então associado ao volume de um sólido correspondente. Em outras palavras, pode-se considerar que, em uma representação do tipo SWEEP, um sólido é definido a partir de um conjunto de entidades 3D ou 2D e por uma trajetória associada [2,22].

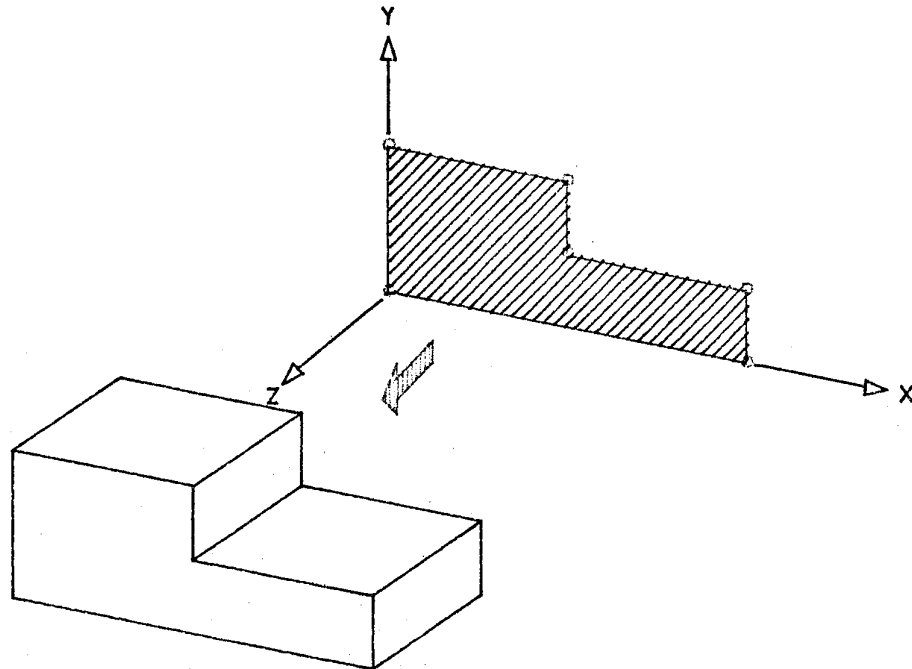


Figura 4.8 - Sólido representado por uma técnica de SWEEP

Dentre as formas de representação conhecidas, pode-se afirmar que, apesar da SWEEP ter sido seguramente uma das primeiras representações propostas, esta é, até o momento, a mais desconhecida no que se refere aos seus critérios de validade. Apesar disso, essa técnica (ou suas variantes) tem sido utilizada largamente em SMS, devido à simplicidade com que sólidos, que possuem algum tipo de simetria, podem ser representados. Como exemplo, podem-se citar os sólidos que são manufaturados em tornos mecânicos circulares. Para que esses objetos sejam modelados em um SMS baseado em uma representação

do tipo **SWEEP**, o usuário deve apenas editar em 2D o seu contorno e especificar o eixo de rotação. Dessa forma, o sistema é capaz de calcular a trajetória que esta entidade 2D realizará no espaço (supondo uma rotação completa de 360°) e gerará um sólido, a partir do volume circunscrito à mesma. A operação de **SWEEP** por translação pode ser realizada de maneira análoga (figura 4.8).

Sistemas que utilizam uma representação por **SWEEP** oferecem uma razoável comodidade aos algoritmos de modelagem e, indiretamente, aos usuários que os manipulam, mas sofrem sérios problemas de consistência e validade das suas representações. A razão para isto é bastante simples; não existe, *a priori*, nenhuma restrição com relação ao conjunto de entidades (3D ou 2D) que podem ser usadas e nem às suas trajetórias. Com isso, permite-se que representações de sólidos não validáveis (não contidos no *Conjunto-R*) sejam facilmente gerados (figura 4.9).

A saída, adotada pela maioria dos sistemas baseados em **SWEEP**, é a de simplesmente limitar o conjunto de entidades e trajetórias disponíveis. Um conjunto de entidades normalmente aceitas são as de poligonais fechadas e não convexas (2D). Dentre as trajetórias normalmente disponíveis estão as lineares (deslocamento por translação) e as rotacionais (deslocamento por rotação).

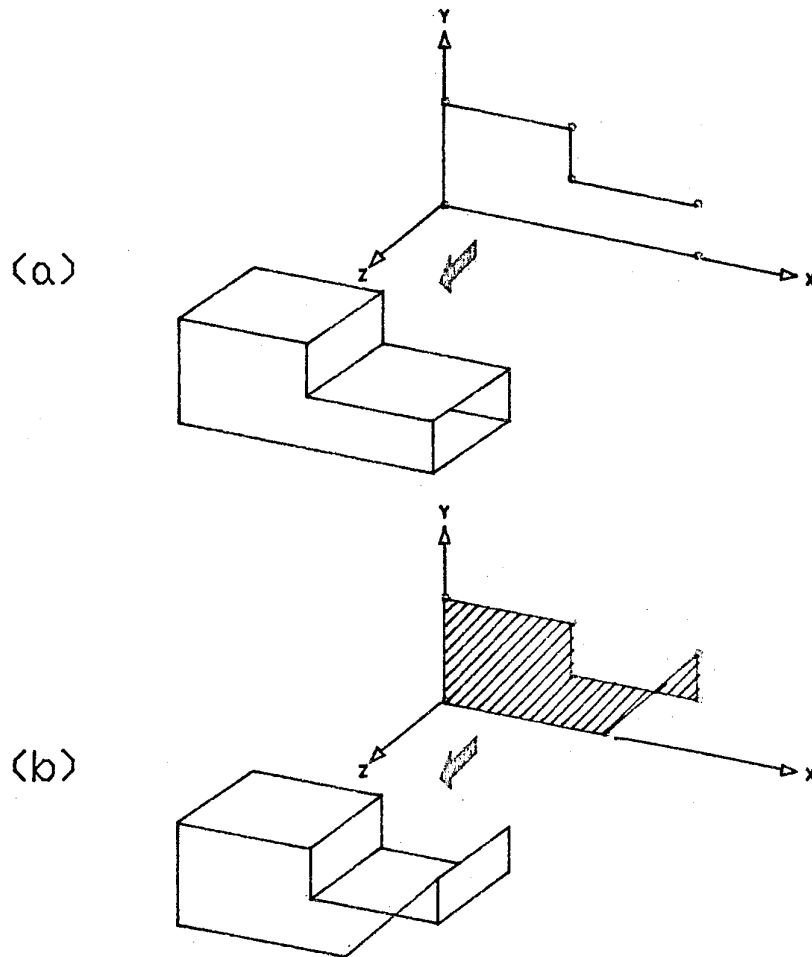


Figura 4.9 - Objetos inválidos gerados através de SWEEP

4.5.1 Definição Matemática

Apenas com o objetivo de ilustração, considera-se um conjunto de entidades bidimensionais A , localizadas no plano XY , e um segmento de reta B , paralelo ao eixo Z , com início na

origem dos eixos (figura 4.10).

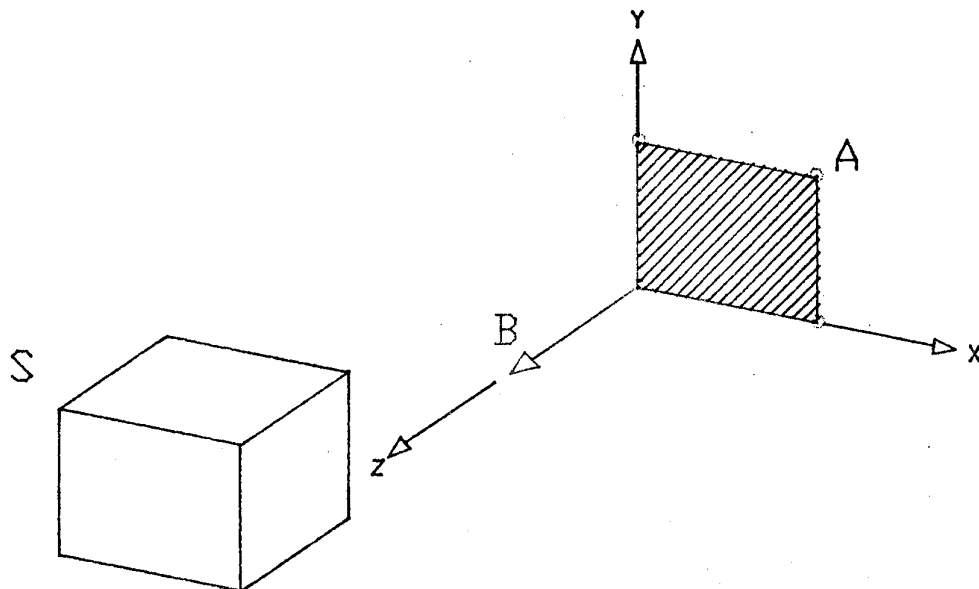


Figura 4.10 - SWEEP por um deslocamento de translação

Seja S o sólido abstrato gerado a partir do deslocamento de A (conjunto 2D) por B (trajetória 3D). Matematicamente, afirma-se que S pode ser definido como o produto cartesiano de A por B , ou seja:

$$S = A \times B .$$

Extendendo a definição matemática exposta acima, podem-se projetar representações gerais do tipo SWEEP,

selecionando-se funções específicas que mapeem conjuntos de produtos escalares no E^3 . Matematicamente, constatar-se-ia, então, que um conjunto S seria representado por um par de conjuntos (A, B), de tal forma que:

$$S = f (A \times B) ,$$

onde f é uma função definida da seguinte forma:

$$f : A \times B \rightarrow E^3$$

4.5.2 Generalizações

O exemplo exposto acima obriga um sistema de modelagem a representar apenas os conjuntos A e B, ambas entidades bidimensionais, reduzindo e simplificando, portanto, vários processos dentro de SMS's. Além disso, essa representação será, conseqüentemente, bastante concisa (compacta) e de fácil manutenção.

Fazendo-se menção aos critérios definidos no capítulo anterior, pode-se qualificar a forma de representação SWEEP como não ambígua e não única. O domínio dessa representação está limitado fortemente aos objetos (sólidos válidos) que possuam simetrias lineares e axiais.

Em alguns casos, encontra-se SMS's que utilizam representações por SWEEP, para permitirem a modelagem de objetos tipo "placa", ou seja, sólidos que possuem espessura constante e normalmente desprezível em função de outras medidas. Nesse contexto, a representação do tipo SWEEP é comumente acompanhada de uma representação do tipo BREP, caracterizando o que se conhece por representações $2\frac{1}{2}$ D.

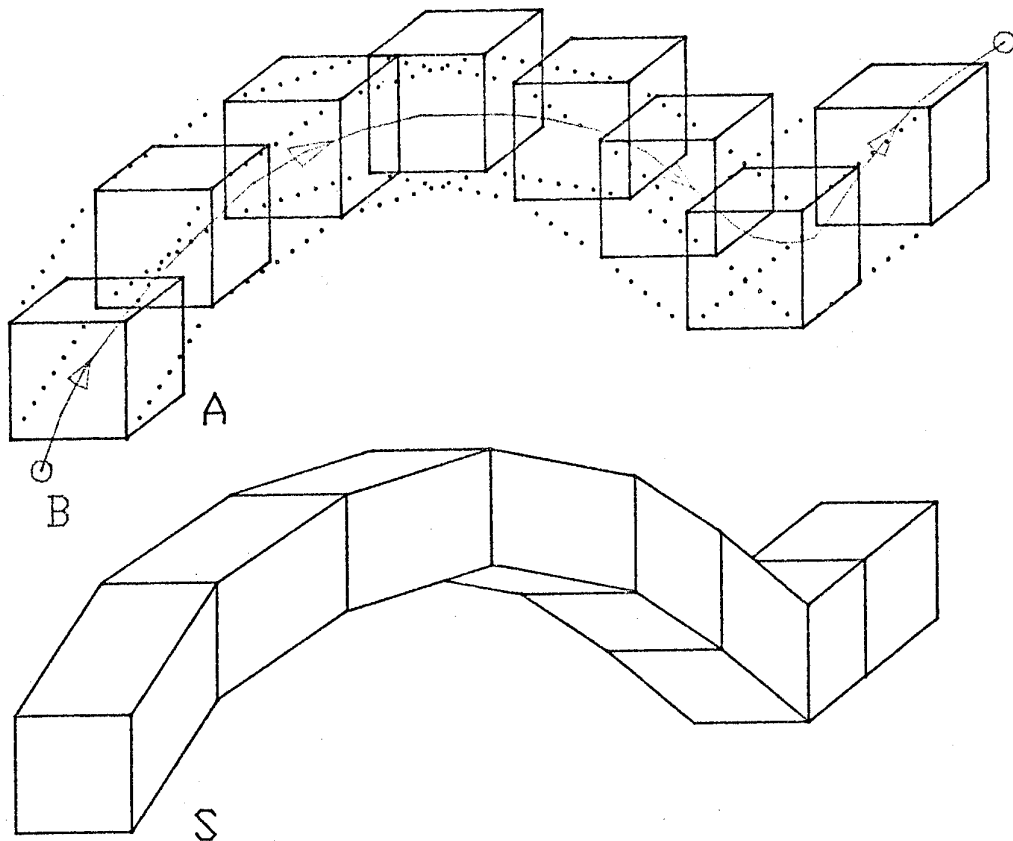


Figura 4.11 - SWEEP por deslocamento de sólidos

Ao se ampliar o domínio das entidades de deslocamento, englobando sólidos tridimensionais (entidades 3D), observar-se-ia uma variante da representação por SWEEP, denominada "deslocamento por movimento de sólidos" (figura 4.11). O conjunto A seria, agora, composto de entidades 3D (sólidos) [23,24].

No contexto de aplicações voltadas para a automação da manufatura, esse tipo de representação pode ser bastante explorado. Tal variante permite, por exemplo, que problemas como a retirada de material de um bloco sendo torneado ou ainda testes dinâmicos de interferência possam ser eficientemente suportados. Aplicações em robótica também podem ser aceleradas e simplificadas. Essas razões têm sido frequentemente utilizadas como justificativas para uma padronização internacional de uma forma de representação do tipo SWEEP.

Finalizando, observam-se, ainda, algumas propostas [*] de modeladores, baseados em uma outra extensão da técnica de SWEEP por deslocamento de sólidos, na qual elimina-se a exigência imposta aos sólidos válidos de serem rígidos. Em tal abordagem, um sólido, ao se deslocar pelo espaço, poderia ter a sua forma alterada em alguns dos seus parâmetros de definição. Como exemplo, mencionam-se sólidos cilíndricos que, ao se deslocarem espacialmente, variariam o seu raio, deformando, portanto, o espaço de uma maneira não regular e uniforme.

Modeladores desse tipo teriam aplicações práticas na modelagem de complexos industriais químicos, aerofólios de aviões, sistemas elétricos, estruturas em construção civil, órgãos humanos, etc.

4.6 - Representação Analítica

A última forma de representação a ser apresentada é a Representação Analítica (*Analytic Representation - AREP*). Nessa técnica, um objeto sólido é representado através de uma composição de entidades tridimensionais, denominada Hipermalhas (*Hyperpatches - HYPT²*). Essas entidades tridimensionais não estão necessariamente contidas no *Conjunto-R*, definido anteriormente, e são, portanto, de difícil análise. A razão maior dessa representação ser dita analítica advém da forma pela qual os HYPT são definidos e modelados.

Um exemplo de tal representação pode ser visto na figura 4.12. Observa-se, de imediato, uma grande semelhança com uma representação do tipo CSG (ou uma variante desta). No entanto, essa forma de representação é bastante diferente. Uma primeira diferença reside na simples inexistência de uma hierarquia na estrutura das entidades tridimensionais utilizadas. Aqui, o sólido modelado é definido, simplesmente,

² O termo Hyperpatch é conhecido como uma extensão do termo Surfacepatch. De uma forma simplificada, pode-se afirmar que tais termos conceituam o espaço 3D e 2D, respectivamente.

através de uma lista de HYPT. Fora isso, tem-se que a representação individual de cada HYPT não possui nenhuma correlação com as primitivas sólidas utilizadas na proposta oferecida pelo modelo CSG.

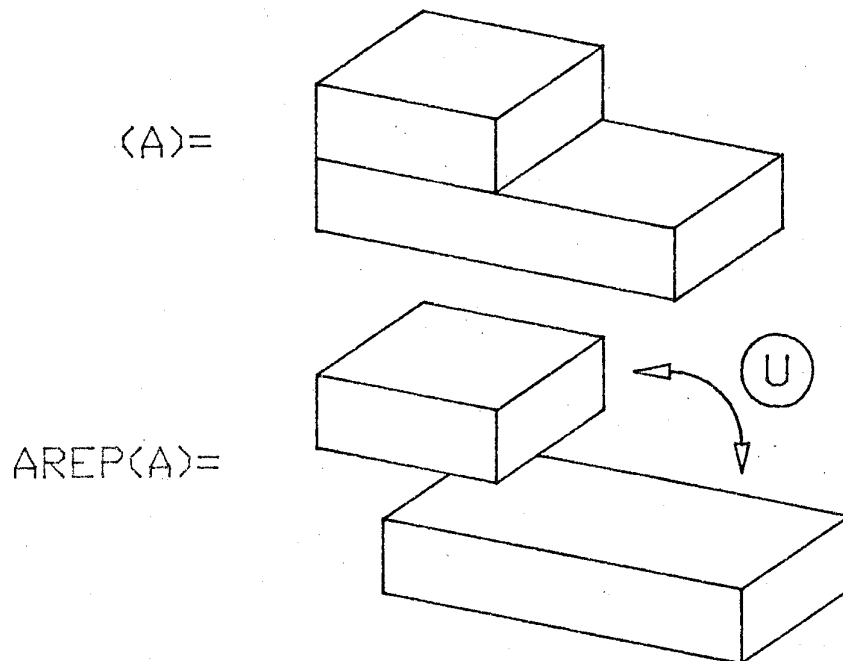


Figura 4.12 - Sólido representado através de AREP

4.6.1 - Hipermalhas

No exemplo apresentado acima, os HYPT's utilizados possuem todos faces planas ou cilíndricas. No caso geral,

entretanto, essas faces são Superfícies Esculturais (*Sculptured Surfaces*). Um HYPT pode ser definido como uma entidade tridimensional a partir da seguinte equação:

$$Z(\xi_1, \xi_2, \xi_3) = \sum_{k=1}^4 \sum_{j=1}^4 \sum_{i=1}^4 S_{ijk} \xi_1^{4-i} \xi_2^{4-j} \xi_3^{4-k}.$$

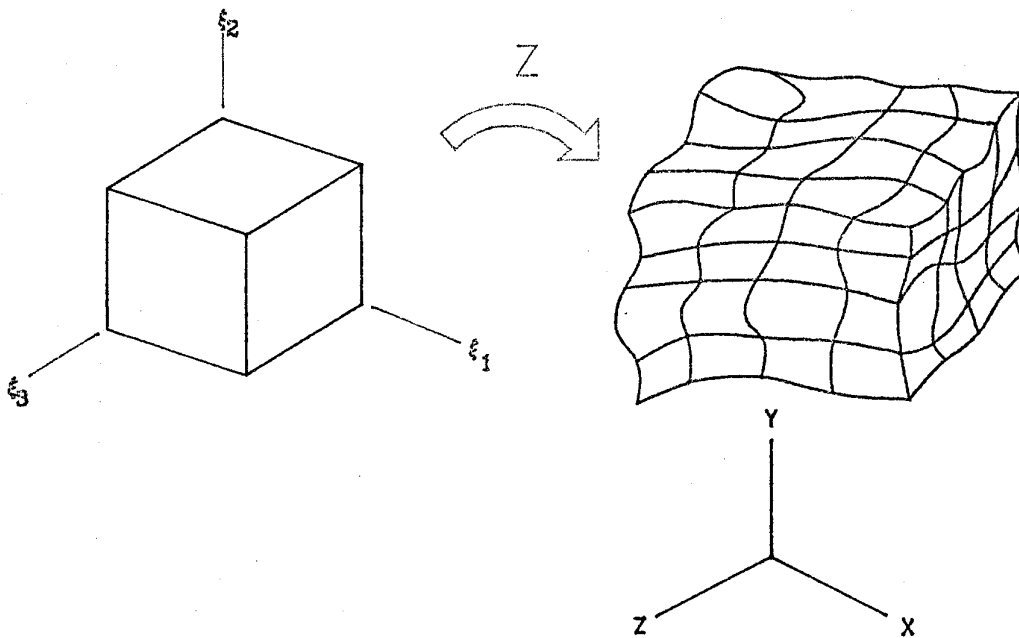


Figura 4.13 - O mapeamento de um *Hyperpatch*

Essa equação determina, matematicamente, um mapeamento paramétrico de um cubo unitário para o espaço tridimensional

(figura 4.13). O domínio desse mapeamento é, algumas vezes, denominado Espaço Paramétrico (*Parametric Space - PSP*). O grau desse mapeamento é cúbico em cada variável, sendo, portanto, a equação do HYPT uma função tri-cúbica, com 64 coeficientes vetoriais. Totalizando, esse HYPT seria definido por 192 números escalares.

