

Représentations par frontières (BRepr): réalisations

Stefka GUEORGUIEVA
stefka.gueorguieva@labri.fr

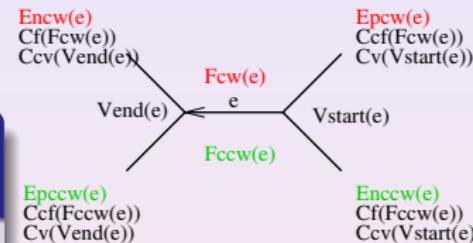
Plan

- ① BRepr : structure topologique
- ② Structures "arête-ailée"
- ③ Structure "Quad-edge"
- ④ Structure "demi-arête" de M.Mantyla
- ⑤ CGAL : Structure "demi-arête" de L.Kettner

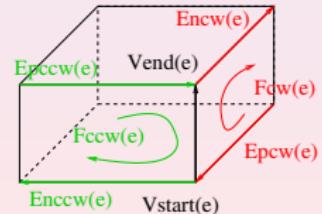
Éléments de base et relations de voisinage

Orientations : Cw ("clockwise") et Ccw ("counterclockwise")

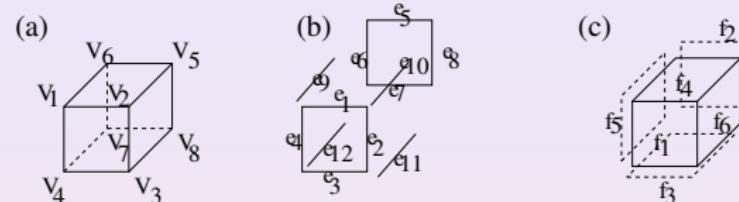
- Arête-Sommet **Vstart(e)** et **Vend(e)**
- Arête-Arête : (préc.) **Ccf(Fcw(e))**, **Ccf(Fccw(e))** et (suiv.) **Cf(Fcw(e))** et **Cf(Fccw(e))**
- Arête-Face : **Fcw(e)** et **Fccw(e)**



Edge Wings

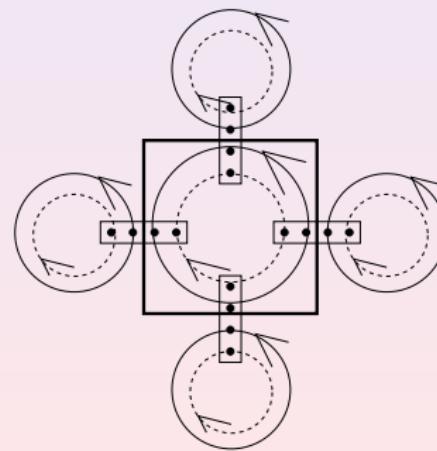


Arête-ailée ("Winged Edge Face")



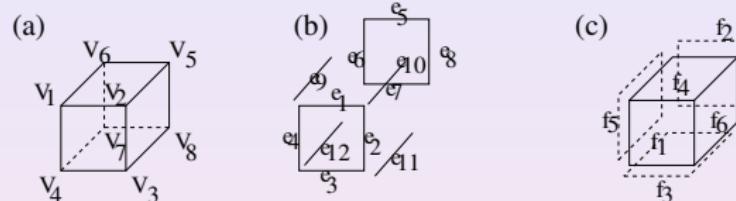
Winged Edge Face

Arête ailée basée
face



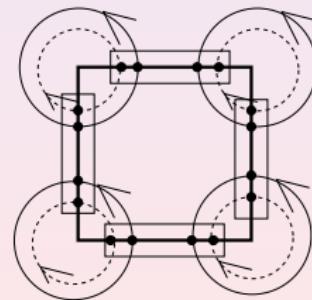
"Winged-Edge-Face"

Arête-ailée ("Winged Edge Vertex")



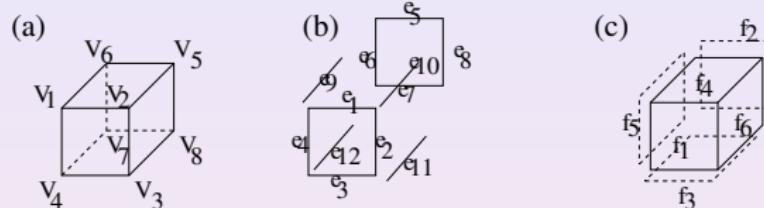
**Winged Edge
Vertex**

Arête ailée basée
sommet



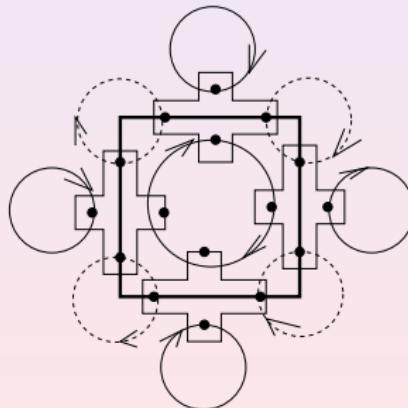
"Winged-Edge-Vertex"

