6. O problema: Estrutura de dados

Nós consideraremos o problema: *dado:* um conjunto finito S de pontos no plano; *encontrar:* o fecho convexo $\text{conv}(S)$ dos pontos em S. Da mesma maneira que fizemos no caso bidimensional, antes de continuarmos devemos ter claro o quê desejamos como resposta dos algoritmos.

Quando tratamos do problema de encontrar o fecho convexo de um conjunto de pontos no plano nos timanhos como resposta dos algoritmos (da maioria deles) o polígono que representava o fecho convexo (vértices extremos junto com uma ordem). No caso tridimensional existem varias possibilidades de representarmos o fecho convexo e, como era de se esperar, diferentes representações levarão a diferentes complexidades de tempo do mesmo algoritmo; algumas mais rápidas e outras mais lentas.

Trataremos nesta seção do problema de como representar um fecho convexo tridimensional. O material que veremos foi extraído do Capítulo 3 de [6] e Capítulo 4 de [10].

6.1. **Listas de faces.** A maneira mais simples de representarmos um poliedro tridimensional e representando cada uma de suas faces através de uma lista (ordenada) de vértices. Desta forma um poliedro é dado através de uma lista de suas faces (polígonos) que formam a sua superfície.

A grande desvantagem desta estrutura de listas de faces é que ela contém só um tipo de informação de adjacência: adjacência entre vértices de uma mesma face. Gastariamos muito tempo se estivessemos interessados em determinar outro tipo de adjacências, e.g. todas as faces adjacentes a uma face dada ou se desejassemos listar todas as faces que são adjacentes a um dado vértice; em ambos os casos deveriamos percorrer toda a estrutura.

6.2. **Estrutura de dados para poliedros simpliciais.** Poliedros *simpliciais* são aqueles em que cada face é um triângulo. Para representarmos um polígono P temos que representar três tipos primitivos de tipos de dados: vértices, arestas e faces. Uma maneira de fazermos isto é mantermos todos os vértices em uma lista duplamente ligada, a ordem dos elementos não tem nenhum significado. Faremos o mesmo com arestas e faces. Cada elemento destas lista tem um tamanho fixo. A estrutura correspondente a um vértice contém a sua coordenada. A estrutura da aresta contém apontadores para os vértices que são seus pontos extremos e apontadores para as duas faces que são incidentes na aresta. A estrutura correspondente uma face contém apontadores para os três vértices que formam a face, além de apontadores para as três arestas.

```
struct tVertexStructure {
        int v[3];
        tVertex next, prev;
};

struct tEdgeStructure {
        tVertex endpts[2];
        tFace adjface[2];
```

```
        tEdge next, prev;
};

struct tFaceStructure {
        tVertex vertex[3];
        Edge edge[3];
        tFace next, prev;
};
```

6.3. **Estrutura de dados winged-edge.** Uma outra maneira de representarmos um poliedro é representando a sua superfície, mais especificamente, o seu 1-esqueleto (grafo planar). (Estruturas de dados que representam propriedades topológicas do poliedro são chamadas de *estruturas dados topológocas* (veja página 57 de [6].)

A representação que veremos chama-se *winged-edge* ('aresta-alada') foi desenvolvida por Baumgart [1] em 1975 e é ainda a estrutura de dados mais popular para representar a superfície de um poliedro.

Esta representação é capaz de produzir rapidamente todas as possíveis relações de adjacência entre vértices, arestas e faces de um poliedro. Por exemplo, dadas duas faces do poliedro podemos determinar se elas são adjacentes em tempo linear no número total de vértices destas faces. Dado um vértice do poliedro seremos capazes de listar todas as faces que são incidentes a este vértice em tempo proporcional ao número destas faces.

O foco da estrutura de dados winged-edge é a aresta. Esta estrutura mantém uma lista de vértices, arestas e faces onde:

Vértice: a estrutura que representa cada vértice v mantém as coordenadas (x, y, z) do vértices juntamente com um apontador para um aresta arbitrária incidente ao vértice, denotaremos por `av`(v) a aresta incidente ao vértice v.

Face: cada face f matemos um apontador uma aresta arbitrária `af`(f) contida na fronteira de f.