Como um HYPT representa um sólido através de um polinômio paramétrico, este será descrito tanto no seu interior, quanto na sua fronteira. Tal característica é bastante marcante, em comparação com as outras formas de representação já vistas (suas implicações serão abordadas mais adiante). A informação com relação ao interior do sólido está disposta em 24 coeficientes na equação do HYPT. Esses coeficientes não existiriam se apenas as 6 (seis) superfícies bi-cúbicas, que delimitam a sua fronteira, fossem utilizadas. Constata-se, aqui, uma outra grande diferença para com uma representação CSG, que utilizasse *Half-Spaces* para definir as suas primitivas básicas. Como foi visto, *Half-Spaces* apenas determinam a fronteira das primitivas, não o seu interior. Os graus de liberdade, oferecidos adicionalmente por essa representação, são normalmente utilizados para o controle do mapeamento no interior do sólido, permitindo, conseqüentemente, que muitas informações a respeito da natureza interior deste sejam indicadas explicitamente.

Objetivando uma definição maior do contexto global de *Hyperpatches* (Sólidos Tri-Cúbicos), é possível se afirmar que estes nada mais são do que uma generalização de Curvas Cúbicas e de Superfícies Bi-Cúbicas, ou seja:

Sólidos Tri-Cúbicos:

$$Z(\xi_1, \xi_2, \xi_3) = \sum_{k=1}^4 \sum_{j=1}^4 \sum_{i=1}^4 S_{ijk} \xi_1^{4-i} \xi_2^{4-j} \xi_3^{4-k},$$

Superfícies Bi-Cúbicas:

$$Z(\xi_1, \xi_2) = \sum_{j=1}^4 \sum_{i=1}^4 S_{ij} \xi_1^{4-i} \xi_2^{4-j},$$

Curvas Cúbicas:

$$Z(\xi) = \sum_{i=1}^4 S_i \xi^{4-i}.$$

Em tal abordagem, constata-se que uma Curva Cúbica é o caminho que um ponto percorre ao se deslocar pelo espaço. Uma superfície Bi-Cúbica seria, então, definida pelo deslocamento de uma curva variável ao se mover em 3D. Por fim, considera-se um sólido como o conjunto dos pontos de uma superfície, também não fixa, que se desloca no espaço. Sendo assim, grande parte das propriedades associadas a curvas e superfícies também estão presentes nos Sólidos Tri-Cúbicos.

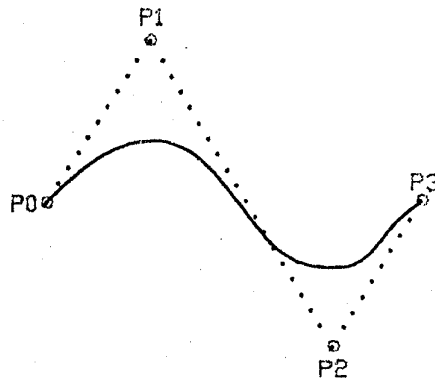


Figura 4.14 - Curva de *Bezier* e os seus pontos de controle

Para que um sólido possa ser descrito através de HYPT, os coeficientes S_{ijk} devem ser computados. A determinação da solução deste problema admite várias soluções. A mais popular delas faz uso de pontos de controle, a fim de gerar um sistema de equações que pode ser resolvido para os coeficientes desejados. Como exemplo, tem-se que, para Curvas Bi-Cúbicas, são necessários 4 (quatro) vetores de coeficientes. Utilizando-se curvas de *Bézier* (figura 4.14), pode-se calcular os coeficientes para quatro pontos de controle, da seguinte forma:

$$S_i = \sum_{l=1}^4 M_{il} P_l ,$$

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Uma generalização de tal procedimento pode ser feito para superfícies e sólidos, apenas modificando-se o cálculo dos coeficientes, da seguinte forma:

$$S_{ij} = \sum_{l=1}^4 \sum_{m=1}^4 M_{il} M_{jm} P_{lm}$$

$$S_{ijk} = \sum_{l=1}^4 \sum_{m=1}^4 \sum_{n=1}^4 M_{il} M_{jm} M_{kn} P_{lmn}$$

4.6.2 - Representação Interna

A utilização de uma representação analítica em SMS's permite estender, significativamente, as suas áreas de aplicação. Os recursos, que esta representação oferece ao possibilitar o reconhecimento exclusivo do interior de um sólido, pode ser compreendido a partir de um exemplo. SMS's têm sido freqüentemente utilizados em aplicações de animação tridimensional. Nesse tipo de aplicação, ao se desejar modelar um corpo humano, bastaria que este fosse descrito pela sua fronteira, considerando-se, portanto, o corpo como um sólido

rígido. Se agora fosse desejável fazer uma análise médica de um osso deste corpo, necessitar-se-ia de uma informação completa e precisa também do seu interior.

Essa informação extra é simplesmente impossível de ser adicionada em uma representação convencional por fronteira, principalmente, pela característica não discretizável desses dados. Situações análogas ocorrem em outras áreas, mas muito pouco (ou quase nada) tem sido feito, a fim de capacitar os atuais SMS's de estruturas simbólicas, para que representem dados internos dos objetos.

O método utilizado pela AREP para adicionar a informação da natureza interior dos sólidos, parte da possibilidade da equação de um Sólido Tri-Cúbico poder mapear dados não geométricos. Se for considerado que a equação de um HYPT é uma função vetorial de três variáveis e que qualquer uma dessas variáveis pode mapear, individualmente, valores escalares para o espaço de modelagem, através de uma função qualquer (definida pela aplicação), tal equação poderia, conseqüentemente, ser utilizada para expressar variações de temperatura, voltagem, densidade ou outra variável de estado (ou material). Combinando-se, agora, essa componente única com o mapeamento de um HYPT, obtém-se um mapeamento de três variáveis sobre quatro. As primeiras três componentes definem uma correspondência entre localizações paramétricas (ξ_1, ξ_2, ξ_3)

e espaciais (x,y,z) , enquanto que a quarta componente define uma função também sobre o espaço paramétrico (domínio). Desta forma, obtém-se uma implícita entre o espaço de modelagem e um valor escalar. Generalizando, qualquer associação de novas componentes pode ser feita, aumentando dessa forma o mapeamento de um HYPT (figura 4.15).

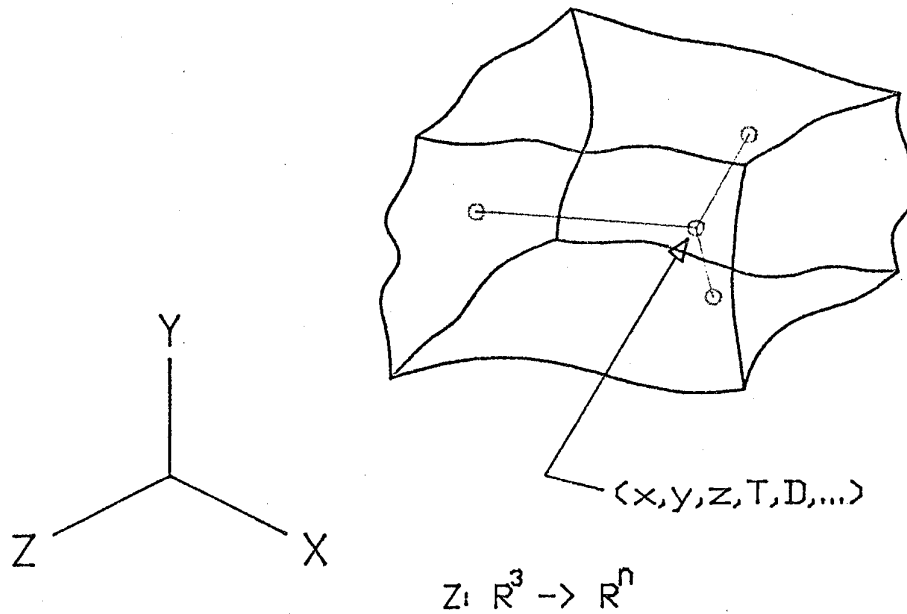


Figura 4.15 - Mapeamento extendido de um HYPT

Normalmente, tal associação de informação (sobre a forma de funções) é realizada somente após a modelagem completa do objeto, não influenciando, portanto, nenhuma análise dessa forma de representação no contexto de SMS's. Vários sistemas,

como o PATRAN [14], têm feito uso de tal funcionalidade da AREP, principalmente em aplicações de distribuição térmica, análise de elementos finitos, e cálculo de propriedades de massa.

4.6.3 - Verificação de Critérios e Conclusões

A partir dos critérios para a análise de formas de representação, vistos no capítulo 3, pode-se afirmar que, apesar de eficiente e concisa, a AREP não preenche por si só os outros critérios (Domínio, Validade, Completa, Única, etc.) de maneira satisfatória. A razão disto é bastante simples e reside no fato dessa representação não ser "avaliada", ou seja, ela não oferece meios para uma manipulação direta por parte dos algoritmos geométricos. Sendo assim, permite facilmente que sólidos inválidos ou ambíguos sejam gerados. Modificando-se os pontos de controle de uma superfície de *Bézier*, por exemplo, podem-se criar situações em que faces se interceptam mutuamente³. Em virtude disso, o próprio sistema deve garantir que as inconsistências não ocorram, ou seja, os algoritmos devem ser projetados de forma a completar e "proteger" a representação central.

³ Outros exemplos de sólidos inválidos podem ser gerados, simplesmente desobedecendo-se outras restrições, listadas no capítulo 3.

Uma tarefa normalmente simples em "Representações Avaliadas" pode ser bastante dispendiosa (computacionalmente) em uma AREP. Por exemplo, a simples tarefa de se qualificar um ponto 3D como fazendo parte ou não da fronteira de um sólido, equivale a resolver uma equação da seguinte forma:

$$Z(\xi_1, \xi_2, \xi_3) - P = 0,$$

onde Z identifica o mapeamento do HYPT, e P é o ponto a ser classificado. Esse problema é conhecido como o problema inverso do mapeamento paramétrico e exige processos numéricos complicados para a sua solução⁴.

A crescente escalada da demanda de representações, que possibilitem um mapeamento completo de objetos (no seu interior e na sua fronteira) e que ofereçam uma maior flexibilidade na sua edição (através de pontos de controle e métodos interativos), tem tornado a AREP uma saída bastante interessante e viável. Principalmente, no contexto de SMS's baseados no Modelo Objeto. Tais atributos têm sido bastante valorizados, permitindo uma fácil inclusão de novas e sofisticadas aplicações.

⁴ Esse problema é, também, muitas vezes utilizado como um anti-go teste, para comprovar se uma forma de representação é completa ou não.

CAPÍTULO 5

FORMAS AVALIADAS DE REPRESENTAÇÃO

5.1 Enumeração por Ocupação Espacial

Uma outra forma de representação para um objeto tridimensional é a Enumeração por Ocupação Espacial (*Spatial Occupancy Enumeration* - SOE), na qual uma lista de células espaciais, ocupadas por aquele, é armazenada como forma de representação interna.

Essas células, normalmente denominadas *Voxels*¹, são cubos de dimensão fixa, posicionados sobre uma malha espacial também fixa (figura 5.1). Outras decomposições (como, por exemplo, a tetraedral) do E^3 poderiam ser igualmente utilizadas, mas, para simplificar, é analisada apenas a malha em cubo, tendo em vista o fato do tipo desta não afetar as propriedades básicas da SOE.

Cada célula é identificada pela coordenada de um único ponto (3D), como, por exemplo, o seu centro de massa. É convencional, ainda, uma ordem fixa para o processo de

¹ O termo Voxel é normalmente utilizado como uma abreviatura em inglês para um "elemento de volume".

varredura espacial sobre as tuplas de coordenadas, denominadas de vetores espaciais. Essa ordem é de fundamental importância, pois todos os algoritmos, que irão manipular tal estrutura, são de alguma forma projetados, assumindo-se essa convenção. Como exemplo, pode-se citar um algoritmo de visualização que, para detectar células encobertas, utiliza o fato da malha espacial estar fixa no espaço e do vetor espacial impôr uma ordenação lógica às mesmas.

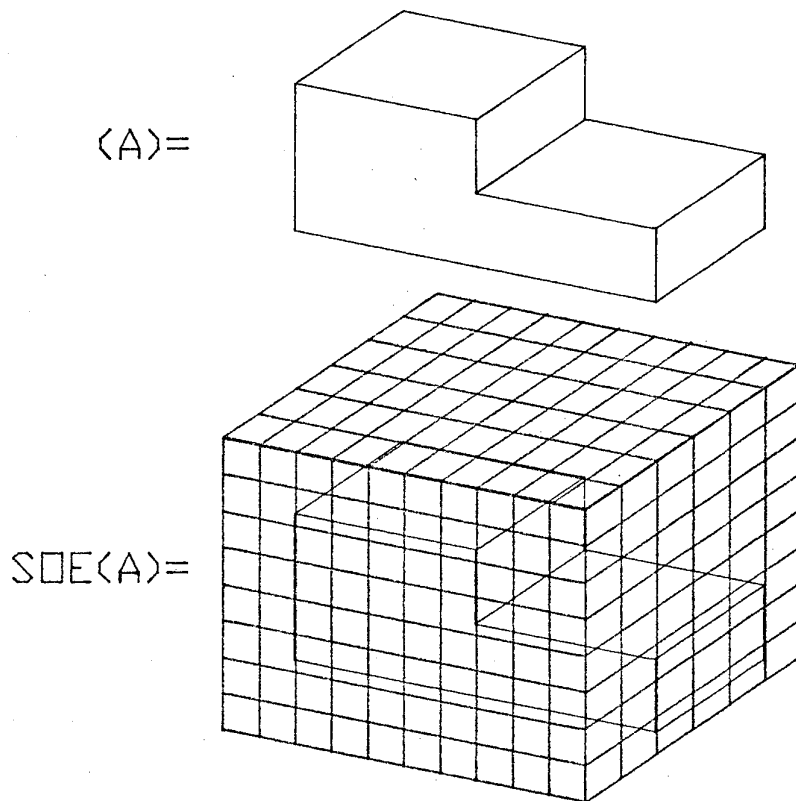


Figura 5.1 - Malha espacial de representação

Vetores espaciais podem ser classificados como não ambíguos, únicos (com exceção, é claro, de unicidade posicional²) e facilmente validáveis; são, no entanto, potencialmente difusos e prolixos, quando não redundantes. O grau de difusidade, associado a essa forma de representação, está intimamente relacionado ao fato da classe dos objetos físicos modelados se encaixar adequadamente ou não no domínio de tal representação.

Sendo assim, vetores espaciais podem ser representações razoáveis em algumas aplicações, como por exemplo:

- (i) Aplicações na Arquitetura, onde as construções são suficientemente modulares e uniformes;
- (ii) Aplicações na Medicina, como tomografias, onde objetos biológicos irregulares são modelados através de aproximações por poliedros;
- (iii) Aplicações na Geologia, nas quais uma aproximação do relevo de uma região pode ser facilmente descrita pela SOE;
- (iv) Aplicações em Robótica, em verificações de colisão e reconhecimento de ocupação espacial.

² A unicidade posicional, como já foi constatado anteriormente, implica no fato de duas entidades dimensionais não poderem possuir o mesmo valor numérico, dentro de uma mesma representação. Dois Vértices de um mesmo objeto, por exemplo, não poderiam possuir as mesmas coordenadas no espaço.

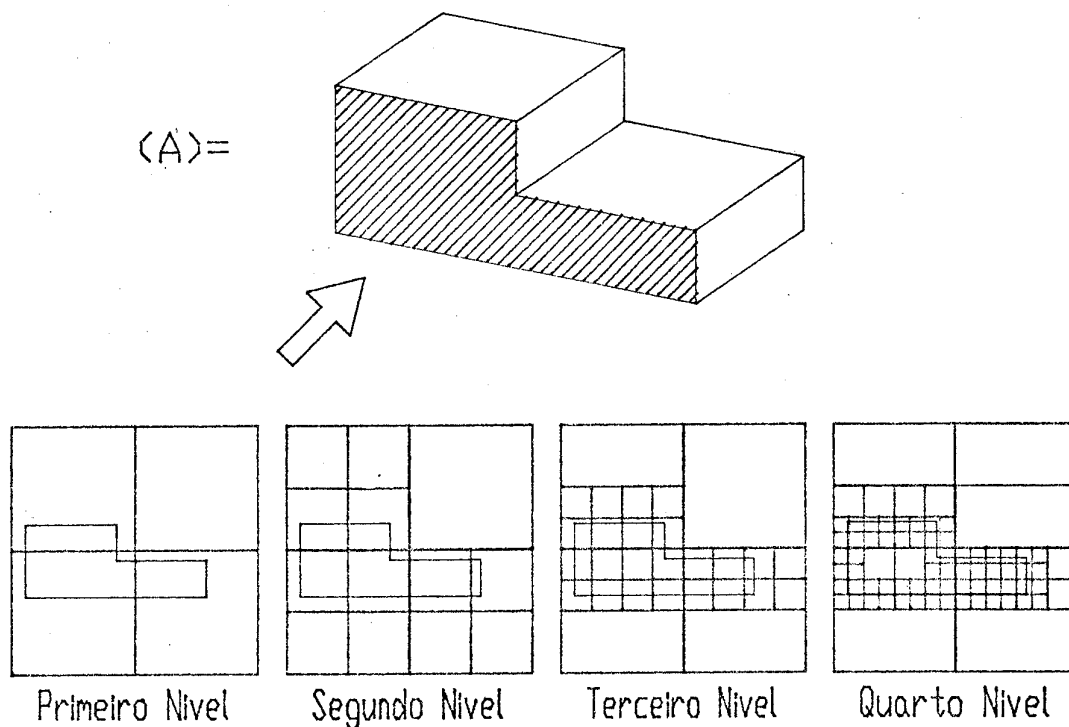


Figura 5.2 - Vista lateral de uma SOE baseada em OCD

No entanto, apesar da grande variedade de aplicações mencionadas acima, observa-se que, para a representação de objetos precisos (peças mecânicas, por exemplo) que não possuem um isomorfismo topológico³ com o tipo de célula utilizada (no presente contexto, cubos), vetores espaciais são demasiadamente

³ O isomorfismo topológico impõe uma série de exigências naturais com relação à forma dos objetos. É demonstrado que objetos isomorfos entre si possibilitam a utilização comum de um mesmo tipo de célula para serem representados precisamente.

difusos para o uso prático, como uma forma primária de representação. Em função disso, desenvolveram-se implementações SOE, estruturadas sobre uma decomposição em Árvores Octógonas (*Octree Decomposition - OCD*). Essa subdivisão espacial é construída em níveis, de forma que, à medida que se passa de um nível para o próximo, a resolução da divisão é dobrada em cada coordenada (x, y, z). Tal fenômeno, no entanto, só ocorre para as células que são interceptadas pelo objeto representado. Células totalmente internas ou externas não precisam, portanto, ser subdivididas (figura 5.2).

Dessa maneira, o espaço que um determinado objeto ocupa, para uma dada resolução, é consideravelmente reduzido. Não obstante isso, se se tomar uma baixa resolução para a discretização dos objetos, tal forma de representação pode continuar sendo imprecisa (figura 5.3).

Entretanto, essa forma de representação, quando observada em um esquema híbrido de representação, pode se tornar bastante útil, pois permite a utilização de uma série de algoritmos geométricos capazes de elevar significativamente o desempenho geral de um SMS. Em tal configuração, um mesmo objeto poderia ser representado internamente através de mais de uma forma de representação combinadas entre si.

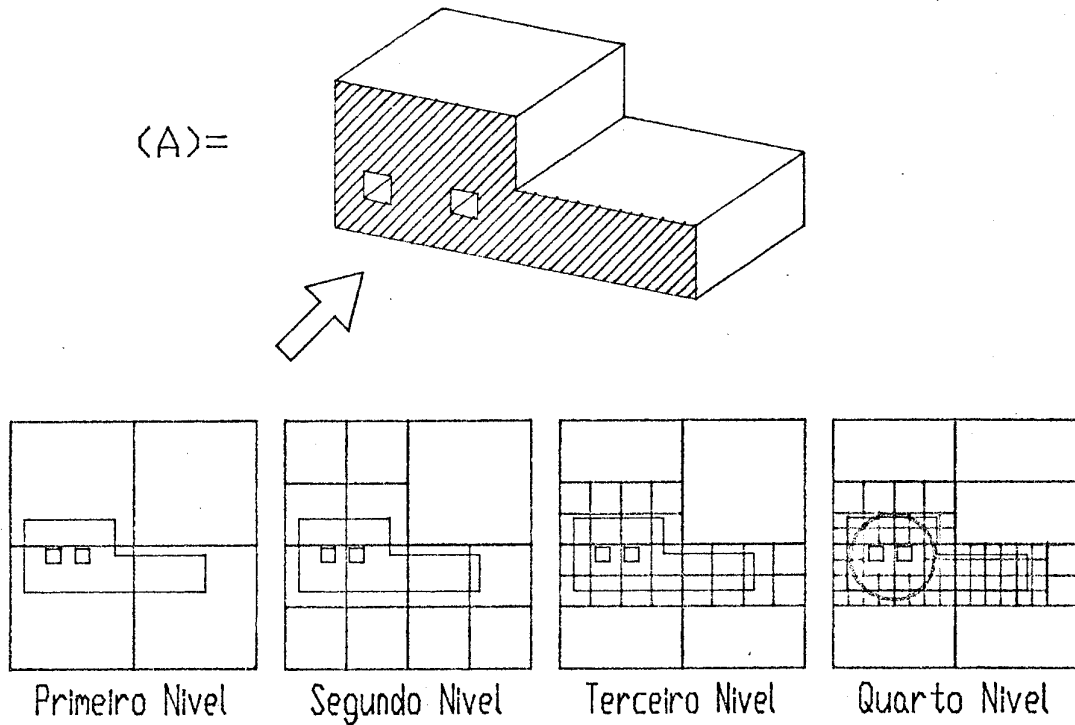


Figura 5.3 - Furo "perdido" por uma baixa resolução

5.2 Decomposição em Células Espaciais

Dentro do contexto de SMS, pode-se, também, representar um sólido através da sua Decomposição em Células Espaciais (*Spatial Cell Decompositions - SCD*), na qual cada célula é representada individualmente. A técnica de Enumeração por Ocupação Espacial (*SOE*), discutida na seção anterior, pode ser vista como um caso particular da SCD, na

qual todas as células eram cubos e estes deveriam pertencer a uma malha fixa.