Arête-ailée ("Quad Edge")



Quad Edge

Dualité de la
représentation des
relations
Sommet-Face



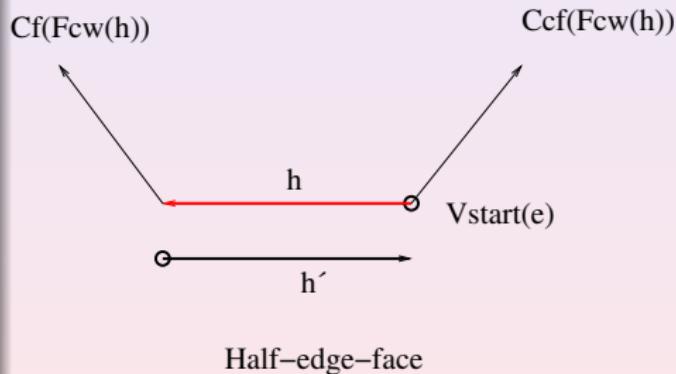
"Quad-Edge"

Demi-arête \leftrightarrow Arête-ailée

"Half Edge" \leftrightarrow "Winged Edge"

Half Edge (HE)

- Début : half-edge($V_{start}(h)$)
- Demi-arête correspondante : half-edge(h)
- Face : half-edge($F_{cw}(h)$)
- Demi-arête préc. : half-edge($Ccf(F_{cw}(h))$)
- Demi-arête suiv. : half-edge($Cf(F_{cw}(h))$)



Half Edge [M.Mantyla, 1988]

M.Mantyla

"Solid Modeling", CompSciPress,
1988

BRepr, HalfEdge

```
typedef float vector[3];
typedef short Id;
typedef struct Solid *Solid;
typedef struct Face *Face;
typedef struct Loop *Loop;
typedef struct HalfEdge *HalfEdge;
typedef struct Edge *Edge;
typedef struct Vertex *Vertex;
typedef union Nodes *Node;
```

Half Edge [M.Mantyla, 1988]

Arête

```
struct Edge
{
    HalfEdge eHe1;
    HalfEdge eHe2;
    Edge nextE;
    Edge prevE;
};
```

Brin d'arête

```
struct HalfEdge
{
    Edge HeEdge;
    Vertex HeVertex;
    Loop HeLoop;
    HalfEdge nextHe;
    HalfEdge prevHe;
};
```

Sommet

```
struct Vertex
{
    Id vNumber;
    HalfEdge vHalfEdge;
    vector vCoordinates;
    Vertex nextV;
    Vertex prevV;
};
```

CGAL HalfedgeDS

L. Kettner

"Using generic programming for designing a data structure for polyhedral surfaces, 1999

- Computational Geometry Algorithms Library CGAL, <http://www.cs.uu.nl/CGAL>
- <http://doc.cgal.org/latest/HalfedgeDS/>
- <http://doc.cgal.org/latest/Polyhedron/>

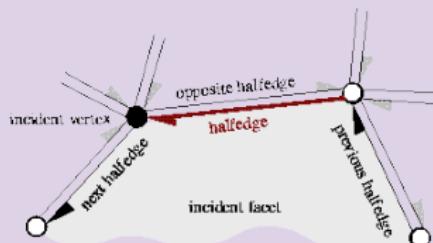
CGAL HalfedgeDS

Hiérarchie des niveaux de la création et la manipulation du BRepr

- ① Vertex, Halfedge et Facet : composantes
- ② Halfedge_data_structure : gestion du stockage des composantes
- ③ Polyhedron : interface pour la construction et la manipulation des BReprs valides.

CGAL HalfedgeDS

[Kettner99], Fig 7



Vertex
Halfedge_handle halfedge() Point& point()

Halfedge
Halfedge_handle opposite()
Halfedge_handle next()
Halfedge_handle prev()
Vertex_handle vertex()
Facet_handle facet()

Facet
Halfedge_handle halfedge()
Normal& normal()
Plane& plane()

Construction incrémentale d'un polyèdre : 1/2

```
template <class HDS>
class Build_triangle : public CGAL::Modifier_base<HDS> {
public:
    Build_triangle() {}
    void operator()( HDS& hds ) {
        // Postcondition: hds is a valid polyhedral surface.
        CGAL::Polyhedron_incremental_builder_3<HDS> B( hds, true );
        B.begin_surface( -3, 1, 6 );
        typedef typename HDS::Vertex     Vertex;
        typedef typename Vertex::Point   Point;
        B.add_vertex( Point( 0, 0, 0 ) );
        B.add_vertex( Point( 1, 0, 0 ) );
        B.add_vertex( Point( 0, 1, 0 ) );
```

Construction incrémentale d'un polyèdre : 2/2

```
B.begin_facet();
    B.add_vertex_to_facet( 0 );
    B.add_vertex_to_facet( 1 );
    B.add_vertex_to_facet( 2 );
    B.end_facet();
    B.end_surface();
}
};
```

Pour approfondir

L. Kettner

"Designing a Data Structure for Polyhedral Surfaces"