

Procesamiento Geométrico y Análisis de Formas

Ivan Sipiran

Estructuras de Datos para Mallas

- Eficiencia de los algoritmos depende de la estructura de datos usada.
- Hay requerimientos
 - Topológicos
 - Necesitamos soporte para mallas no manifold?
 - Sólo mallas triangulares o polígonos en general?
 - Algorítmicos
 - Qué tipo de operaciones queremos realizar? Rendering? Consultas de vecindad?
 - La malla cambia con el tiempo?

Estructuras basadas en caras

- La información básica es el triángulo. Se almacena la información de los vértices por cada cara.

Triangles								
x_{11}	y_{11}	z_{11}	x_{12}	y_{12}	z_{12}	x_{13}	y_{13}	z_{13}
x_{21}	y_{21}	z_{21}	x_{22}	y_{22}	z_{22}	x_{23}	y_{23}	z_{23}
...				
...				
...				
x_{F1}	y_{F1}	z_{F1}	x_{F2}	y_{F2}	z_{F2}	x_{F3}	y_{F3}	z_{F3}

- Alta redundancia
- No conectividad

Estructura con indexación de caras

- Los vértices se almacenan una sola vez

Vertices	Triangles
x_1 y_1 z_1	i_{11} i_{12} i_{13}
...	...
x_v y_v z_v	...
	...
	...
	i_{F1} i_{F2} i_{F3}

- Se reduce la redundancia
- No hay conectividad

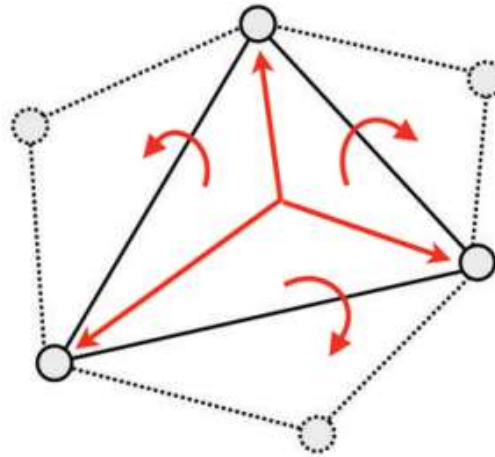
Operaciones de conectividad

- Acceso a vértices, caras y aristas
- Recorrer aristas de una cara
- Acceso a caras compartidas por aristas
- Acceso a vértices de una arista
- Acceso a topología alrededor de un vértice

Conectividad basada en caras

Vertex	
Point	position
FaceRef	face

Face	
VertexRef	vertex[3]
FaceRef	neighbor[3]



- Vértice almacena referencia a cara
- Cara almacena referencias a caras vecinas

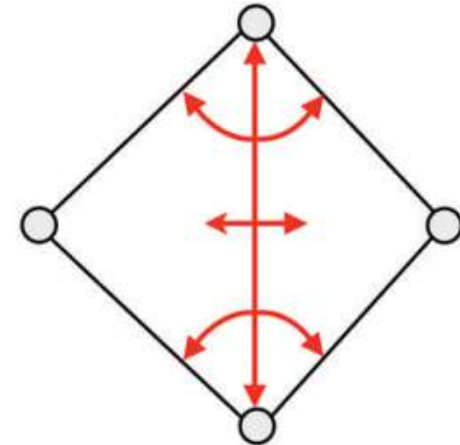
Estructuras basadas en aristas

- Winged-Edge

Vertex	
Point	position
EdgeRef	edge

Face	
EdgeRef	edge

Edge	
VertexRef	vertex[2]
FaceRef	face[2]
EdgeRef	next[2]
EdgeRef	prev[2]



Estructuras basadas en aristas

- Half-edge

Vertex	
Point	position
HalfedgeRef	halfedge

Face	
HalfedgeRef	halfedge

Halfedge	
VertexRef	vertex
FaceRef	face
HalfedgeRef	next
HalfedgeRef	prev
HalfedgeRef	opposite