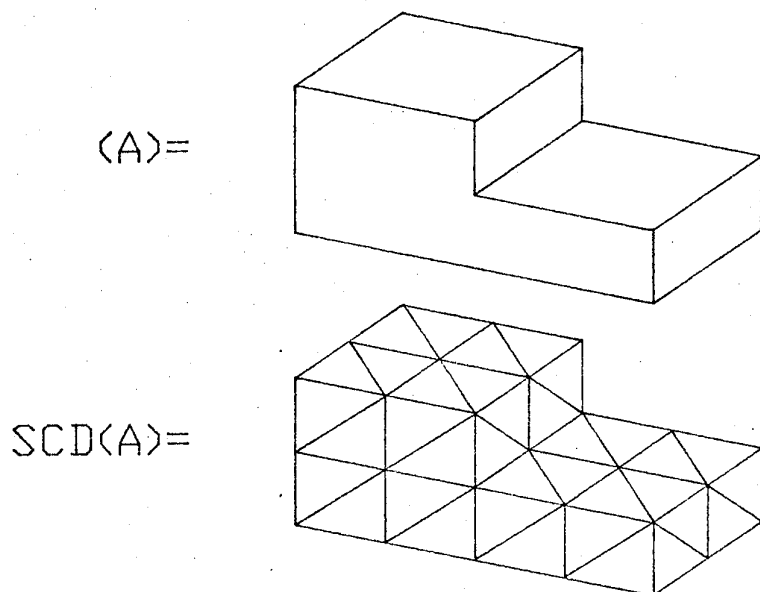


Figura 5.4 - Representação por SCD

As células em uma representação SCD são escolhidas através de um processo de decomposição do sólido original em *Voxels*. Uma maneira de se realizar esta decomposição é através da triangularização de um poliedro plano em tetraedros, que, por sua vez, devem ser disjuntos ou devem se interceptar, precisamente, apenas em faces, arestas ou vértices comuns. Poliedros curvos também podem ser decompostos em tetraedros curvos, desde que estes satisfaçam as exigências já mencionadas

para poliedros planos. Pode-se facilmente observar que o processo de decomposição em células é uma generalização da técnica de triangularização. A analogia se torna mais clara ao se substituírem células 2D na triangularização por uma versão 3D destas, na decomposição de um sólido (figura 5.4).

Independentemente do tipo de célula utilizada, nenhuma restrição é feita com relação ao número de lados que esta deve possuir. O termo "lados", às vezes ambíguo, refere-se, em 2D, à região bidimensional delimitada por uma aresta, enquanto que, em 3D, ter-se-ia a região tridimensional cortada por uma face genérica.

A dificuldade maior, inerente à utilização de uma representação do tipo SCD, reside no fato de não ser única, e a sua validade ser de difícil comprovação, apesar de não ser ambígua. Além disto, não se conhece, ainda, uma metodologia formal, para que se realize o processo de decomposição de um sólido em células válidas (especialmente em se tratando de sólidos curvos).

Entretanto, algumas propriedades topológicas dos objetos representados através da SCD podem torná-la bastante interessante. Existem, por exemplo, algoritmos conhecidos para a determinação do fato de um Sólido ser ou não convexo, ou seja, se este possui regiões interiores vazias ou não.

Propriedades geométricas também podem ser facilmente computadas através de SCD, como sugerem as operações para o cálculo das propriedades de massa de um sólido (volume, área, momentos de inércia, centro de massa, etc.). Esse tipo de aplicação é bastante freqüente, tendo em vista o fato da decomposição em células exigir que estas sejam disjuntas, permitindo, em consequência, que tais propriedades sejam computadas através de simples somatórios parciais.

É importante lembrar, ainda, que a SCD é a forma de representação utilizada em Métodos de Elementos Finitos (*Finite Element Methods - FEM*⁴) para a resolução de equações diferenciais e merece, portanto, uma atenção especial quando da sua consideração em uma implementação voltada para um sistema híbrido de representação.

5.3 - Representação por Fronteira

Dentre todas as formas de representação utilizadas atualmente em SMS's, a mais conhecida e importante é, sem sombra de dúvida, a Representação por Fronteira (*Boundary Representation - BREP*). Como o próprio nome já indica, essa técnica representa um sólido a partir da descrição da sua fronteira com o espaço. Descrever a fronteira de um objeto

⁴ Atualmente já existem outras representações sendo utilizadas para tal aplicação. Entre elas, se destaca a BREP modificada, que é capaz de representar Sólidos não Múltiplos [48].

tridimensional pode ser visto, de uma forma simplificada, como o processo de se escolherem em entidades tridimensionais (geométricas e topológicas), que, quando associadas, instanciam completamente a descontinuidade existente entre o interior e o exterior de um sólido (figura 5.5).

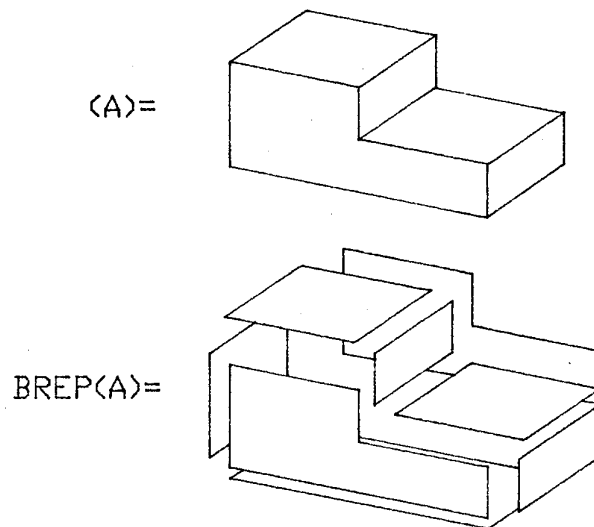


Figura 5.5 - Sólido representado através da BREP

Historicamente, a importância da BREP em SMS's só é comparável com a utilização de dispositivos gráficos interativos na filosofia do Projeto Assistido por Computador (CAD). Como já foi visto anteriormente, uma das primeiras funcionalidades, que um SMS deve possuir, é a de permitir meios (de preferência interativos) para a modelagem dos objetos.

Neste processo, a visualização gráfica⁵ é de fundamental importância, tendo sido consistentemente suportada pela BREP. A razão maior dessa intrínseca ligação reside na semelhança natural entre a forma com que objetos são representados em uma BREP e visualizados graficamente. Linhas contidas nas projeções de um objeto, por exemplo, poderiam estar diretamente relacionadas com as *Bordas* físicas deste, e estas poderiam ser representadas numa BREP pela entidade topológica *Aresta*. Generalizando-se essa analogia, pode-se apresentar a essência da BREP através de um simples mapeamento entre entidades físicas, topológicas e geométricas (tabela 5.1).

ENTIDADES		
Físicas	Topológicas	Geométricas
<i>Sólido</i>	<i>Casca</i>	<i>Sub-Espaço</i>
<i>Fronteira</i>	<i>Face</i>	<i>Superfície</i>
<i>Borda</i>	<i>Aresta</i>	<i>Curva</i>
<i>Canto</i>	<i>Vértice</i>	<i>Ponto</i>

Tabela 5.1 - Relacionamento natural entre entidades.

⁵ A visualização gráfica tridimensional apresenta-se, no momento, como a principal área de pesquisa em computação gráfica, constituindo, também, uma das principais aplicações para SMS's.

Essa associação de entidades pode ser vista como uma maneira formal de se estruturar uma representação por fronteira, sem, no entanto, provocar uma particularização da classe dos objetos efetivamente representáveis (domínio da BREP). Observa-se que esta "analogia da representação" independe da forma com que essas entidades (topológicas ou geométricas) serão implementadas em uma estrutura simbólica.

A BREP tem sido quase sempre utilizada como a forma primária de representação em SMS's. Isso se deve, principalmente, ao fato dela possuir, dentre todas as outras formas de representação já analisadas, a maior capacidade de representação geométrica. Em outras palavras, pode-se dizer que, geometricamente, o contradomínio da BREP é o mais extenso, e os sólidos nela representados estão quase sempre contidos no *Conjunto-R*. Além disso, a BREP é também bastante valorizada por ser uma representação já totalmente avaliada, ou seja, o processo de BEV é inerente à própria definição dessa representação. Conseqüentemente, outras formas de representação encontram na BREP todo o suporte necessário para conversões e para a troca de informação.

Como já foi visto, toda a força de representação contido na BREP origina-se da união associativa de entidades topológicas e geométricas. No contexto de SMS's orientados para o Modelo Objeto, introduz-se uma outra classe de entidades,

denominada Atributos, à forma de representação utilizada. Quando a forma de representação considerada é a BREP, esta nova associação de atributos se faz de uma forma bem natural e prática, pois a BREP já inclui entidades topológicas que são utilizadas quase que como elos para a comunicação de dados externos ao ambiente da modelagem propriamente dito. Como exemplo, poderia-se ter um módulo aplicativo para o processamento de imagens, que desejasse visualizar cada *Face* de um objeto com uma determinada cor. Isso poderia ser feito com a adição de um índice de cor associado à entidade topológica *Face*, não comprometendo, em nenhum momento, a consistência da BREP.

Durante a difícil fase da modelagem tridimensional de um objeto, ou seja, durante as etapas em que uma determinada representação é criada e alterada, observa-se a existência de um vasto conjunto de técnicas, altamente favorecidas pela utilização de uma BREP. Essas técnicas dizem respeito quase que exclusivamente às operações locais sobre um determinado objeto, isto é, às tarefas, onde já existe uma representação aproximada do mesmo, e se deseja, apenas, efetuar algumas pequenas alterações geométricas ou topológicas⁶. Numa BREP, a tarefa de se deslocar levemente um vértice de um objeto, por exemplo, se resume a alterar o conteúdo da informação geométrica associada

⁶ Um exemplo de uma operação topológica sobre um objeto poderia ser a divisão de uma *Face* em duas, através da inserção de uma *Aresta* no seu interior.

a este (figura 5.6). As conhecidas operações de *Extrude*, *Slice*, *Blend*, *Chamfer*, entre outras, podem ser fácil e eficientemente implementadas em um sistema baseado na BREP, minimizando o *overhead* do usuário durante a modelagem e simplificando a manutenção do sistema [15,18].

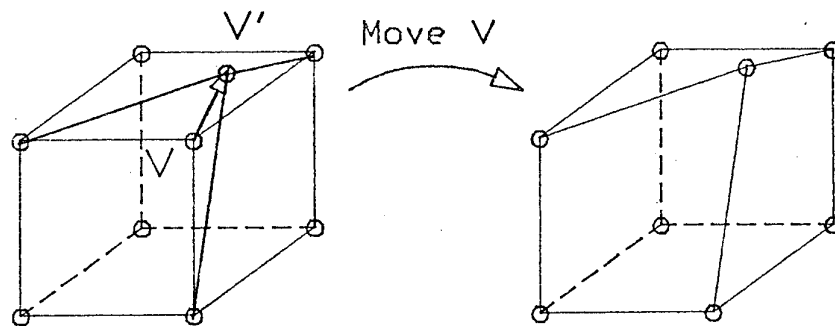


Figura 5.6 - Operação geométrica local

Durante as partes que se seguem neste trabalho, serão utilizados, freqüentemente, termos para denotar entidades topológicas e suas relações. Atualmente, já existe um senso comum sobre uma nomenclatura internacional para estes termos. Sendo assim, apresenta-se, aqui, uma pequena tabela de conversão entre os termos padronizados (em inglês) e os seus significados correspondentes na língua portuguesa (tabela 5.2).

ENTIDADES		
Abreviação	Inglês	Português
B	<i>Solid</i>	<i>Sólido</i>
S	<i>Shell</i>	<i>Casca</i>
V	<i>Vertex</i>	<i>Vértice</i>
F	<i>Face</i>	<i>Face</i>
E	<i>Edge</i>	<i>Aresta</i>
L	<i>Loop</i>	<i>Ciclo</i>
G	<i>Genus</i>	<i>Genus</i>
R	<i>Ring</i>	<i>Anel</i>

Tabela 5.2 - Conversão entre termos padronizados

5.3.1 - Integridade Topológica

Entidades topológicas são *Cascas*, *Faces*, *Arestas*, *Vértices* e *Ciclos*, caracterizando, unidas, o núcleo topológico de uma BREP. Essas entidades são comumente apresentadas como um "adesivo", capaz de manter a coerência e a consistência de uma representação por fronteira, sem, no entanto, "fixar" definitivamente a sua forma geométrica. Na figura 5.7, encontra-se uma boa demonstração para a liberdade geométrica que uma específica representação topológica de uma *Face* pode permitir.

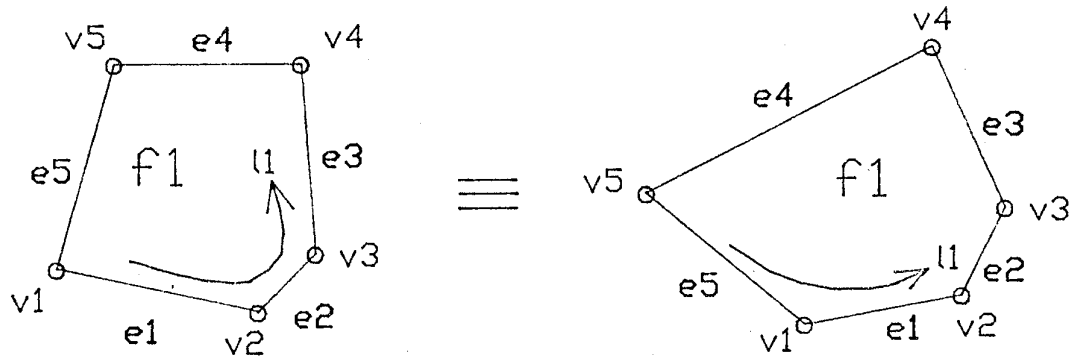


Figura 5.7 - Mesma topologia para diferentes geometrias

Em uma outra abordagem acerca das entidades topológicas, um objeto é definido como sendo feito de borracha, cuja forma só é determinada quando sua geometria é fixada.

A BRKP foi apresentada, inicialmente, como uma forma de representação, na qual a superfície de um objeto é decomposta em uma ou mais partes, e cada uma destas devem ser, ao longo da sua fronteira, totalmente descritas. Essa fronteira toma normalmente a forma de superfícies, curvas ou pontos. É necessário, ainda, que essa forma de representação seja capaz de descrever como cada um dos pedaços de superfície se encaixam, de maneira a formar um volume fechado e orientável. Nesse momento, entidades topológicas possuem uma fundamental importância, a fim de responderem questões sobre a organização relacional da representação de um sólido. Tais questões são

mais conhecidas como as relações de adjacência entre entidades topológicas (figura 5.8) [4,8,47,52].

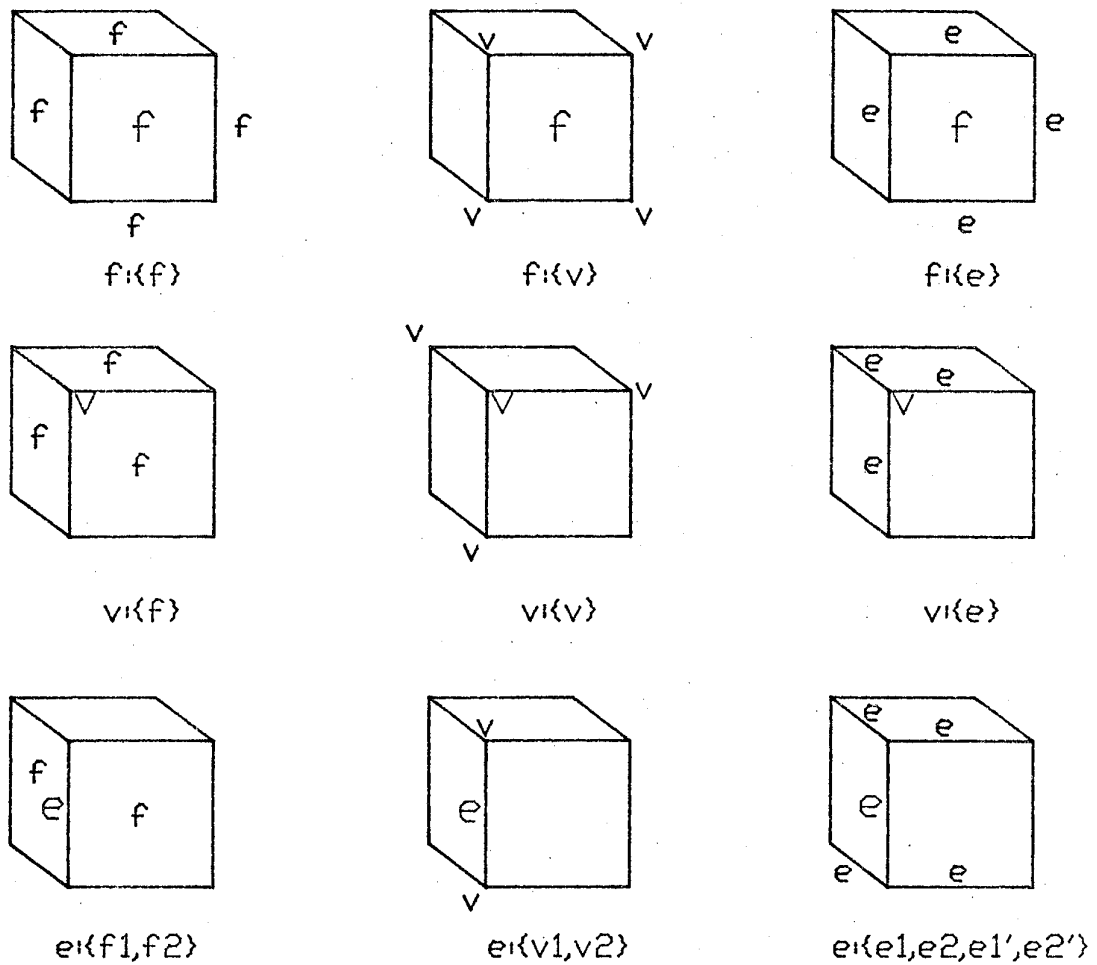


Figura 5.8 - Algumas relações de adjacência

Um exemplo de uma relação de adjacência é a que existe entre uma *Face* e as *Arestas* que delimitam a sua fronteira. Em

SMS's, é freqüente a necessidade de se estabelecer este tipo de relação, objetivando, por exemplo, a geração de uma visualização gráfica do tipo *Wire-Frame*.

As relações de adjacência entre os vários componentes de uma superfície poderiam ser obtidos a partir de técnicas numéricas, analisando-se proximidades geométricas entre os pedaços dessas superfícies. Esta saída envolveria, no entanto, uma série de problemas de precisão numérica, bem como um custo computacional insuportável.

Para que entidades topológicas e as suas relações de adjacência possam ser efetivamente utilizadas numa BREP, algumas regras específicas e definições, devem ser apresentadas:

- (i) A *Casca* de um sólido é um conjunto conexo de *Faces*, descrevendo a sua fronteira com o espaço. Um sólido deve possuir, pelo menos, uma *Casca*, mas, nos casos em que um sólido possua regiões interiores vazias, estas também deverão ser descritas como *Cascas* separadas;
- (ii) Uma *Face* é uma região fechada;
- (iii) A fronteira de uma *Face* é um conjunto conexo de *Vértices* e *Arestas* alternados, denominado *Ciclo*. Em uma analogia com o problema de se representarem regiões vazias no interior de um sólido, é, às vezes,

necessário representar *Faces* com furos. Nesses casos, utiliza-se um *Ciclo*, para representar a sua fronteira exterior, e um ou mais *Ciclos*, para cada furo contido no seu interior. *Ciclos* interiores a uma *Face* são também conhecidos com o termo *Anel*;

(iv) Uma *Aresta* pode pertencer a uma ou a duas *Faces*, mas não a mais de duas. Isto equivale a se exigir que *Faces* nunca se interceptem, senão em *Arestas*;

(v) *Arestas* devem sempre se encontrar em *Vértices*, ou seja, em cada extremidade de uma *Aresta* deve existir sempre um *Vértice*. Existem situações, em que uma *Aresta* pode ter o mesmo *Vértice* nas suas duas extremidades.

Um sólido será dito topologicamente válido, se obedecer às regras especificadas acima. A figura 5.9 demonstra uma representação topologicamente válida. Nota-se, que a decomposição do sólido em suas entidades topológicas, é consistente e estruturada em hierarquias, que preservam as leis topológicas apresentadas acima.

