

# **Gràfics i Visualització de Dades**

## **Resum representacions de malles poligonals**



UNIVERSITAT<sub>DE</sub>  
BARCELONA

**Març 2021**

**Eduard Vives Isern**

## Índex

Introducció representacions de malles poligonals .....	3
1. Representació explícita.....	3
2. Vèrtexs indexats .....	3
3. Adjacència de Cares .....	4
4. Model Winged-edge .....	4

## Representacions de malles poligonals

A continuació es mostra un breu resum sobre les possibles representacions de malles poligonals presentades a teoria.

Cada una de les representacions surt per unes certes necessitats, no hi ha una representació millor o pitjor, sinó que depenent de la consulta que estiguem fent sobre la malla de polígons ens anirà millor una representació o un altre.

### 1. Representació explícita:

- Representació més de força bruta. És directament la llista de vèrtexs de cada polígon. Tenim una llista dels triangles (numerats) que formen la malla poligonal i per cada triangle guardem els 3 vèrtexs (punts) pels que està format ordenats en sentit anti-horari, i cada punt són 3 floats (x, y i z).
- Quantitat de memòria: Per tant, per n triangles:  $3 \cdot 3 \cdot n$  floats.
- No connectivitat explícita: no es guarda explícitament quin triangle està connectat amb quin.
- Repetició de vèrtex: A part, estem repetint vèrtexs.

### 2. Vèrtexs indexats:

- Llista única de vèrtexs sense repeticions: Aquí tenim una llista única de vèrtexs sense repeticions (numerats), i guardem en una altra llista, per cada polígon (els quals es troben numerats), els vèrtexs que els formen, representats en aquesta llista amb el seu valor (índex) a la primera llista.  
D'aquesta forma ens estalviem molta quantitat de memòria en no repetir coordenades de vèrtex (els tenim únics) i al guardar únicament els índexs d'aquests vèrtexs per cada cara.
- Sentit antihorari: A més aquests índexs estan ordenats en sentit antihorari.
- Connectivitat triangle: Tenim la topologia de polígon, o sigui, com s'uneixen els vèrtexs explícitament (quin s'uneix amb quin).
- No es coneixen les cares veïnes: No tenim topologia de mesh explícita.
- Quantitat de memòria:

Per  $n$  triangles:  $3*n$  ints + 1 vector de vèrtexs (3 floats per vèrtex).

Gastem menys memòria que amb la representació explícita.

### 3. Adjacència de Cares:

- Llista única de vèrtexs sense repeticions: Tenim com abans una llista única de vèrtexs sense repeticions (numerats).
- Llista de Triangles: Igual que abans també guardem en una altra llista, per cada polígon o triangle (numerats), els vèrtexs que els formen, representats en aquesta llista amb el seu valor (índex) a la primera llista. La diferència amb la representació de Vèrtexs indexats, és que per cada triangle tenim amb quins altres triangles està connectat i, per tant, quins són adjacents.
- A nivell de vèrtex a part de les seves coordenades també tenim un dels triangles al qual pertany aquest vèrtex.
- Amb tot això el que aconseguim és:
  - En primer lloc saber des dels triangles, els triangles que són adjacents sense haver-los de calcular com en les representacions anteriors. Tenim la topologia explícita de la malla.
  - En segon lloc des d'un vèrtex qualsevol podem saltar a la seva cara, i de la seva cara podríem saltar a les cares adjacents, navegant per tota la mesh gràcies a aquesta estructura.
- Quantitat de memòria:  
Per  $n$  triangles: 6 enters per triangle i 3 floats + 1 enter per vèrtex.  
 $3*n\_vertexs$  floats + 1  $*n\_vertexs$  enter + 6 $*n$  enters
- Connectivitat de mesh i de polígon explícites: Amb aquesta representació tenim connectivitat de polígon, ja que sabem quins vèrtexs s'uneixen amb quins, i connectivitat de malla, ja que sabem amb quines cares estem connectats.

### 4. Model Winged-edge:

- Representació més completa i més utilitzada en gràfics. Guarda cares, arestes i vèrtex. Ens dona tota la possible navegació per una malla de polígons.
- L'element central ja no són els vèrtexs o les cares, són les arestes.
- Llista d'arestes:

- Vèrtex inicial i final
- Les dues cares adjacents
- Arestes anterior i posterior de les dues cares
- Llista de vèrtexs:
  - Posició
  - Una aresta adjacent
- Per cada cara guardem una aresta adjacent, amb aquesta ja podem navegar per totes les cares. Té topologia de malla totalment.
- Ocupa molta més memòria que les representacions anteriors.