Aresta: a estrutura correspondente a cada aresta e tem quatro oito apontadores:

1. Dois apontadores para os vértices $v_1(e)$ e $v_2(e)$ que são pontas da aresta e. A ordem destes vértices fornece uma orientação para a aresta.
2. Um apontador para cada uma das duas faces `fccw`(e) e `fcw`(e) incidentes a e. A face `fccw`(e) é a esquerda de $e = v_1(e)v_2(e)$ (*counterclockwise*, já que a orientação induzida por e nesta face é anti-horária) e a face `fcw`(e) é a aresta a direita (*clockwise*, já que a orientação induzida por e nesta face é horária).
3. Quatro apontadores para arestas adjacentes a e (as *wings* (asas) de e), um apontador para cada arestas que precedem e sucedem e nas faces `fccw`(e) e `fcw`(e). Assim, `pccw`(e) (previus conunterclockwise)

e `nccw`(e) (*next counterclockwise*) representam as arestas que precedem e sucedem a aresta e na face `fccw`(e), respectivamente (orientada de acordo com a orientação da aresta e, i.e. sentido anti-horário). Analogamente `pcw`(e) e `ncw`(e) representam as arestas que precedem e sucedem a aresta e na face `fcw`(e). A Figura 8 ilustra esta estrutura.

Note que as estruturas que representam um vértice, aresta e face têm tamanho constante—uma propriedade que é bem útil.

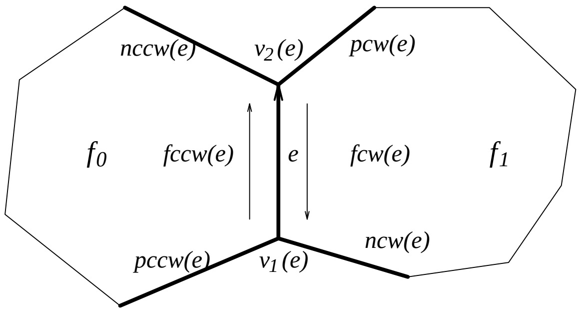

FIGURA 8. A estrutura de dados winged-edge.

A estrutura winged-edge correspondente ao cubo da Figura 9 está representado pelas lista de vértices, arestas e faces abaixo.

vértice	(x, y, z)	`av`
p_1	$(1,0,0)$	e_1
p_2	$(1,1,0)$	e_2
p_3	$(1,1,1)$	e_3
p_4	$(1,0,1)$	e_4
p_5	$(0,0,0)$	e_9
p_6	$(0,1,0)$	e_{10}
p_7	$(0,1,1)$	e_{11}
p_8	$(0,0,1)$	e_{12}

face	`af`
f_1	e_1
f_2	e_2
f_3	e_{10}
f_4	e_{12}
f_5	e_1
f_6	e_8

aresta	v_1	v_2	`fccw`	`fcw`	`pccw`	`nccw`	`pcw`	`ncw`
e_1	p_1	p_2	f_1	f_5	e_4	e_2	e_6	e_5
e_2	p_2	p_3	f_1	f_2	e_1	e_3	e_7	e_6
e_3	p_3	p_4	f_1	f_6	e_2	e_4	e_8	e_7
e_4	p_4	p_1	f_1	f_4	e_3	e_1	e_5	e_8
e_5	p_1	p_5	f_5	f_4	e_1	e_9	e_{12}	e_4
e_6	p_2	p_6	f_2	f_5	e_2	e_{10}	e_9	e_1
e_7	p_3	p_7	f_6	f_2	e_3	e_{11}	e_{10}	e_2
e_8	p_4	p_8	f_4	f_6	e_4	e_{12}	e_{11}	e_3
e_9	p_5	p_6	f_5	f_3	e_5	e_6	e_{10}	e_{12}
e_{10}	p_6	p_7	f_2	f_3	e_6	e_7	e_{11}	e_9
e_{11}	p_7	p_8	f_6	f_3	e_7	e_8	e_{12}	e_{10}
e_{12}	p_8	p_5	f_4	f_3	e_8	e_5	e_9	e_{11}

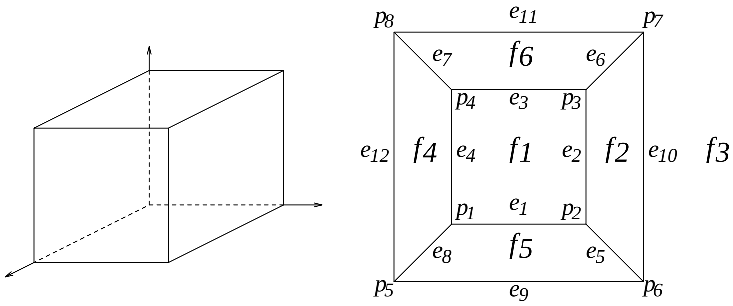

FIGURA 9. Um cubo e seu 1−esqueleto (grafo planar).