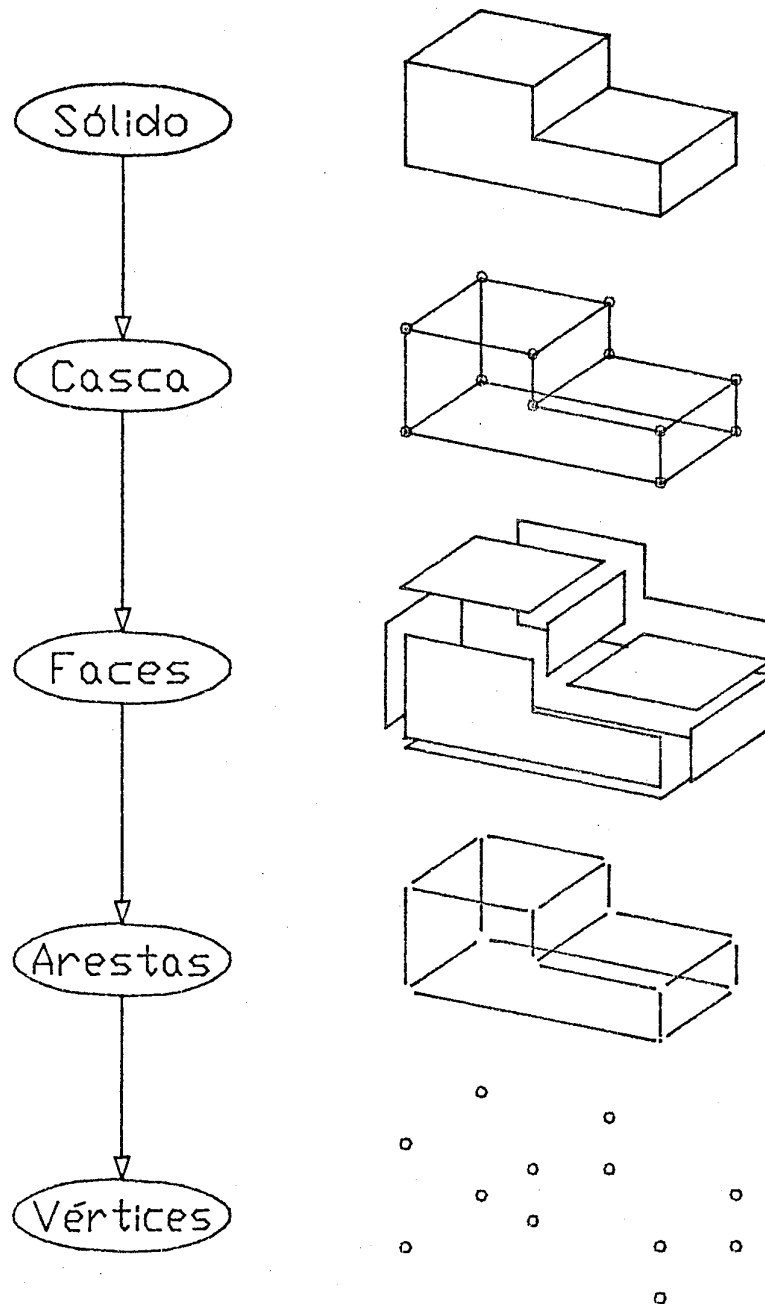


Figura 5.9 - Representação topologicamente válida

5.3.1.1 - Fórmula de Euler-Poincaré

A principal dificuldade, associada à utilização da BREP em SMS's, reside, exatamente, em se garantir que as exigências topológicas sejam sempre satisfeitas durante os processos de modelagem. A resposta mais atual para esse problema advém, no entanto, de uma observação feita em 1893 por *Poincaré*, ao estudar o Teorema de Euler [28]. *Poincaré* constatou que esse teorema dizia respeito a entidades puramente topológicas e não geométricas. O famoso número:

$$V - E + F, \quad (1)$$

também conhecido como um invariante topológico, na verdade, é uma extensa particularização da fórmula genérica:

$$V - E + F = 2 * (S - G) + R, \quad (2)$$

adequada à caracterização topológica de objetos fechados e orientáveis [3,10,31,49]. Nesta fórmula, V, E, F, S, R, e G denotam, respectivamente, o número de *Vértices*, *Arestas*, *Faces*, *Cascas*, *Aneis* (*Ciclos* interiores a uma *Face*) e o *Genus* do sólido considerado.

A nova entidade topológica *Genus*, contida na fórmula de *Euler-Poincaré*, merece uma delicada explicação matemática,

pois costuma ser alvo de muita confusão. Para tanto, existe a necessidade de se definir, inicialmente, o conceito de *homeomorfismo*. Dois objetos P e Q são *homeomorfos* entre si, quando existe uma transformação contínua:

$$f : P \rightarrow Q ,$$

cuja inversa

$$f^{-1} : Q \rightarrow P$$

também é contínua. Dessa forma, diz-se que f é um *homeomorfismo* de P sobre Q . De uma maneira menos formal, pode-se ilustrar o *homeomorfismo* entre dois objetos a partir de um simples exemplo. Se forem considerados dois poliedros regulares P e Q , feitos de borracha, e, ao serem inflados, injetando-se ar, ambos serão transformados em esferas. Pode-se concluir, portanto, que P e Q são *homeomorfos* a uma esfera (figura 5.10-a). Não obstante tal fato, se P e Q forem ambos furados, ao serem inflados, irão se transformar em toros, ou seja, P e Q serão *homeomorfos* a um toro (figura 5.10-b). Resumidamente, pode-se afirmar que o *Genus* de um objeto é o número de "passagens internas" que esse possui. Dessa maneira, todos os objetos *homeomorfos* a uma esfera terão *Genus* igual a 0, e todos que forem *homeomorfos* a um toro serão classificados com *Genus* igual a 1, e assim por diante.

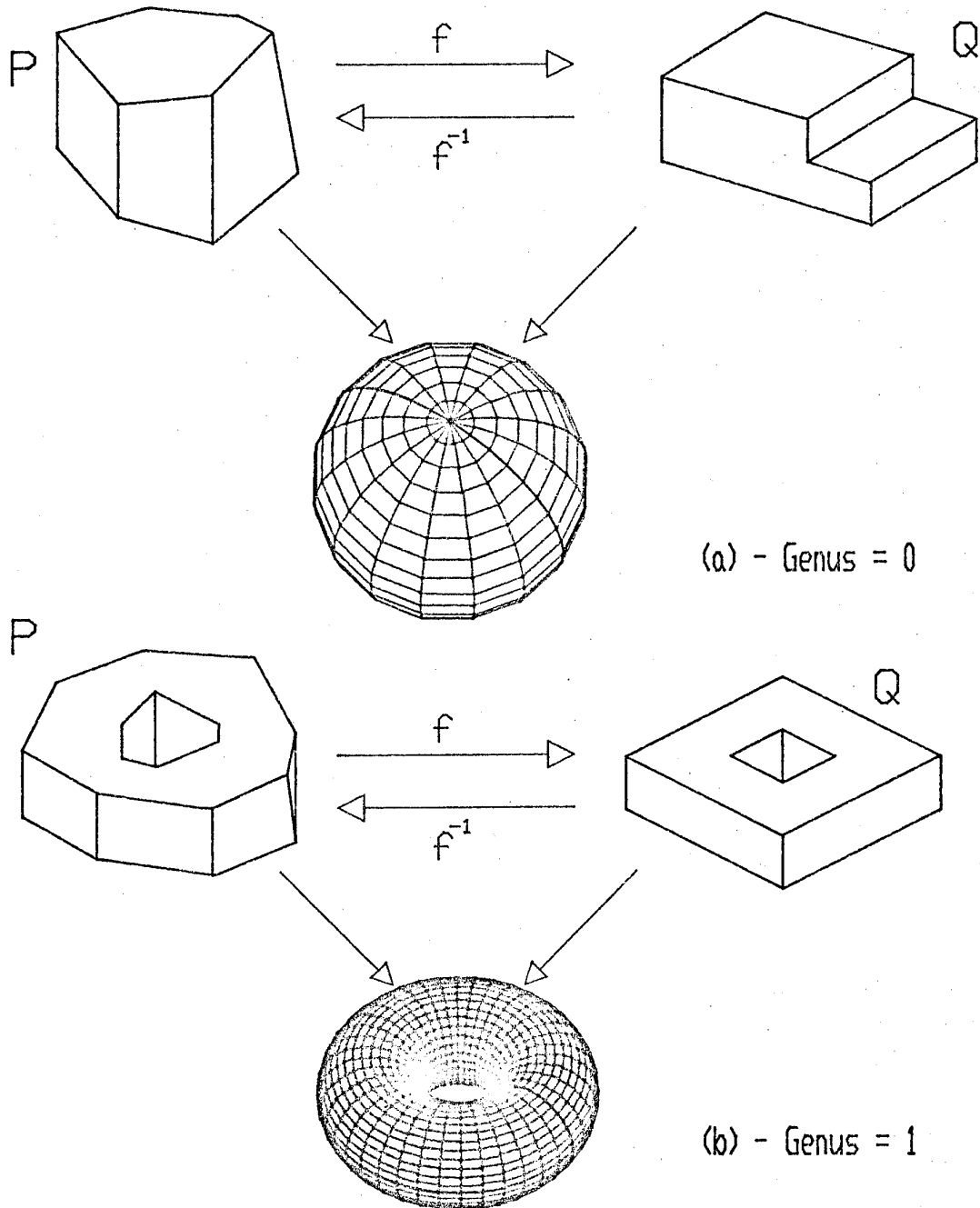


Figura 5.10 - Exemplos de Homeomorfismos

Depois da apresentação da entidade *Genus*, pode-se explicar melhor a real importância da fórmula de *Euler-Poincaré* para a modelagem de sólidos tridimensionais. Como já foi visto, um dos maiores problemas com a *BREP* é a necessidade de se garantir que as regras topológicas, referentes a tal representação, sejam sempre satisfeitas. Ao se, utilizar essa fórmula para testar a topologia de um objeto, poder-se-á facilmente classificá-lo como válido ou não. Tal procedimento seria, no entanto, bastante dispendioso computacionalmente, pois teria de ser realizado após cada etapa da modelagem. Em virtude disso, propôs-se uma metodologia diferente, na qual o próprio objeto seria modelado através de Operadores de Euler (*Euler Operators - EOP*).

Analisando-se a fórmula (2), nota-se que esta representa uma equação de uma Hipermalha discreta, de dimensão 6, contida no espaço de mesma dimensão. Ou seja, o vetor genérico $\langle v, e, f, g, s, r \rangle$ especifica, de forma única, uma posição nessa Hipermalha, isto é,

$$\nabla z \cdot \underline{x} = 0 \quad , \quad (3)$$

onde

$$\underline{x} = \langle v, e, f, g, s, r \rangle \quad e \quad (4)$$

$$z(\underline{x}) = V - E + F - 2S + 2G - R \quad (5)$$

Todos os objetos válidos podem ser representados por um ponto nessa malha, mas o inverso não é necessariamente verdade, pois existem, ainda, algumas restrições topológicas (não mencionadas), que limitam as opções. São elas:

- (i) V, E, F, S, G, R devem ser todos ≥ 0 ;
- (ii) Se $V = E = F = R = 0$ então $G = S = 0$;
- (iii) Se $S > 0$ então $V > S$ e $F > S$.

Naturalmente, essa malha pode ser gerada por uma coleção de 6 vetores linearmente independentes que satisfaçam a equação (2). Cada vetor desse tipo é denominado, informalmente, um Operador de Euler. Diferenciando-se a equação (3), obtém-se:

$$\nabla z \cdot \underline{d} = 0, \quad (6)$$

onde \underline{d} é uma transição correspondente a um Operador de Euler. Ao se restringirem as transições para àquelas, que envolvam apenas um deslocamento unitário em cada eixo, observam-se 98 opções diferentes a serem escolhidas. Destas, 54 são combinações das outras, restando 43 transformações capazes de alterarem um objeto em um, dois, ou três eixos, juntamente, com a transformação nula.

EOP	Transição						Descrição abreviada
	V	E	F	R	S	G	
MVFS	1	0	1	0	1	0	cria <i>Vértice, Face, Casca</i>
MVE	1	1	0	0	0	0	cria <i>Vértice, Aresta</i>
MEF	0	1	1	0	0	0	cria <i>Aresta, Face</i>
KEMR	0	-1	0	1	0	0	remove <i>Aresta</i> , cria <i>Anel</i>
KFMRG	0	0	-1	1	0	1	remove <i>Face</i> , cria <i>Anel, Genus</i>
KVFS	-1	0	-1	0	-1	0	remove <i>Vértice, Face, Casca</i>
KVE	-1	-1	0	0	0	0	remove <i>Vértice, Aresta</i>
KEF	0	-1	-1	0	0	0	remove <i>Aresta, Face</i>
MEKR	0	1	0	-1	0	0	cria <i>Aresta</i> , remove <i>Anel</i>
MFKRG	0	0	1	-1	0	0	cria <i>Face</i> , remove <i>Anel, Genus</i>

Tabela 5.3 - Um possível conjunto gerador de EOP

Esses operadores poderiam ser utilizados incrementalmente, para se construir, desde o vazio, uma completa representação topológica de um objeto. Tal construção estaria, por conseguinte, totalmente isenta de qualquer tipo de ambigüidades topológicas, liberando o sistema da árdua tarefa de garantir a integridade topológica da BREP. Na tabela 5.3, encontra-se uma possível escolha para o conjunto gerador desses vetores, levando-se em consideração o menor deslocamento possível na Hipermalha e, é claro, um sentido prático para a

modelagem e implementação desses EOP. Na prática, cada um desses operadores é identificado por um mnemônico, iniciado, pelas letras M ou K, indicando, respectivamente, as operações de criação (*Make*) e remoção (*Kill*) de uma tupla de valores.

Uma propriedade bastante interessante desses Operadores de Euler é o fato de todos possuírem inversas, permitindo, portanto, que operações de UNDO e REDO sejam facilmente implementadas em um SMS¹ [16,17,19,30,44].

Grande parte dos SMS's em desenvolvimento (baseados na BREP) faz farto uso da filosofia dos EOP. Em muitos casos, inclusive, os objetos são representados internamente como uma seqüência de operadores, que, quando executados, geram a BREP correspondente. Em tal abordagem, insere-se, junto aos EOP, toda a informação geométrica necessária para a "fixação" da sua forma física. Sistemas deste tipo costumam ser robustos e de fácil manutenção [20,32].

Um fator ainda bastante positivo dessa metodologia é o fato dela não amarrar, em momento algum, o formato interno da BREP, ou seja, a sua estrutura de dados é totalmente independente dos procedimentos de modelagem. Todas as operações de modelagem são descritas por intermédio de uma seqüência de

¹ Operações de UNDO e REDO possibilitam a retomada ou a repetição de um determinado estado simbólico de uma estrutura de representação.

EOP, juntamente com sua informação geométrica. Sendo assim, apenas os implementadores dos EOP devem estar atentos para a estrutura de dados a ser utilizada.

Atualmente, já existe o consenso em que um conjunto fixo de Operadores de Euler seja padronizado, de forma a oferecer, assim, uma maior capacidade de integração das diversas técnicas de modelagem conhecidas, baseadas na BREP. Vários autores, como *M. Mantyla* [29,33,35] e *I.C. Braid* [9,10], já demonstraram que, para garantir a integridade topológica de uma BREP, é indispensável a utilização de EOP.

5.3.2 - Integridade Geométrica

Garantir a integridade topológica de uma BREP é bastante importante, mas não se pode esquecer de assegurar que o modelo topológico possua um significado físico aceitável, ou seja, que a sua geometria seja consistente. De nada adiantaria observarem-se objetos topologicamente corretos, cuja geometria "fixa" sólidos inválidos, conforme analisados segundo os critérios estabelecidos no capítulo 3. Em outras palavras, sólidos topologicamente válidos não são, necessariamente, válidos geometricamente [25].

Na figura 5.11, encontra-se um bom exemplo para o problema da validade geométrica. No caso (b), o *Vértice v6* não

faz parte do *Ciclo* externo l_1 , mas sim do *Ciclo* interno l_2 . Já no caso (a), o problema é ainda mais crítico, pois o *Vértice* v_5 não divide a *Aresta* e_1 . Nota-se que, em ambos os casos, o problema se resume, simplesmente, à indevida coordenada geométrica dos *Vértices* v_6 e v_5 .

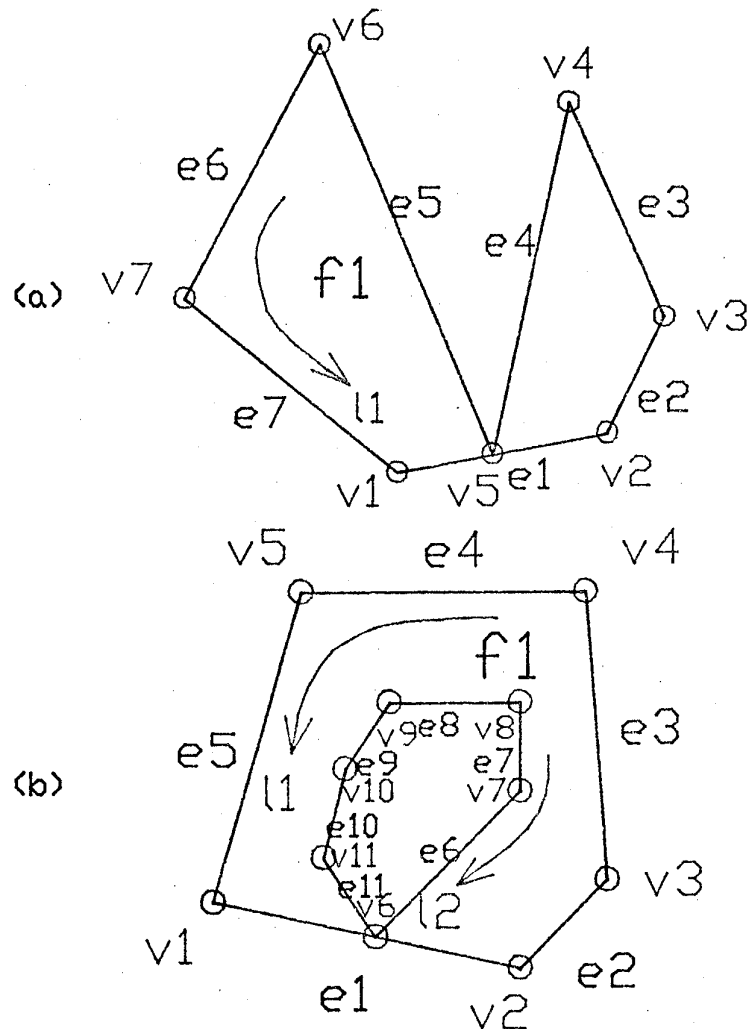


Figura 5.11 - Faces geometricamente inválidas

Uma extensão tridimensional do problema bidimensional acima encontra-se ilustrada na figura 5.12. Mais uma vez, a invalidade geométrica está sendo gerada por uma coordenada geométrica imprópria, atribuída a um *Vértice* (neste caso, o *Vértice v9*).

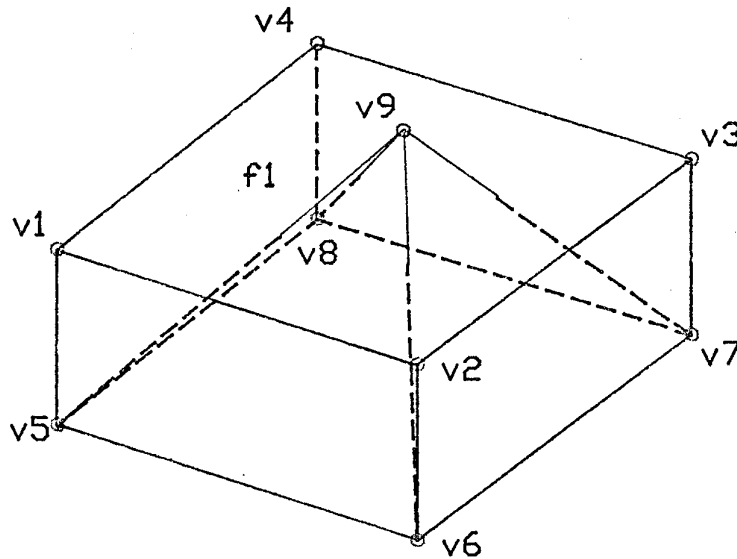


Figura 5.12 - *Vértice v9* geometricamente inválido

De uma forma geral, os problemas geométricos expostos acima podem ser ainda mais complexos e difíceis de serem detectados e evitados. Tal fato ocorre, principalmente, quando se permite a modelagem de superfícies curvas. Se as *Faces* curvas forem geradas, por exemplo, através de superfícies de

Bezier, é fácil se imaginar a situação, em que o deslocamento indevido de alguns pontos de controle, geraria inconsistências ainda maiores.

Em razão desse alto grau de complexidade, é que grande parte dos SMS's, hoje em utilização, fazem uso de aproximações por poliedros regulares. Nestes, não apenas a representação interna é bastante simplificada, como também o são os processos de se assegurar a integridade geométrica dos mesmos.

É importante que seja feita uma clara distinção entre os SMS's (baseados na BREP) que oferecem funcionalidades para a geração de faces curvas, mas que, no entanto, as representam por intermédio de aproximações por faces planas, e os sistemas que realmente implementam superfícies esculturais (*Sculptured Surfaces* - SSRF). No primeiro caso, o que se tem é, na verdade, um modelador de poliedros regulares que, no máximo, oferece recursos para que se especifique a granularidade da discretização, ou seja, o usuário poderia requisitar que uma *Face* cilíndrica, por exemplo, fosse aproximada através de *n* *Faces* planas. Essa discretização forçada pode ser irrelevante para algumas aplicações, desde que os usuários estejam, é claro, cientes disso.

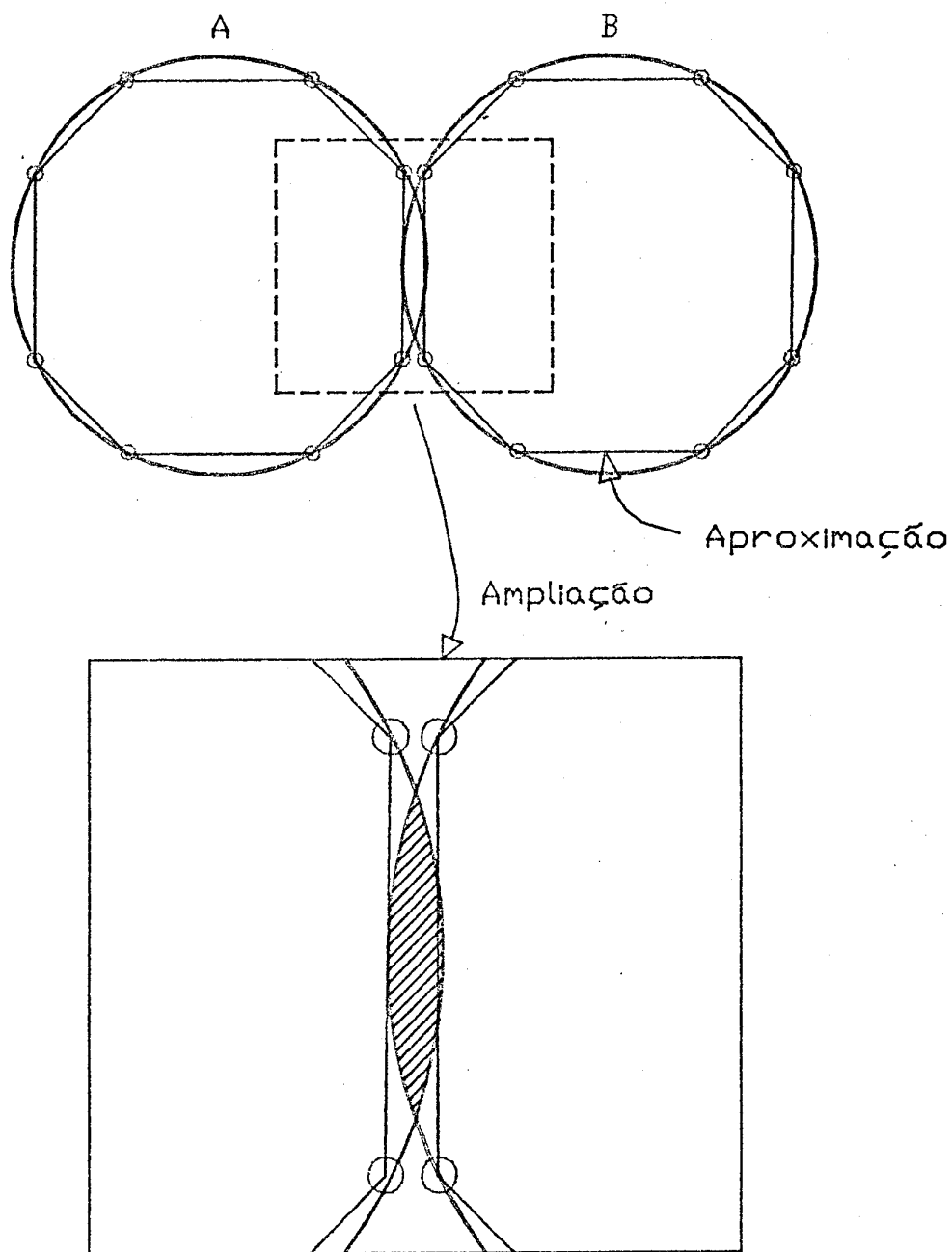


Figura 5.13 - Perda de informação durante a discretização

Operações booleanas podem ser consideradas como as principais vítimas dessas aproximações. Durante uma operação de intersecção de dois sólidos, por exemplo, o sistema pode responder ao usuário com uma mensagem do tipo: "Intersecção Nula" (figura 5.13). Na verdade, os objetos estavam posicionados corretamente, mas, durante a sua discretização, houve uma perda irreparável de informação [21,26,33,43,44,45].

Muitos trabalhos teóricos têm sido dedicados à solução dos problemas de integridade geométrica. Na prática, no entanto, a solução mais comumente adotada consiste, simplesmente, em se garantir que, durante a própria modelagem, inconsistências não sejam geradas. O problema da figura 5.12 poderia ter sido evitado se o usuário tivesse sido simplesmente inibido de cruzar o plano da *Face* f_1 . Da mesma maneira, outras restrições análogas poderiam ser inseridas nos próprios processos de modelagem, tornando a representação robusta e sensível a inconsistências. Já no caso da figura 5.13, a única saída correta seria a rerepresentação dos dois sólidos por equações que expressassem exatamente as superfícies em questão.

5.3.3 - Estruturas de Dados

Dentro do contexto de SMS's, uma estrutura de dados pode ser vista como uma realização prática de uma particular forma de representação. Em virtude da grande importância que a

BREP possui para a área de modelagem geométrica, serão expostas algumas das mais importantes estruturas de dados, conhecidas atualmente. Para tanto, é necessário oferecer, inicialmente, um pequeno embasamento teórico sobre alguns conceitos relacionados com a teoria dos grafos.

5.3.3.1 - Diagrama de Schlegel

A Teoria dos Grafos possui uma forte e intrínseca ligação com a BREP. Como já foi exposto, são as entidades topológicas da BREP, que lhe dão consistência e oferecem os meios para que as relações de adjacência sejam levantadas. Além disso, mencionou-se o fato de que é, através dessas entidades topológicas, que as informações geométricas são associadas. Sendo assim, quando se deseja "visualizar" um determinado objeto representado numa BREP, o meio, mais intuitivamente utilizado, é o de se gerar um mapeamento abstrato das suas entidades topológicas sob a forma de um grafo planar.

Surge, no entanto, a dúvida de como é possível se imaginar um objeto tridimensional mapeado para um grafo planar. Primeiramente, é importante notar que não é o objeto propriamente dito que irá ser mapeado, mas sim as suas entidades topológicas, ou seja, *Faces*, *Arestas*, etc. A idéia básica consiste em se "embutir" um grafo sobre a superfície de um objeto (*Embedded Boundary Graph - EBG*), de maneira que os

Nós e *Arcos* desse grafo, coincidam, respectivamente, com os *Vértices* e *Arestas* da sua representação topológica. *Faces* serão mapeadas, dessa forma, para regiões fechadas do plano, limitadas por *Arcos* e *Nós*.

O formato desse grafo é conhecido como um *Diagrama de Schlegel*. Um *Diagrama de Schlegel* é uma projeção das *Faces*, *Arestas* e *Vértices* de um objeto, quando observado de um ponto bem próximo à sua superfície, apresentando, por conseguinte, um efeito perspectivo bem alto.

Da mesma forma que um objeto não pode possuir *Arestas* que se interceptam, senão em *Vértices* comuns, o mesmo se aplica a *Arcos* e *Nós* em um *Diagrama de Schlegel*. Na figura 5.14, pode-se observar um exemplo clássico de um *Diagrama de Schlegel* para um cubo.

Por conveniência, utilizar-se-á, futuramente, os termos *Aresta* e *Vértice* para denotar os seus equivalentes no grafo, isso é, *Arcos* e *Nós* respectivamente. Essa analogia não traz nenhuma descaracterização dos seus significados enquanto elementos de um grafo.

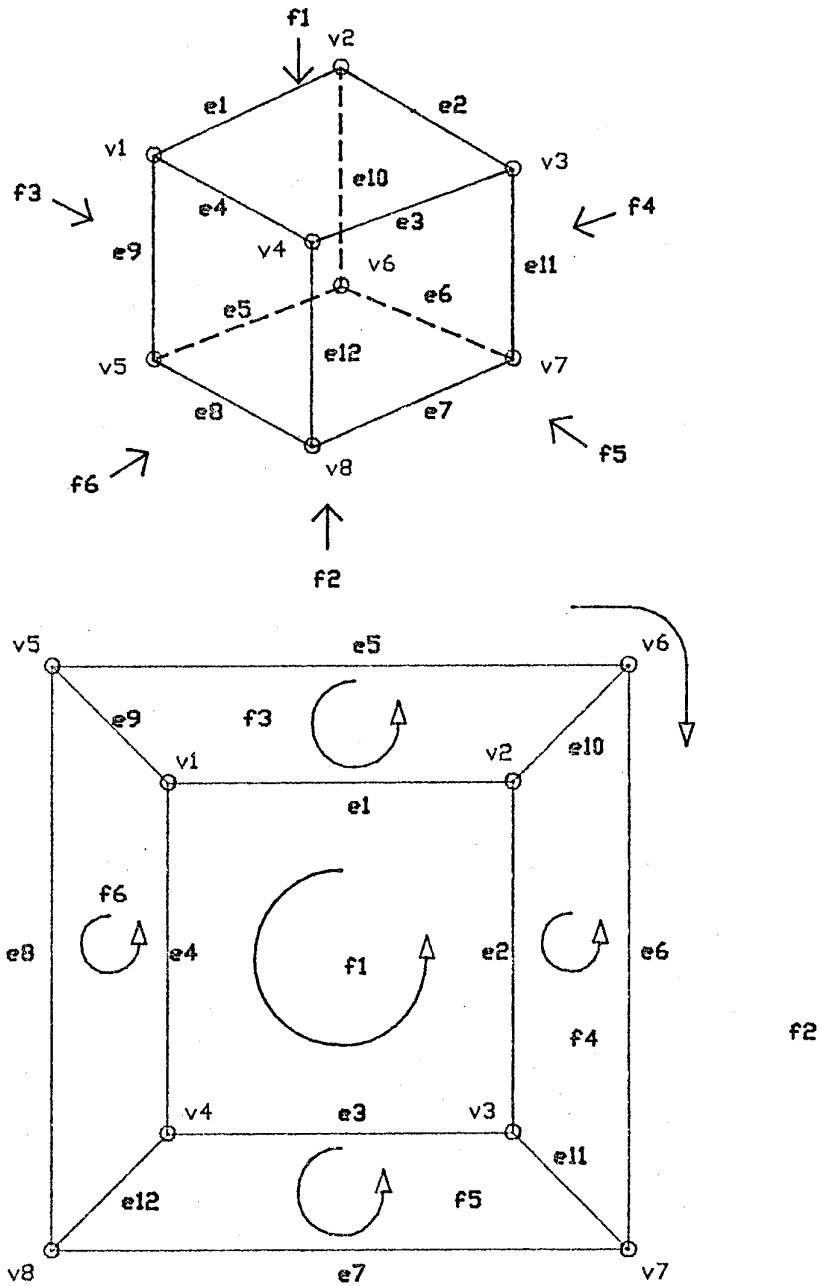


Figura 5.14 - Diagrama de Schlegel para um cubo

Uma primeira dúvida, que ocorre quando se observa um *Diagrama de Schlegel*, é o fato de se ter a impressão de que uma das *Faces* do objeto descrito tem limites infinitos. Tal fato, no entanto, não ocorre. Como foi visto anteriormente, um cubo é *homeomorfo* a uma esfera. Se for considerado, no presente momento, apenas um pequeno pedaço desta esfera e se esta for transformada em plana, observar-se-á exatamente o *Diagrama de Schlegel* desse cubo, no qual a *Face* cortada é fechada no infinito. No exemplo da figura 5.14, pode-se constatar tal fato com relação à *Face* f_2 . Essa abstração é bastante intuitiva e requer, normalmente, um pouco de familiaridade com o processo.

No caso dos objetos mapeados possuírem *Genus* superior a zero, isto é, objetos cujos grafos de fronteira não são planares, utiliza-se, freqüentemente, um mecanismo adicional para representá-los num pedaço plano de papel, conhecido como *Identificadores de Vazios (Voids Labels, [48])*. *Identificadores de Vazios* denotam, de uma forma única, regiões do grafo que sempre ocorrem em pares. Procura-se, através destes, conectar porções de um grafo, estabelecendo conceitualmente uma "ponte" sobre o plano do papel. Dessa maneira, todas as *Arestas* e *Vértices*, que rodeiam um *Vazio*, ocorrem duas vezes no *Diagrama de Schlegel* e são utilizados em direções opostas pelas *Faces* adjacentes ao mesmo. *Faces*, que são adjacentes a uma *Aresta* na fronteira de um *Vazio*, são, na verdade, adjacentes a si próprias, podendo ser imaginadas como estando "coladas" uma na

outra (figura 5.15).

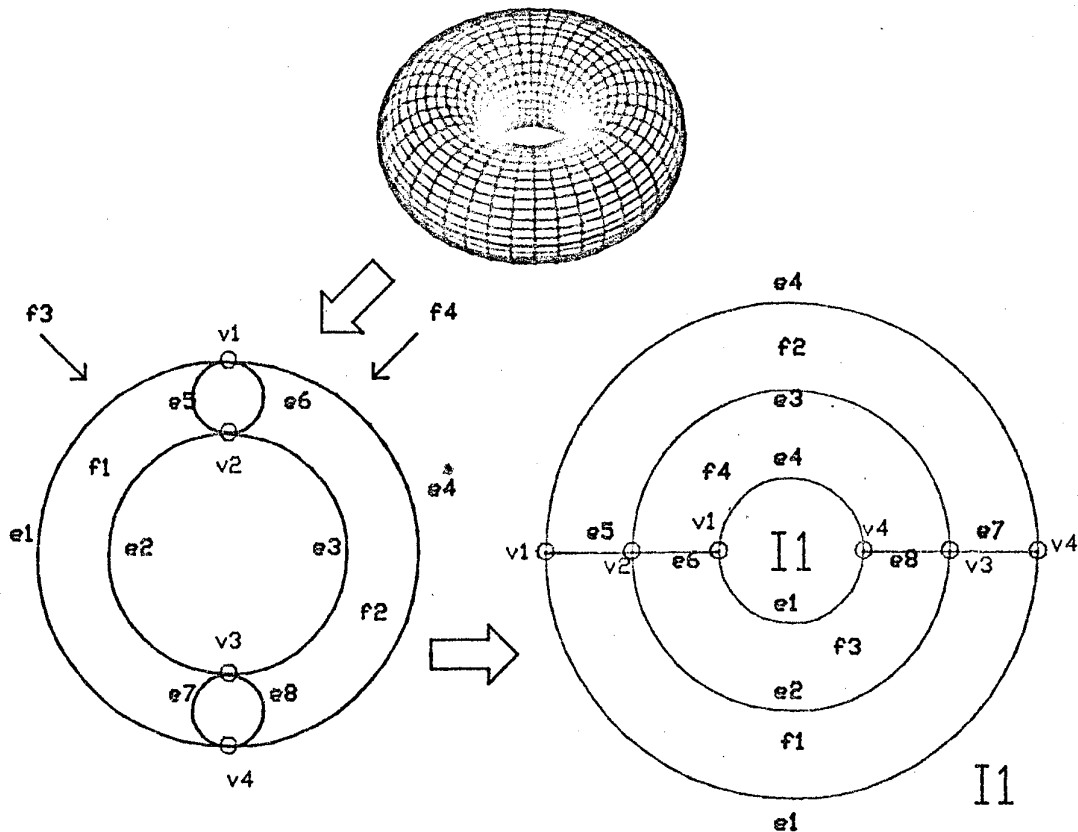


Figura 5.15 - Diagrama de Schlegel para um objeto com Genus 1

É importante notar que *Identificadores de Vazios* não possuem nenhuma ligação com entidades topológicas e são utilizados apenas como uma maneira conveniente e simples de se gerarem diagramas planares de objetos com *Genus* superior a zero. O número de pares de *Vazios* necessários num Diagrama de

Schlegel é igual ao *Genus* do objeto mapeado.

A entidade topológica *Casca* foi utilizada para aglutinar, em si, a representação completa da fronteira de um sólido. No caso desse sólido possuir mais de uma *Casca*, ou seja, conter regiões interiores disjuntas, o seu mapeamento para um *Diagrama de Schlegel* deve ser feito de forma diferente. Cada *Casca* deverá ser descrita por um diagrama exclusivo e posicionada uma em cima da outra, como se fossem folhas de papel sobrepostas numa mesa. O primeiro diagrama dessa "pilha" estará caracterizando a *Casca-Pai* (*Pattern-Shell*), e os restantes mapearão as *Cascas-Filhas* (*Child-Shell's*). Como as *Cascas* são totalmente disjuntas no sólido real, as suas representações mapeadas também não poderão conter nenhum tipo de ligação, a não ser o simples conceito de hierarquia. Formalmente, o mapeamento de objetos com mais de uma *Casca* resulta em grafos desconexos.

Uma pergunta lógica, que se faz nesse momento, é qual o sentido prático de se mapear a representação topológica de um objeto para um grafo planar. As principais justificativas para tal questionamento são as seguintes:

- (i) A "visualização topológica" de um objeto é muito mais simples quando este se encontra totalmente mapeado para um grafo de *Schlegel*. Possíveis ambigüidades,

geradas por más interpretações, são totalmente removidas através desse processo, oferecendo-se uma consistente base para análises topológicas;

- (ii) Os levantamentos das relações de adjacência de um sólido são realizados a partir do percorrimto da sua topologia. O fato do sólido ser mapeado para um *Diagrama de Schlegel* facilita bastante este percorrimto, em razão da existência de um completo embasamento teórico a respeito de estruturas em grafo. De uma forma geral, pode-se afirmar que uma relação só pode ser verificada, caso se possa percorrer esse grafo de uma forma completa e não ambígua;

- (iii) As principais estruturas de dados conhecidas são caracterizadas pelo número e tipo de informações de adjacência que guardam diretamente. Sendo assim, é comum expressar-se uma determinada estrutura de dados como um pedaço hipotético de um *Diagrama de Schlegel*, demonstrando-se as suas limitações para percorrê-lo e, conseqüentemente, os seus problemas para a solução de questões de adjacência.

5.3.3.2 - Estrutura Básica

Como já foi dito anteriormente, para que se possam

levantar as relações de adjacência de um objeto, deve-se percorrer o seu grafo planar. Segundo os conceitos descritos na teoria dos grafos, tal processo é realizado, "visitando-se" *Nós* vizinhos por meio dos *Arcos* que os ligam (supondo-se um grafo não orientado). Fazendo uma analogia com o grafo gerado em um *Diagrama de Schlegel*, esta tarefa resumir-se-ia em percorrer todas as *Arestas*, passando pelos seus respectivos *Vértices*.

Por ser a entidade *Aresta* o elemento de elo nesse "caminhamento", é comum atribuir-lhe a função de servir como ponto de referência para a determinação das relações de adjacência com outros elementos. Notoriamente, as principais estruturas de dados são conhecidas por serem, todas, baseadas na entidade *Aresta*, no sentido de que toda a informação necessária para a reprodução e "percorrimto" do grafo planar está contida em estruturas relacionadas com essa entidade.

Sendo as *Arestas* o ponto central de classificação e avaliação das estruturas, definir-se-á uma estrutura geral básica, comum e independente da sub-estruturada *Aresta*. Esta estrutura maior tem como objetivo representar as outras entidades topológicas (*Casca*, *Face*, *Ciclo*, e *Vértice*) e oferecer uma hierarquia consistente para a representação de um objeto numa BREP.

```

typedef struct _sólido *PTR_SÓLIDO;
typedef struct _casca *PTR_CASCA;
typedef struct _face *PTR_FACE;
typedef struct _ciclo *PTR_CICLO;
typedef struct _aresta *PTR_ARESTA;
typedef struct _vértice *PTR_VÉRTICE;
typedef float COORDENADA;

typedef struct _sólido
<
    PTR_CASCA lista_cascas;
> SÓLIDO;

typedef struct _casca
<
    PTR_CASCA última_casca,
    próxima_casca;
    PTR_FACE lista_faces;
> CASCA;

typedef struct _face
<
    PTR_FACE última_face,
    próxima_face;
    PTR_CICLO lista_ciclos;
> FACE;

typedef struct _ciclo
<
    PTR_CICLO último_ciclo,
    próximo_ciclo;
    PTR_ARESTA lista_arestas;
> CICLO;

typedef struct _aresta
<
    /* A ser ainda definida ! */
> ARESTA;

typedef struct _vértice
<
    COORDENADA x, y, z;
> VÉRTICE;

```

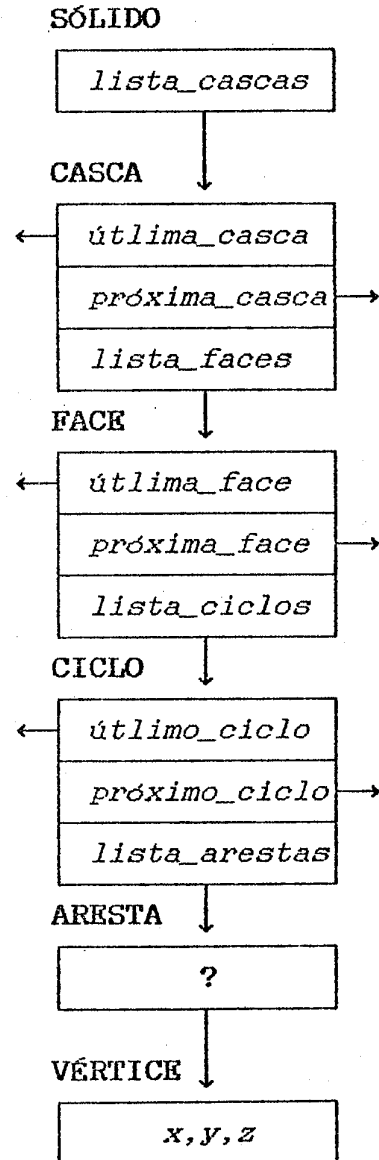


Figura 5.16 - Descrição em C da estrutura de dados básica

A figura 5.16 contém uma descrição completa de tal estrutura básica (implementada na linguagem de programação C), bem como uma descrição gráfica do seu espaço de alocação. Nota-se a inserção de uma nova entidade denominada de SÓLIDO, que oferece, simplesmente, o acesso hierárquico a toda estrutura. É convencional, que todos os tipos declarados são descritos por letras maiúsculas e precedidos por "PTR_", quando denotam ponteiros para outros tipos ou sub-estruturas.

Nessa estrutura básica, pode-se observar, também, que *Cascas* e *Ciclos* são implementados como listas duplamente encadeadas, ou seja, permitem acesso seqüencial nas duas direções. É convencional que a primeira *Casca* na lista de cascas de um sólido indicará a sua *Casca-Pai* (mais externa), e as seguintes serão as suas *Cascas-Filhas* (todas internas a *Casca-Pai*). Analogamente, observa-se estar o primeiro *Ciclo* de uma *Face* representando a sua fronteira mais externa (*Ciclo-Pai*) e os restantes, os seus furos internos (*Ciclos-Filhos - Anéis*).

Destaca-se o fato de, nessa estrutura básica, não existir nenhuma preocupação em se armazenar entidades geométricas, a não ser, é claro, as coordenadas dos *Vértices*. Numa implementação real dessa estrutura, outras informações geométricas e topológicas costumam ser inseridas de maneira a diminuir o número de acessos e, conseqüentemente, aumentar a

performance de várias consultas. Apenas como exemplo, poder-se-iam inserir no registro **CASCA** ponteiros para listas de *Vértices*, *Faces*, ou *Arestas*. Analogamente, poderiam ser associadas informações de geometria, como, por exemplo, equações de curvas e superfícies nos registros **ARESTA** e **FACE**. Por fim, poder-se-iam, ainda, inserir ponteiros de retorno (*Backpointers*), ou seja, informações para o "caminhamento" de baixo para cima.

Todo o acesso a entidades topológicas nessa estrutura pressupõe um acesso hierárquico do tipo *Top-Down*. Tal forma de organização tem-se mostrado bastante eficaz, pois é capaz de eliminar significativamente muitos acessos desnecessários, permitindo, em consequência, uma maior eficiência do **SMS**. Se, por exemplo, fosse armazenada no registro **CASCA** a descrição geométrica da esfera de raio mínimo, que engloba totalmente uma casca, seria possível, com poucos acessos, determinar se dois sólidos potencialmente se interceptam ou não.

Das quatro estruturas a serem apresentadas, duas delas (*Winged-Edge* e a *Winged-Edge-Modificada*) mantêm a *Aresta* como uma unidade única, enquanto que as outras duas (*Vértice-Aresta* e *Ciclo-Aresta*) a dividem em duas partes separadas. Estas últimas possuem, também, uma alocação de dados igual, mas diferem bastante no seu significado semântico.

5.3.3.3 - Estrutura Winged-Edge

A estrutura *Winged-Edge* (WE) representa uma *Aresta* e as suas relações de adjacência como uma unidade básica, utilizando uma estrutura de tamanho fixo¹.

Essa estrutura foi seguramente a primeira proposta prática, para se representar a topologia de objetos sólidos. Desenvolvida por *Bruce G. Baumgart* [6,7,47] em 1975, a WE tinha como objetivo primordial a representação de poliedros sólidos na área de robótica, em particular, para o reconhecimento de formas em técnicas de Visualização Computacional (*Computer Vision - CVIS*). Em SMS's, essa estrutura tem sido bastante utilizada para representar as relações topológicas de adjacência, contidas no grafo planar "embutido" sobre a superfície de modelos sólidos.

É importante, também, mencionar o fato de ser constatada, no trabalho de *Baumgart*, a primeira proposta de utilização dos EOP como uma forma, em "alto nível", de se criar e modificar o grafo planar, independentemente da estrutura de dados implementada [10].

¹ Todas as quatro estruturas a serem apresentadas gozam da propriedade de possuírem uma alocação individual fixa.

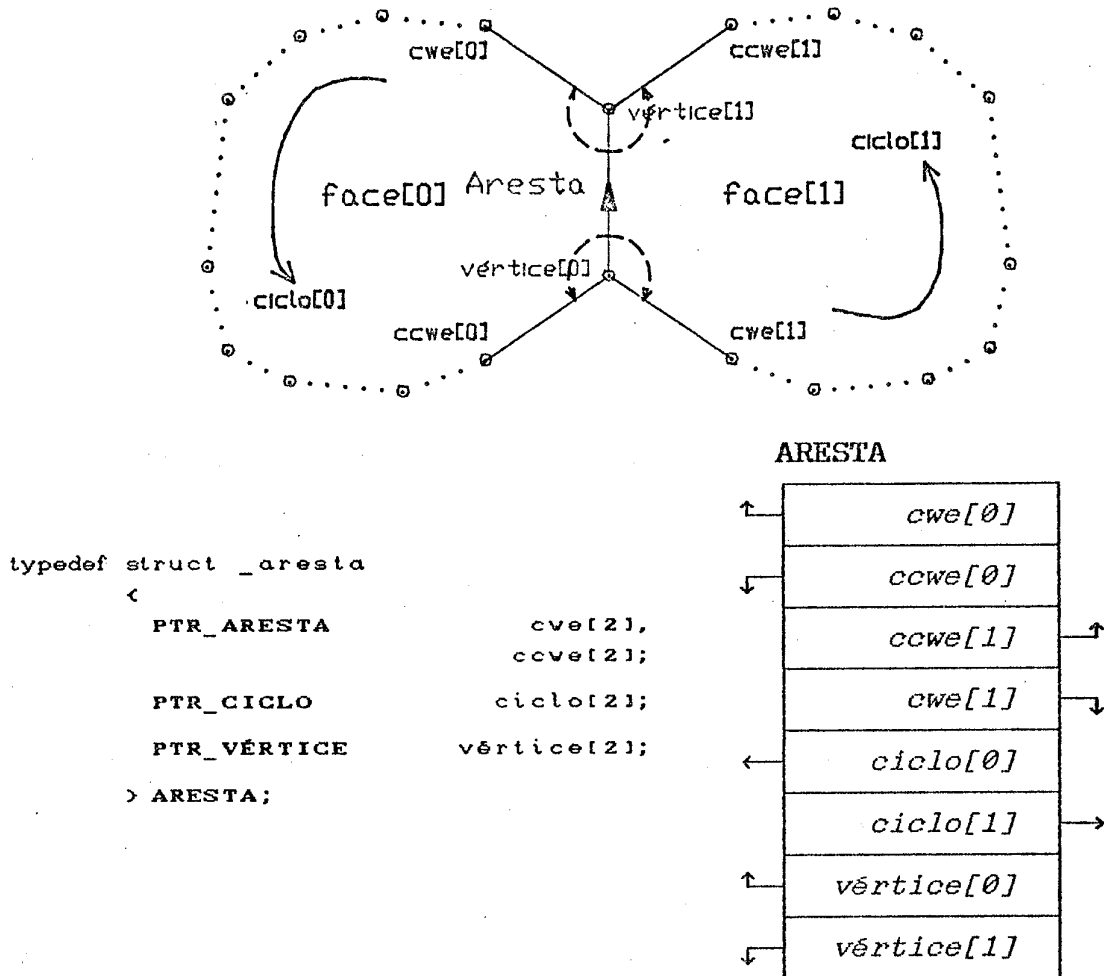


Figura 5.17 - Estrutura *Winged-Edge*

A informação topológica, contida na *WE* para cada *Aresta*, é composta da sua relação de adjacência com outras *Arestas*, *Ciclos*, e *Vértices*. A idéia básica consiste em se retirar do grafo planar uma determinada *Aresta*, que se deseja

analisar, e anotar, neste processo, todas as entidades vizinhas à mesma. Se isso for realizado, obtém-se, para cada *Aresta*, a informação de adjacência dos seus dois *Vértices* (*vértice[0]* e *vértice[1]*) das quatro *Arestas* (*cwe[0]*, *ccwe[0]*, *cwe[1]* e *ccwe[1]*), é claro, dos seus dois *Ciclos* adjacentes (*ciclo[0]* e *ciclo[1]*). Ver figura 5.17.

O nome *Winged-Edge*, atribuído a essa estrutura, advém da sua semelhança com uma asa. Para se explicar a nomenclatura utilizada na identificação das *Arestas* vizinhas à *Aresta* representada, deve-se descrever o mecanismo de associação das adjacências, isso é, a maneira com que a estrutura, propriamente dita, é construída e percorrida.

Como foi visto, cada fronteira, possuída por uma *Face*, deverá ser representada por um *Ciclo* de *Arestas*. Quando se percorre esse *Ciclo*, caminhando-se sobre o grafo planar correspondente, deve-se considerar como próxima *Aresta* aquela, cuja orientação, relativa à *Aresta* e ao *Ciclo* correntes, descreve um arco no sentido horário (*Clockwise Edge - cwe*). Dessa forma pode-se "visitar" todos os *Vértices* do respectivo *Ciclo*. Tal "caminhamento" é equivalente ao sentido gerado pelo produto vetorial dos vetores diretores de duas *Arestas* adjacentes, quando percorridas pelo lado externo do sólido.

No caso de se estar percorrendo um *Ciclo* interno de uma *Face* (*Ciclo-Filho*), tal orientação é simplesmente invertida (*Counter Clockwise Edge - ccwe*), de forma a manter-se a *Face* percorrida sempre do lado esquerdo da *Aresta* corrente (figura 5.18).

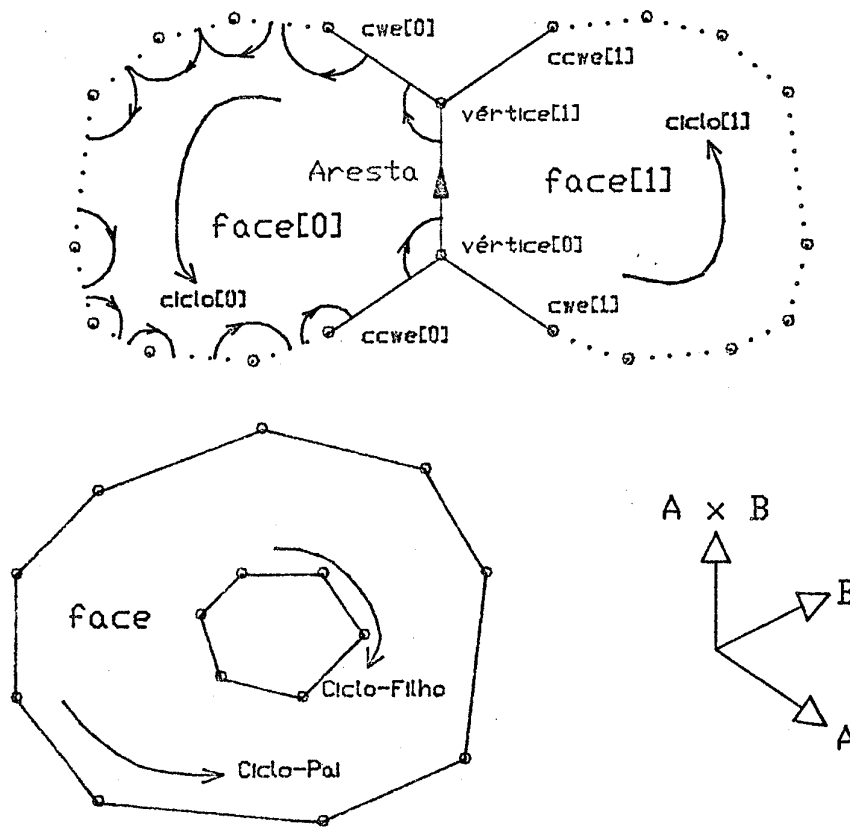


Figura 5.18 - "Percorrimento" de uma *Face* através da *WE*

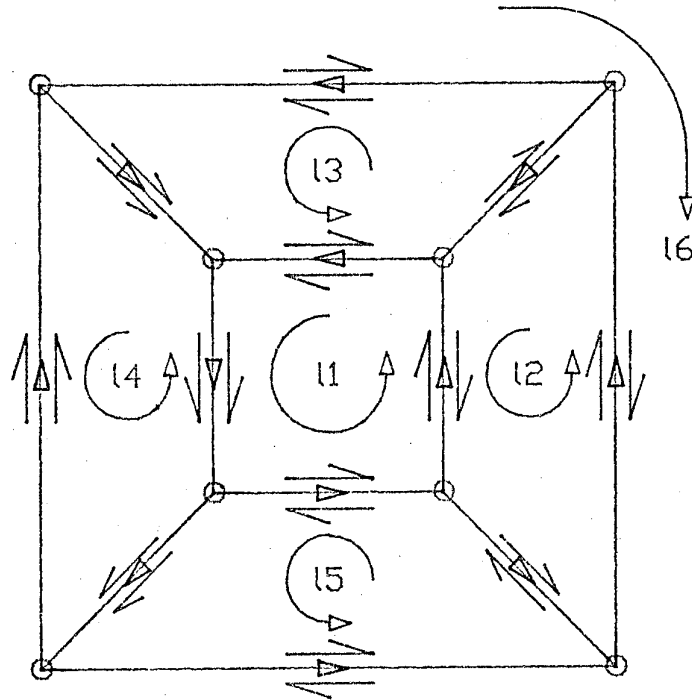


Figura 5.19 - Impossibilidade de se orientarem *Arestas*

Nesse momento, é importante que seja observado o fato de tal "percorrimto" só ser possível, se for fixado, inicialmente, o *Ciclo* desejado, pois é ele que mantém o "caminhamento" sempre consistente. A justificativa para tal fato é bastante simples. Ao se considerar, como exemplo, um cubo e, ao se tentar orientar as suas *Arestas* de forma a estas coincidirem com o sentido de cada *Ciclo*, verifica-se que tal ocorrência não é possível, existindo, sempre, *Arestas* que não podem ser orientadas num sentido único. Em razão disso é que se

afirma que o grafo planar, gerado pelo *Diagrama de Schlegel*, é não orientável (figura 5.19).

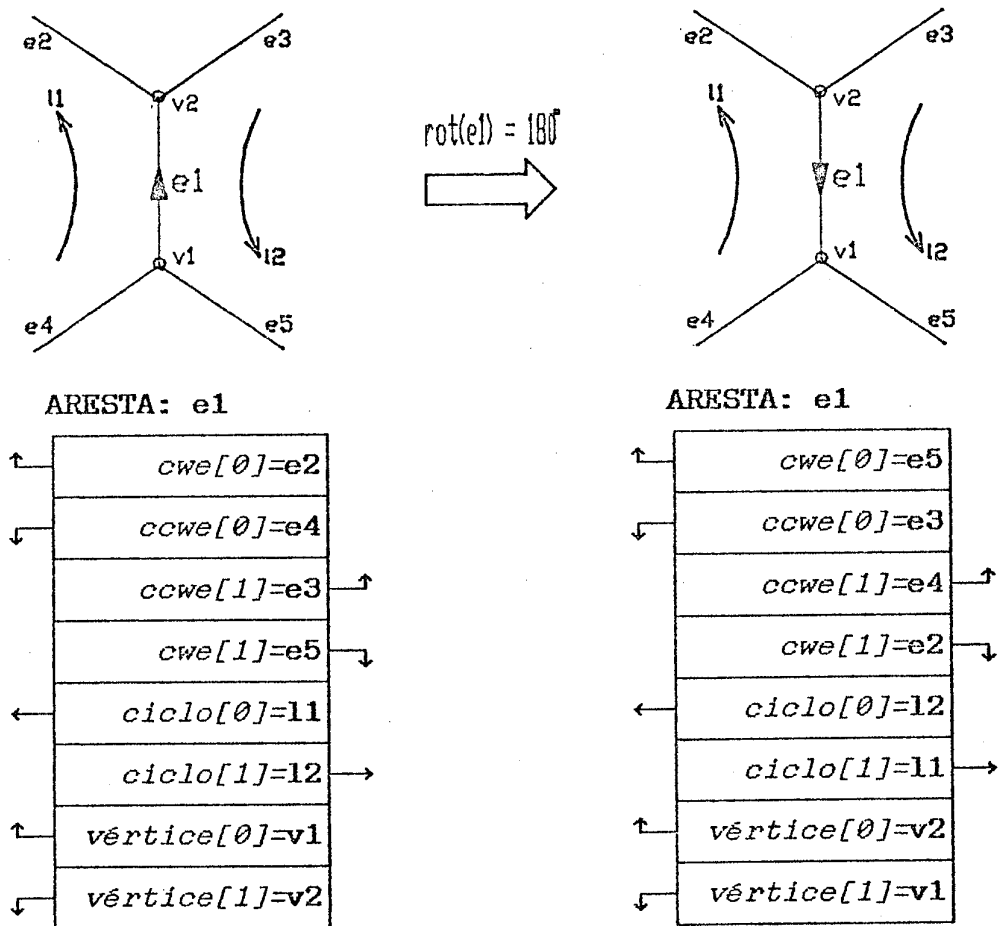


Figura 5.20 - Dupla representação da entidade *Aresta*

O mesmo problema observado durante o "percorrimento" de um *Ciclo*, ou seja, durante a identificação das *Arestas*

seguintes, ocorre quando se deseja saber qual o *Vértice* inicial e final de um *Aresta*. A própria utilização dos termos "inicial" e "final" não faz, aqui, nenhum sentido, quando se considera a *Aresta* uma entidade não orientável. Mais uma vez é necessário que se especifique um *Ciclo* ou uma outra *Aresta* como referência para que a presente pesquisa possa ser feita.

Ao se retirar uma *Aresta* qualquer do grafo e ao se efetuar sua rotação em 180° , reposicionando-a em seguida, obtêm-se as informações de adjacência simplesmente espelhadas (figura 5.20). Analogamente, não é possível a orientação das *Faces* de uma *Casca* de uma maneira padronizada.

O aparente conflito que pode ter sido gerado na rápida observação das figuras, nas quais a entidade *Aresta* é apresentada com uma seta direcionando-a, deve ser bem explicado. Estas setas têm, como objetivo único, mostrar que a referida *Aresta* está sendo mapeada internamente para a estrutura *ARESTA*, com tal orientação, ou seja, das duas possíveis orientações que poderia possuir, escolheu-se esta, como a que seria efetivamente utilizada. É importante lembrar ainda, que não existe nenhuma prioridade para a escolha de tal orientação, visto que ambas são totalmente equivalentes e não possuem nenhuma associação intrínseca com a direção, propriamente dita, dos ciclos em questão.

A proposta inicial, apresentada por *Baumgart*, possuía uma sutil diferença com relação a WE aqui apresentada. Naquela, a entidade *Vértice* era tomada como ponto de referência para a identificação das *Arestas* adjacentes. Além disso, em tal estrutura, não havia sido incorporada a entidade *Ciclo*, tendo esta sido substituída diretamente por ponteiros para as *Faces* vizinhas, impossibilitando, conseqüentemente, a representação de furos.

5.3.3.4 - Estrutura Winged-Edge-Modificada

Os problemas de "percorrimto", verificados na seção anterior, originaram várias propostas de solução. Observou-se, no entanto, que uma solução direta não é possível, pois, como já foi visto, não se pode organizar uma estrutura topológica baseada numa orientação única para a entidade *Aresta*.

Durante o "percorrimto" de uma *Face* na WE, é necessário que se teste sempre a orientação da *Aresta* consultada, com a do *Ciclo* de referência (*Ciclo-Pai* da *Face* em questão). Caso estes não coincidam, deve-se tomar os valores espelhados, ou seja, inverter o índice que acompanha cada nome de registro. Por exemplo, se for desejado encontrar a *cwe[0]* da *Aresta* corrente, e o *Ciclo* de referência não for o *ciclo[0]*, mas sim o *ciclo[1]*, então deve-se considerar a *cwe[1]*, pois a *Aresta* corrente se encontra invertida com relação ao *Ciclo* de

referência. Em resumo, em qualquer pesquisa, deve-se testar, primeiramente, a orientação da *Aresta* corrente com a do *Ciclo* sendo "percorrido", para depois poder-se selecionar a informação desejada.

Imaginando-se agora uma aplicação de visualização tridimensional, que deseje percorrer todas as *Faces* de um objeto com, por exemplo, 1000 *Faces* (todas representadas através de apenas um *Ciclo* de 4 *Arestas* e 4 *Vértices*), seriam necessárias 4000 comparações, para se levantar a relação de adjacência, $f:\{e\}$, de todo este sólido. Sendo assim, devido à motivação oriunda das necessidades das aplicações, desenvolveu-se uma variante da *WE*, denominada de estrutura *Winged-Edge* modificada (*mWE*). O principal objetivo desta variante é a de evitar as exaustivas comparações necessárias durante o "caminhamento" de uma *Face* na *WE* [47].

Como nos principais aspectos de Sistemas Gerenciadores de Bancos de Dados (*Database Management Systems - DBMS*), a única solução possível, para se evitarem constantes pesquisas, é a de se guardar, explicitamente, a informação desejada. A técnica utilizada na *mWE* não é diferente e baseia-se num simples aumento da informação contida originalmente na sua precursora.

Em tal abordagem, inserem-se dois registros adicionais

a estrutura **ARESTA**, denominados *lados_cwe[]* e *lados_ccwe[]*. Estes vetores de duas posições conterão quatro índices, para a especificação da orientação das *Arestas* seguintes (para os dois *Ciclos*) a ser considerada durante o seu caminhamento (figura 5.21).

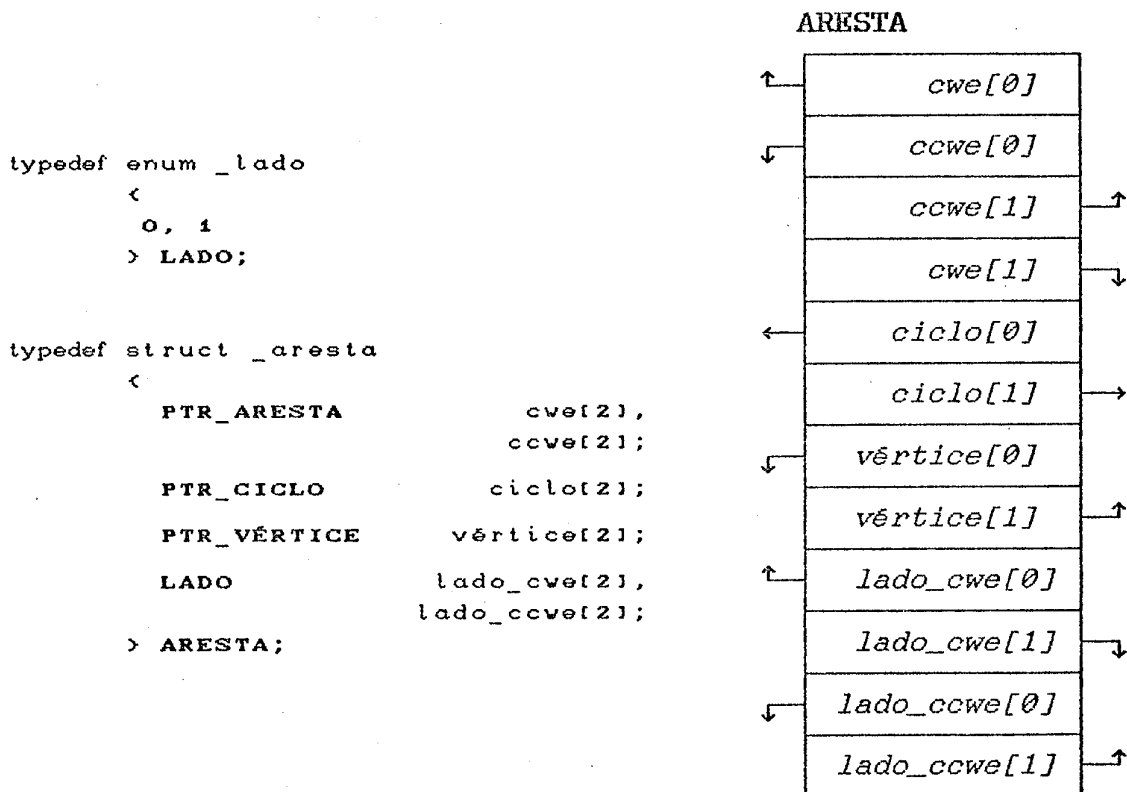


Figura 5.21 - Estrutura *Winged-Edge* modificada

Se, por exemplo, o registro *lado_cwe[0]* contiver um

índice "0", isso indicará que, ao tomar-se a *Aresta cwe[0]*, esta deverá ser indexada por este índice, de tal forma, a manter a orientação do *Ciclo* percorrido consistente. Observa-se, como resultado de tal prática, o fato de que, ao se percorrer uma *Face*, será apenas necessária uma comparação inicial para a especificação da orientação da primeira *Aresta* percorrida. As *Arestas* seguintes irão ser indexadas através dos índices armazenados nas suas anteriores (já percorridas).

Para esta estrutura, um "caminhamento" circular horário (*Clockwise - cw*) sobre um *Ciclo-Pai* exigiria, inicialmente, a identificação de qual lado da *Aresta* inicial (*e*) se deseja percorrer (índice $i_e = 0$ ou 1). Tal identificação é feita simplesmente comparando-se o *Ciclo* de referência com o *ciclo[0]* da *Aresta* inicial, da mesma forma que isto é feito na WE^1 . Se ambos coincidirem, tem-se:

$$i_e = 0 .$$

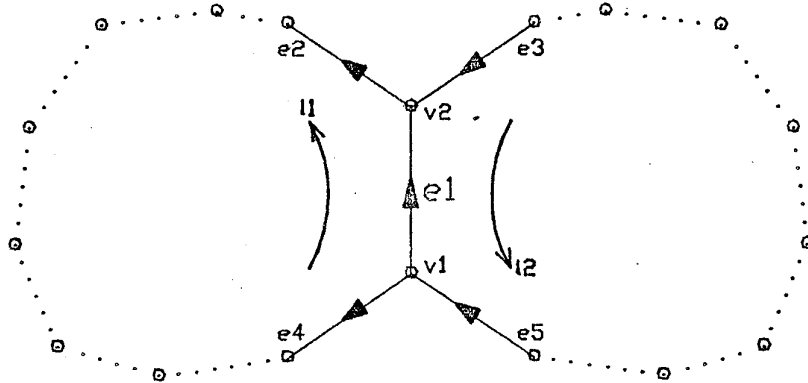
Caso contrário,

$$i_e = 1 .$$

Em seguida à escolha deste índice, indexa-se o vetor *lado_cwe[]*, obtendo-se, então, o índice a ser utilizado no "percorrimto" da próxima *Aresta*, ou seja:

¹ O *Ciclo* de referência será sempre aquele, no qual se deseja "caminhar" dentro de uma *Face*.

$$i_{\text{próxima_aresta}} = e \rightarrow \text{lado_cwe}[i_e]$$



ARESTA: e1

↑	$cwe[0]=e2$	
↓	$ccwe[0]=e4$	
	$ccwe[1]=e3$	↑
	$cwe[1]=e5$	↓
←	$ciclo[0]=l1$	
	$ciclo[1]=l2$	→
↑	$vértice[0]=v1$	
↓	$vértice[1]=v2$	
↑	$lado_cwe[0]=0$	
	$lado_cwe[1]=0$	↓
↓	$lado_ccwe[0]=1$	
	$lado_ccwe[1]=1$	↑

Figura 5.22 - Aresta representada através da mWE

Um exemplo de uma possível configuração para a representação de uma *Aresta*, através da *mWE* pode ser observado na figura 5.22.

Considerando-se o exemplo do sólido com 1000 *Faces*, apresentado anteriormente, observar-se-ia, através da *mWE*, um aumento no espaço de alocação necessário para representá-lo (em comparação com a *WE*); no entanto, necessitar-se-ia de apenas 1000 comparações para realizar o seu "caminhamento", ou seja, uma redução considerável de aproximadamente 75%.

Tal método obriga o sistema a manter os vetores de indexação (*lado_cwe[]* e *lado_ccwe[]*) sempre corretamente atualizados, oferecendo, em contrapartida, uma razoável simplificação nas operações de "caminhamento". Para algumas aplicações, esta simplificação pode ser bastante desejada, principalmente no âmbito de modelos com *Faces* curvas.

5.3.3.5 - Estrutura Vértice-Aresta

A estrutura de dados *Vértice-Aresta* (*VA*) oferece mais uma opção para o mapeamento da entidade topológica *Aresta* [47]. Nesta nova proposta, a entidade *Aresta* é quebrada em duas partes. O objetivo de tal divisão é permitir que as duas possíveis orientações de uma *Aresta* (com relação aos seus

Ciclos) sejam representadas explicitamente. Conseqüentemente, para cada *Aresta* do objeto considerado, observam-se duas sub-estruturas, cuja orientação acompanha a dos respectivos *Ciclos*. Cada sub-estrutura conterà toda a informação referente ao lado da *Aresta* representada. Armazena-se, também, um registro de elo entre as duas sub-estruturas, a fim de se manter a consistência da entidade básica *Aresta* (figura 5.23).

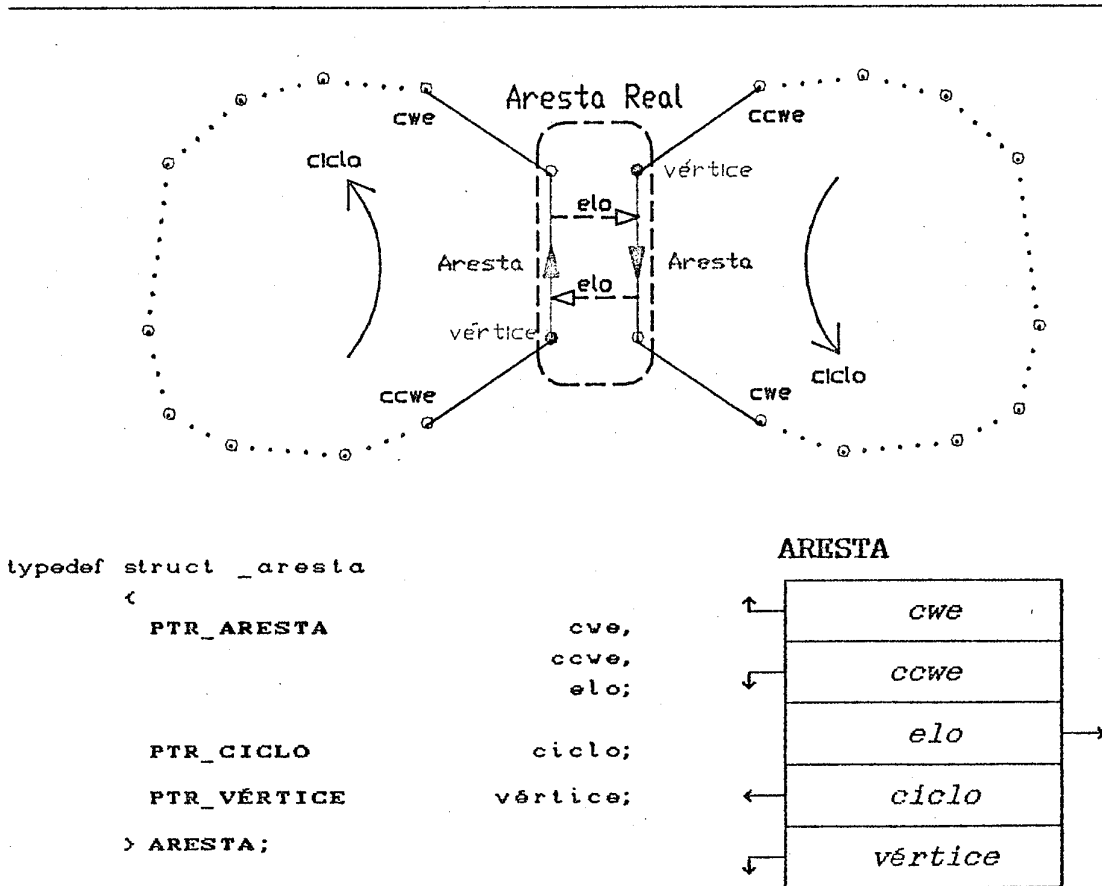


Figura 5.23 - Estrutura *Vértice-Aresta*

O "percorrimento" de um *Ciclo* em tal estrutura pressupõe apenas um simples "caminhamento" circular horário ou anti-horário, dependendo do registro escolhido (*cwe* ou *ccwe*). Caso se deseje percorrer o *Ciclo* complementar da *Aresta* corrente, utiliza-se o registro de elo para "passar-se de um lado para o outro".

Nota-se que a *VA* poderia ter sido elaborada, mantendo-se uma estrutura única para a entidade *Aresta*, como a *WE* e a *mWE* o fazem. Esta proposta, no entanto, não permitiria que, durante o "percorrimento" de um *Ciclo*, se pudesse identificar qual a metade da *Aresta* representada estaria sendo referenciada, a não ser que se fizesse um prévio teste de orientação. Neste caso, observar-se-ia a problemática levantada para as estruturas já mencionadas (*WE* e *mWE*).

5.3.3.6 - Estrutura *Ciclo-Aresta*

A estrutura *Ciclo-Aresta* (*CA*) representa a informação de adjacência de uma *Aresta* através da sua divisão em duas estruturas separadas [47]. Cada uma destas estruturas está relacionada com um dos lados da *Aresta* real, quando observada na periferia dos seus *Ciclos* adjacentes. A sua similaridade com a *VA* é grande, no entanto, conceitualmente, bastante diferente. Na *CA*, armazena-se, para cada *Aresta*, a informação de quem são as suas próximas *Arestas* adjacentes, com relação à orientação

do *Ciclo* corrente (figura 5.24).

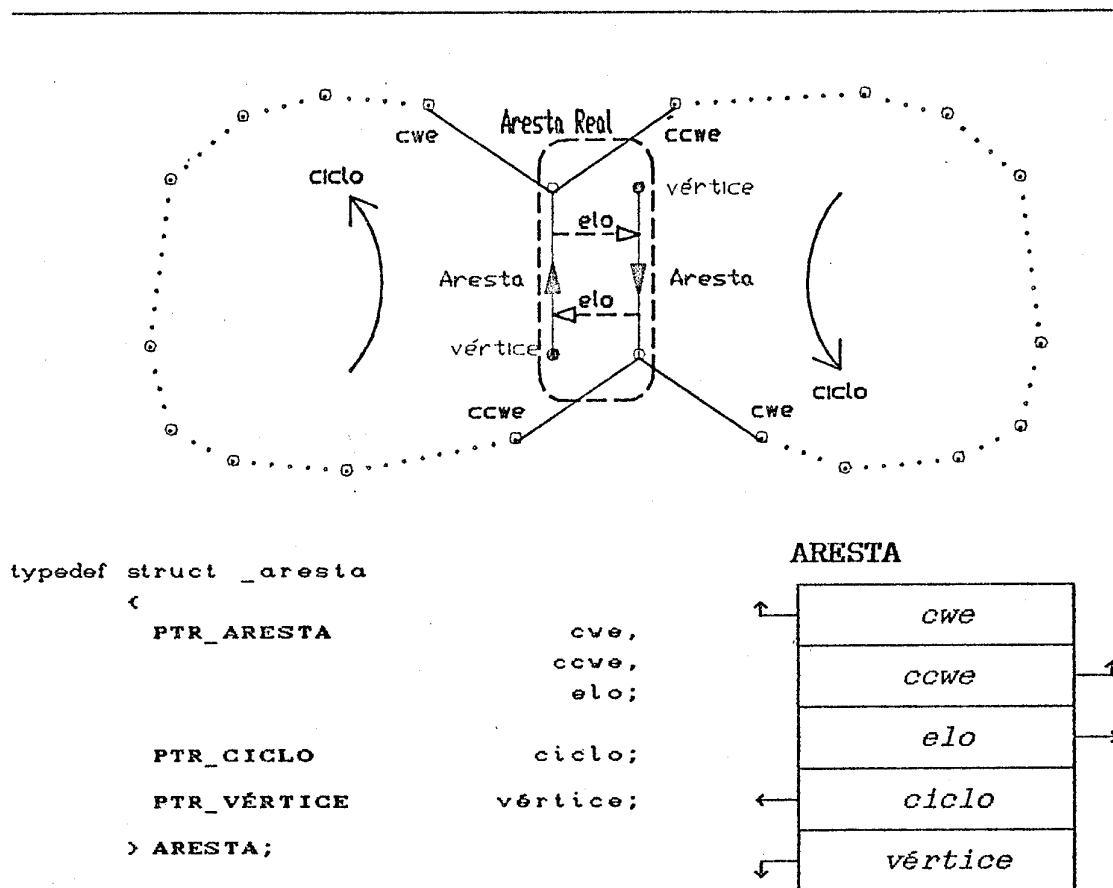


Figura 5.24 - Estrutura *Ciclo-Aresta*

Originalmente, a *CA* era denominada de estrutura *Face-Aresta* (*FA*), mas, com a introdução da entidade *Ciclo*, a referência anterior da *Aresta* para as suas *Faces* adjacentes foi substituída (na *CA*) por ponteiros para os seus respectivos *Ciclos*.

É importante notar que, apesar de diferentes, todas as quatro estruturas propostas são capazes de responder a qualquer questão, relacionada com a obtenção de informações de adjacência entre as suas entidades topológicas. A estrutura VA e a CA possuem algoritmos mais simples para a avaliação das relações de adjacência, mas consomem, em consequência, aproximadamente 20% a mais de espaço físico para a sua alocação, que as estruturas WE e mWE. Dentre estas, a mWE é a mais promissora para sistemas que permitem a representação de *Faces* curvas, pois não exige pesquisas de orientação. A mWE é além disso, a mais utilizada atualmente, existindo, conseqüentemente, consenso de que seja a mais adequada estrutura para a implementação de uma BREP.

5.3.4 - Classificação

A BREP é, atualmente, a forma de representação com o mais vasto domínio de representação. É capaz de englobar, com razoável facilidade, os objetos representados por outras técnicas (EOE, CSG, SWEEP, etc.). As razões para tal afirmação residem, principalmente, nos seguintes fatores:

- (i) Através da BREP, pode-se construir um vasto e poderoso arcabouço de ferramentas (técnicas) para a modelagem (criação e edição) de objetos sólidos. Tal arcabouço inclui operações locais e globais que, nem sempre,

estão disponíveis através de outras formas de representação²;

- (ii) Diferentemente das outras formas de representação, a BREP oferece meios estruturados de se associarem informações não geométricas ao modelo. Este fato possibilita um fácil e organizado interfaceamento entre o SMS e as suas aplicações;

- (iii) Em função da quase ilimitada possibilidade de se introduzirem informações geométricas à forma de representação, permite-se uma adequação das necessidades das aplicações aos recursos do modelador. Em outras palavras, sempre que for necessária a rápida e eficiente representação de um conjunto particular de sólidos, novas técnicas (eventualmente com novos dados geométricos) podem ser desenvolvidas sem comprometer a *performance* do sistema como um todo;

- (iv) Em razão da BREP fornecer uma completa representação topológica dos objetos, possibilita-se, em consequência, o fato de que, durante qualquer estágio da modelagem, a estrutura seja percorrida, obtendo-se importantes informações de "como" o objeto está organizado fisicamente. Pode-se, por exemplo,

² Sistemas, que não utilizam a BREP, oferecem, geralmente, apenas operações locais ou globais, mas não ambas simultaneamente.

identificar se um objeto é internamente sólido, ou se este possui regiões interiores vazias, percorrendo-se simplesmente a sua lista de *Cascas*;

(iv) Notoriamente, todos os SMS's fazem farto uso de recursos gráficos para a interação homem/máquina. Em tais sistemas, a complexidade dos objetos que podem ser modelados, está diretamente relacionada com a capacidade do usuário de poder interagir com a representação gráfica dos sólidos. Neste contexto, a BREP é, sem dúvida alguma, a forma de representação que oferece o maior suporte para as conhecidas técnicas de visualização tridimensional, facilitando, portanto, a implementação e utilização de tais algoritmos.

(v) De todas as formas de representação conhecidas, a BREP oferece os mais simples e consistentes algoritmos de conversão entre representações. Sendo assim, mesmo que a representação central não seja uma BREP, esta pode ser incorporada a uma aplicação, de forma a permitir conversões para outros sistemas;

Apesar dos pontos positivos observados anteriormente, a BREP traz consigo, também, alguns problemas. Os itens abaixo relacionam alguns destes aspectos, a saber:

- (i) Com execução da SOE e da SCD, a BREP é a forma de representação com as mais altas exigências de espaço de alocação, obrigando, muitas vezes, os próprios SMS's a realizarem o gerenciamento das áreas de memórias disponíveis. Com o surgimento de sistemas operacionais, que oferecem memórias virtuais³, tal ponto tem se tornado, no entanto, pouco relevante;
- (ii) Toda a integridade topológica do modelo representado só pode ser garantida pelo SMS, caso sejam utilizados EOP, como visto nas seções anteriores;
- (iii) Modeladores, baseados na BREP, são inerentemente pouco robustos. A razão maior para tal fato decorre dos já conhecidos problemas numéricos, com erros de arredondamento. Algoritmos tolerantes a erros devem ser desenvolvidos com a responsabilidade de não se comprometer a consistência geométrica do sólido como um todo. Um bom exemplo, para esse tipo de problema ocorre em operações booleanas, nas quais objetos não múltiplos podem ser facilmente gerados. Tais operações são baseadas quase que unicamente em cálculos de interseção de curvas e superfícies, estando estes,

³ Através do mecanismo de memórias virtuais, permite-se que as aplicações sejam dispensadas da árdua tarefa de gerenciar áreas de memória. A aplicação recebe por parte do sistema operacional, um espaço de endereçamento totalmente exclusivo.

conseqüentemente, sujeitos a erros numéricos de toda espécie.

Não obstante esses problemas, a BREP pode ser considerada como a forma de representação mais consistente e segura para implementações não específicas de SMS's, isto é, aquelas não direcionadas para uma única classe de aplicações.

Por fim, analisando-se o conjunto de critérios apresentados no capítulo 3 para a classificação de formas de representação, conclui-se que:

- 1^o Representações por Fronteira não são ambíguas, se as suas entidades topológicas (em particular as suas *Faces*) puderem ser representadas de maneira não ambígua;
- 2^o Por meio de profundos teoremas matemáticos, pode-se garantir que o *Conjunto-R*, seja definido de forma não ambígua através da especificação de sua fronteira. Decorre, portanto, de tal afirmação, o fato de que a BREP pode ser classificada como uma forma de representação completa;
- 3^o A validade de uma Representação por Fronteira pode ser garantida através da obtenção da sua integridade

geométrica e topológica, conforme visto nas seções anteriores. É importante notar que tais exigências devem ser satisfeitas de uma forma não isolada, ou seja, deve-se focalizar os problemas topológicos e geométricos sobre um mesmo ponto de vista;

4^o Como já foi exposto anteriormente, a BREP não pode ser classificada como sendo uma forma de representação simples e concisa. Tal característica é, no entanto, bastante comum nas representações avaliadas, uma vez que armazenam, explicitamente, a sua informação de fronteira;

5^o O critério de unicidade da BREP é foco de constantes discussões. Teoricamente, pode-se afirmar que a fronteira de um sólido é única. Na prática, no entanto, pode-se facilmente gerar, através da BREP, duas ou mais representações distintas para o mesmo objeto físico. Tendo em mente a importância de se possuir uma representação única para um objeto físico, propõe-se, na presente dissertação, um mecanismo novo a ser incorporado a SMS's, baseados na BREP, que permite a transformação de uma representação não única. Tal procedimento é denominado *Filtro-Topológico* (FTop) e pode ser invocado automaticamente pelo sistema após uma determinada seqüência de operações

topológicas sobre o modelo. A idéia básica por trás de um FTop é a de eliminar, de uma maneira consistente, entidades topológicas redundantes a uma representação única. A figura 5.25 demonstra o efeito de um FTop sobre uma representação não única de um cubo. Resumidamente, pode-se descrever o algoritmo utilizado na implementação de um FTop, da seguinte maneira:

- (i) Identificação das entidades topológicas a serem eliminadas através de procedimentos geométricos. Tais procedimentos incluem, por exemplo, a seleção de *Faces* (adjacentes) coplanares e de *Arestas* (adjacentes) colineares, quando percorridas sobre um mesmo *Ciclo*;
- (ii) Composição das entidades topológicas em grupos de eliminação. Cada grupo de eliminação é, então, submetido a uma seqüência de EOP's, de forma a retirá-los da estrutura topológica, segura e eficientemente.

6^o Em termos práticos, a eficiência de uma BREP está intimamente relacionada com a classe dos objetos nela representados (sólidos com faces planas ou curvas) e, naturalmente, com a quantidade/qualidade da informação

topológica e geométrica explicitamente disponível, na sua particular implementação (estrutura de dados).

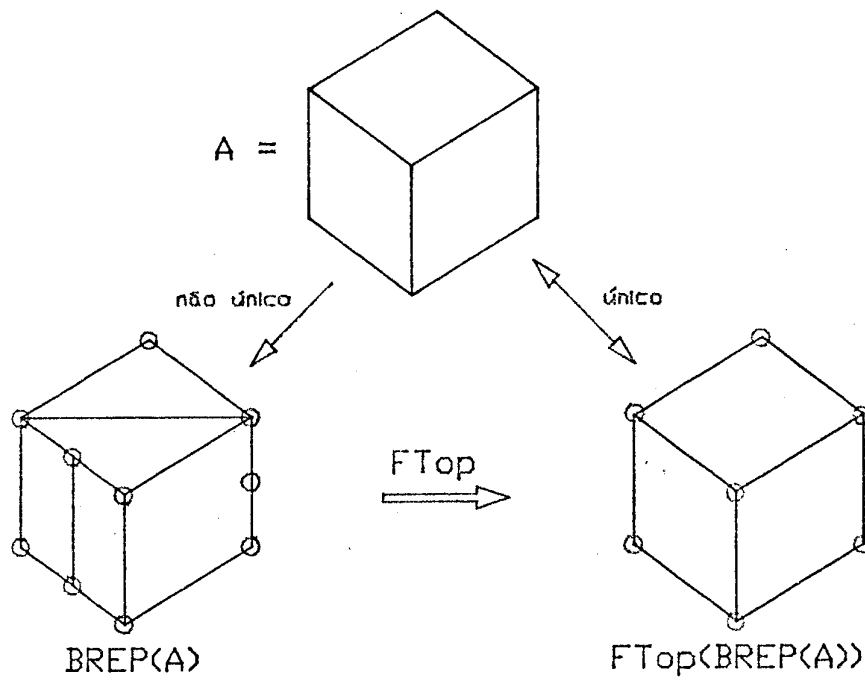


Figura 5.25 - Transformação gerada por um $FTop$

CAPÍTULO 6

FORMAS HÍBRIDAS DE REPRESENTAÇÃO

Estabeleceram-se, no capítulo 3, as exigências necessárias para a elaboração de formas de representação e, posteriormente, apresentaram-se, nos capítulos 4 e 5, as principais propostas de representação conhecidas e utilizadas atualmente. Mostrou-se, também, que, apesar de todas essas propostas oferecerem facilidades relevantes para a representação e modelagem de sólidos tridimensionais, nenhuma delas foi capaz de preencher, satisfatoriamente, os requisitos de uma completa forma de representação.

Durante os últimos anos, vários SMS's têm sido comparados. Dentre os critérios normalmente adotados nestas avaliações, ressalta-se a capacidade dos SMS's de simplesmente representarem um determinado objeto de prova. Repare-se que não se exige, em tais comparações, que os respectivos sistemas sejam capazes de modelar os objetos testados, mas que ao menos possuam estruturas simbólicas capazes de gerarem a sua única, completa, não ambígua, concisa, válida e eficiente, representação.

Não obstante o enfraquecimento de tais exigências,

observou-se que poucos sistemas obtiveram sucesso nessas avaliações e, raramente, um único SMS foi capaz de representar todos os objetos testados. Na procura por uma justificativa para tal contingência, notou-se que, selecionando-se alguns modeladores (com formas de representação distintas), poder-se-ia, facilmente, modelar e representar um número muito superior de objetos.

Tal afirmativa, aparentemente óbvia de um ponto de vista puramente teórico, já podia ser observada na própria indústria, que não podia simplesmente permitir a limitação e a incapacidade de um determinado sistema de modelar um objeto, que lhe fosse vital. Era comum em tais ambientes profissionais a utilização de dois ou mais sistemas de modelagem, cada qual com a suas facilidades e limitações.

Surgiu, então, a partir de tal fato, a idéia de que, em vez de se multiplicar o número de sistemas, se aumentasse o poder de representação destes, através da utilização de mais de uma forma de representação. Tais sistemas são classificados, portanto, como híbridos ou não homogêneos (*Hybrid Solid Modeling System - HSMS*), por utilizarem mais de uma forma de representação. Desta maneira, objetiva-se a complementação e fortificação das capacidades individuais de cada uma das representações adotadas por tais SMS's. A figura 6.1 apresenta a arquitetura de um SMS baseado em uma forma de representação

híbrida.

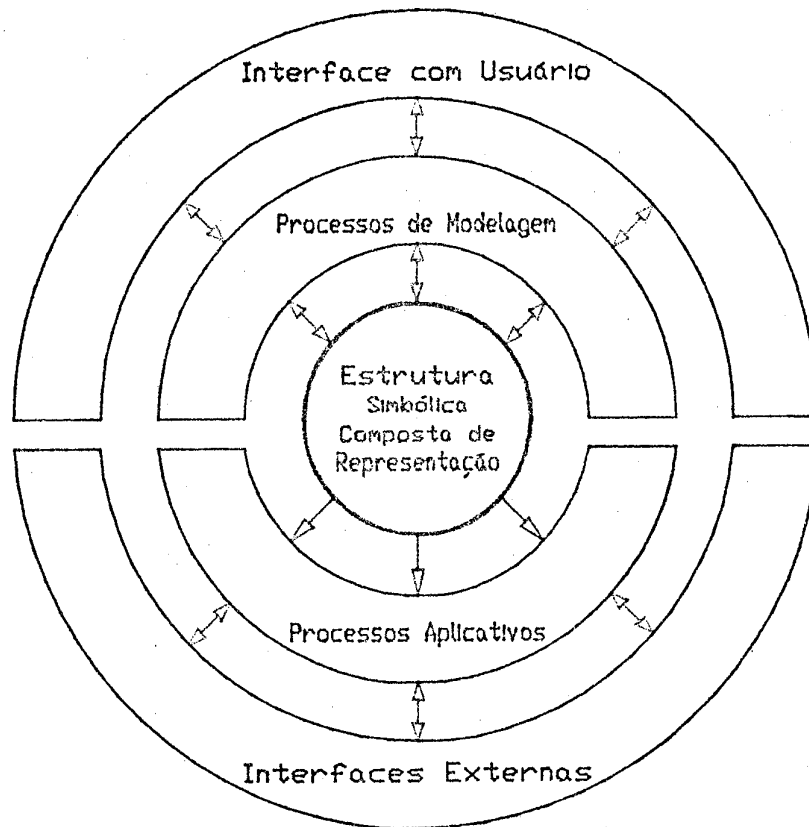


Figura 6.1 - Estrutura geral de um HSMS

Simplificando-se a idéia por detrás dos sistemas híbridos, verificou-se que estes não trazem, conceitualmente, nenhuma novidade sobre a já milenar filosofia de se concentrar esforços para superar obstáculos. No entanto, a simples aglomeração de formas de representação envolve questões de difícil abordagem.

CAPÍTULO 7

CONCLUSÕES

Uma vez abordados os principais conceitos e características da SRS, pode-se compreender a sua importância como sub-área da 3DGM. Apesar de, hoje em dia, a área da Modelagem Geométrica Tridimensional englobar, em si, um variado número de sub-áreas, é ainda a SRS a área, onde se concentram os maiores esforços de pesquisa. A principal justificativa para isso é o seu pobre poder de representação. Tomando-se, como exemplo, um simples arbusto de folhas, verifica-se, imediatamente, a impossibilidade, ou melhor, a inviabilidade de representá-lo através de qualquer uma das técnicas apresentadas.

De uma forma geral, acredita-se que o objetivo maior de expor os problemas e soluções, atualmente conhecidos para a representação de sólidos, foi alcançado. Não obstante isso, deve-se mencionar o fato de várias propostas, ainda pouco difundidas ou em fase de pesquisa, terem sido omitidas. Dentre elas, destacam-se algumas implementações em *Hardware* da CSG [53] sobre a forma de árvores octógonas, novos conceitos e metodologias de representação através da AREP e, ainda, de novas extensões da BREP [48], capazes de representar Sólidos de

Variedade múltipla (*Non-Manifold Solids - NMFLD*), importantes para algumas aplicações como as de FEM e Projeto & Manufatura de Circuitos Eletrônicos (*VLSI Design & Manufacturing - VLSIDM*).

Finalizando, lista-se, abaixo, uma série de conclusões e observações com relação à análise global da SRS, realizada nas partes I e II desse trabalho. São elas:

- (i) Sistemas de Modelagem Sólida não devem ser confundidos com o termo Sistemas de Modelagem Geométrica. Estes são apenas uma classe particular dos SMG's, que objetivam, unicamente, a modelagem e representação de objetos físicos manufaturáveis;
- (ii) A classificação exata de SMS's só pode ser realizada, analisando-se a(s) sua(s) forma(s) de representação e o efetivo Modelo que esse enfatiza (Imagem, Superfície, Objeto ou Sistema);
- (iii) A verificação dos critérios para a representação de objetos sólidos, apresentadas no capítulo 3, mostra-se indispensável para o projeto e implementação de SMS's, bem como para a elaboração e análise de novas formas e representação;

- (iv) Uma estrutura de dados não é capaz de representar um objeto tridimensional propriamente dito, mas sim a sua forma de representação, que estará sendo mapeada, simbolicamente, para um formato digital de informação e embutida, intrinsecamente, nos processos aplicativos e de modelagem;
- (v) Nenhuma das formas de representação, conhecidas até o momento, foi capaz de fornecer uma completa e aceitável proposta de solução para os problemas da SRS. Entretanto, a filosofia por trás dos HSMS's pode ser considerada, até o presente último, como a mais razoável, plausível e promissora abordagem para o projeto e implementação de sofisticados modeladores. A sua viabilidade prática exige, no entanto, um investimento alto em termos de *Hardware* e *Software*, somente justificável pelas promessas de um extenso poder de representação, uma fácil adaptação de processos aplicativos e a disponibilidade de um farto, eficiente e robusto conjunto de técnicas de modelagem.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] *Alexandroff, P.: Elementary Concepts of Topology; Dover, New York, (1961).*
- [2] *Allen, G.: An Introduction to Solid Modeling; Computer & Graphics, Vol.8, No.4, pp.439-447, (1984).*
- [3] *Agoston, M.K.: Algebraic Topology; Marcel Dekker, Inc., New York, (1976).*
- [4] *Ansaldi, S. L.De Floriani, & B.Falcidieno: Geometric Modeling of Solid Objects using a Face Adjacency Graph Representation; Computer Graphics and Applications, Vol.19, pp.131-140, (1985).*
- [5] *Baer A., Eastman C. & Henrion M.: Geometric Modeling: a Survey; Computer Aided Design, Vol.11, No.5, pp.253-272, (1979).*
- [6] *Baumgart, B.G.: Geometric Modeling for Computer Vision; Report STANCS-74-463, Standford University, Standford Artificial Intelligence Laboratory, (1974).*
- [7] *Baumgart, B.G.: A Polyhedron Representation for Computer Vision; AFIPS Conf. Proc. Vol.44, pp.589-596, (1975).*
- [8] *Braid, I.C.: The Synthesis of Solids Bounded by many Faces; Communications of ACM, Vol.18, pp.209-216, (1975).*

- [9] *Braid, I.C.: Notes on a Geometric Modeller; CAD Group Document No.101, Computer Laboratory, University of Cambridge, (1979).*
- [10] *Braid, I.C., Hillyard, R.C. & Stroud, I.A.: Stepwise Construction of Polyhedra in Geometric Modeling; Mathematical Methods in Computer Graphics and Design, K.W. Brodlie, Ed. New York, Academic Press, pp.123-141, (1980).*
- [11] *Brown, C.M.: PADL-2: a Technical Summary; IEEE Computer Graphics and Applications, Vol.2, pp.69-84, (1982).*
- [12] *CAM-I: An Interface between Geometric Modelers and Application Programs; Report No. R-80-GM-04, CAM-I, Inc., Arlington, Texas, (1980).*
- [13] *CAM-I: CAM-I Geometric Modeling Project Boundary File Design - XBF2; Report No. R-80-GM-04, CAM-I, Inc., Arlington, Texas, (1980).*
- [14] *Casale, S.M. & Stanton, L.E.: An Overview of Analytic Solid Modeling; IEEE Computer Graphics and Applications, No.2, pp.45-56, (1985).*
- [15] *Chiyokura, H. & Kimura, F.: Design of Solids with Free-Form Surfaces; Computer Graphics, Vol.17, No.3, pp.289-298, (1983).*
- [16] *Chiyokura, H. & Kimura, F.: A Method of Representing the Solid Design Process; IEEE Computer Graphics and Applications, No.4, pp.32-41, (1985).*

- [17] *Chiyokura, H. & Kimura, F.:* A Representation of Solid Design Process Using Basic Operations; Proc. Computer Graphics Tokyo '84, Ed. Kunii, T.L., (1985).
- [18] *Chiyokura, H.:* An Extended Rounding Operation for Modeling Solids with Free-Form Surfaces; IEEE Computer Graphics and Applications, No.12, pp.27-36, (1987).
- [19] *Chiyokura, H.:* Solid Modeling with DESIGNBASE - Theory and Implementation; Wesley Publishing Company, (1988).
- [20] *Eastman, C. & Weiler, K.:* Geometric Modeling using the Euler Operators; 1st Annual Conf. Computer Graphics in CAD/CAM, MIT, pp.248-259, (1979).
- [21] *Flaquer, J., Carbajal, A., Mendez, M.A.:* Edge-edge Relationships in Geometric Modeling; Computer Aided Design, Vol.19, No.5, pp.237-244, (1987).
- [22] *Grayer, A.R.:* Alternative Approaches in Geometric Modeling; Computer Aided Design, Vol.12, No.4, pp.189-192, (1980).
- [23] *Gossling, T.H.:* The DUCT System for Design of Practical Objects; Proc. World Congress on the Theory of Machines and Mechanisms, (1976).
- [24] *Hinds, J.K., & Kuan, L.P.:* Surfaces Designed by Curve Transformations; Proc. 15th Numerical Control Society Annual Meeting, pp.325-340, (1978).
- [25] *Hoffmann, C.M. & Hopcroft, J.E:* Geometric Ambiguities in Boundary Representations; Computer Aided Design, Vol.19, No.3, pp.141-147, (1987).

- [26] *Joshua, U.T.:* Accurate Solid Modeling Using Polyhedral Approximations; IEEE Computer Graphics and Applications, No.4, pp.14-28, (1988).
- [27] *Kemper, A. & Wallrath, M.:* An Analysis of Geometric Modeling in Database Systems; ACM Computing Surveys, Vol.19, No.1, pp.47-90, (1987).
- [28] *Lima, E.L.:* O Teorema de Euler sobre Poliedros; IMPA, Instituto de Matemática Pura e Aplicada, RJ, (1985).
- [29] *Mäntylä, M. & Sulonen, R.:* GWB: a Solid Modeler with Euler Operators; IEEE Computer Graphics and Applications, Vol.2, pp.17-31, (1982).
- [30] *Mäntylä, M.:* An Inversion Algorithm for Geometric Models; ACM Computer Graphics, Vol.16, No.3, pp.51-59, (1982).
- [31] *Mäntylä, M.:* Topological Analysis of Polygon Meshes; Computer Aided Design, Vol.15, No.4, (1983).
- [32] *Mäntylä, M.:* Computational Topology: A study of Topological Manipulations and Interrogations in Computer Graphics and Geometric Modeling; Acta Polytechnica Scandinavica, Mathematics and Computer Science Series No.37. Helsinki, (1983).
- [33] *Mäntylä, M. & Tamminen, M.:* Localized Set Operations for Solid Modeling; Computer Graphics, Vol.17, pp.279-288, (1983).

- [34] *Mäntylä, M.:* A Note on the Modeling Space of Euler Operators; Computer Vision, Graphics and Image Processing, Vol.26, No.1, pp.45-60, (1984).
- [35] *Mäntylä, M.:* An Introduction to Solid Modeling; Computer Science Press, (1988).
- [36] *Mortensen, M.:* Geometric Modeling; Wiley, New York, NY, USA, (1985).
- [37] *Pratt, M.J.:* Solid Modeling and the Interface between Design and Manufacture; IEEE Computer Graphics and Applications, Vol.6, (1984).
- [38] *Requicha, A.G. & Voelcker, H.B.:* Constructive Solid Geometry; Production Automation Project, Tech. Memo. No.25, University of Rochester, (1977).
- [39] *Requicha, A.G.:* Representations of Rigid Solids - Theory, Methods, and Systems; ACM Computing Surveys, Vol.12, No.4 pp.437-464, (1980).
- [40] *Requicha, A.G. & Voelcker, H.B.:* Solid Modeling: a Historical Summary and Contemporary Assessment; IEEE Computer Graphics and Applications, Vol.2, pp.9-24. (1982).
- [41] *Requicha, A.G. & Voelcker, H.B.:* Solid Modeling: Current Status and Research Directions; IEEE Computer Graphics and Applications, Vol.3, pp.25-37, (1983).
- [42] *Sutherland, I.E.:* Sketchpad - a Man Machine Communication System; MIT, Lincoln Laboratory Technical Report, No.296, (1963).

- [43] *Tilove, R.B. & Requicha, A.G.:* Closure of Boolean Operations on Geometric Entities; *Computer Aided Design*, Vol.12, No.5, (1980).
- [44] *Toriya, H., Satoh, T., Ueda, K. & Chiyokura, H.:* Undo and Redo Operations for Solid Modeling; *IEEE Computer Graphics and Applications*, No.4, pp.35-42, (1986).
- [45] *Toriya, H., Satoh, T., Ueda, K. & Chiyokura, H.:* Invertible Set Operations for Solid Modeling; *Proc. Computer Graphics Tokyo '85*, Ed. Kunii, T.L., (1985).
- [46] *Weiler, K.J.:* Adjacency Relationships in Boundary Graph Based Solid Models; (1983).
- [47] *Weiler, K.J.:* Edge Based Data Structures for Solid Modeling in Curved-Surface Environments; *IEEE Computer Graphics and Applications*, Vol.5, pp.21-40, (1985).
- [48] *Weiler, K.J.:* Topological Structures for Geometric Modeling; Ph.D. Thesis, Rensselaer Polytechnic Institute Troy, New York, (1986).
- [49] *Wilson, P.:* Euler Formulas and Geometric Modeling; *IEEE Computer Graphics and Applications*, Vol.5, pp.24-36, (1985).
- [50] *Wilson, P., Faux, I.D., Ostrowski, M.C., Pasquill, K.G.:* Interfaces for Data Transfer Between Solid Modeling Systems; *IEEE Computer Graphics and Applications*, No.1, pp.41-51, (1985).
- [51] *Wittek, D.:* Solid Modeling and System Design; *Computer Graphics*, Vol.8, No.4, pp.423-431, (1984).

- [52] *Woo, T.C.:* A Combinatorial Analysis of Boundary Data Schemata; IEEE Computer Graphics and Image Processing, pp.19-27, (1985).
- [53] *Yamaguchi K., Kunii T.L., Fujimura K.:* Octree-Related Data Structures and Algorithms; IEEE Computer Graphics and Applications, Vol.1, pp.53-59, (1984).