



Alcuni algoritmi per mesh di poligoni

Mesh - algoritmi

- Esistono numerosi algoritmi che agiscono su mesh di poligoni
- Ne vediamo alcuni:
 - Calcolo del bounding-box
 - Semplificazione di un mesh
 - Marching cubes
 - Calcolo del convex-hull di un insieme di punti 3D

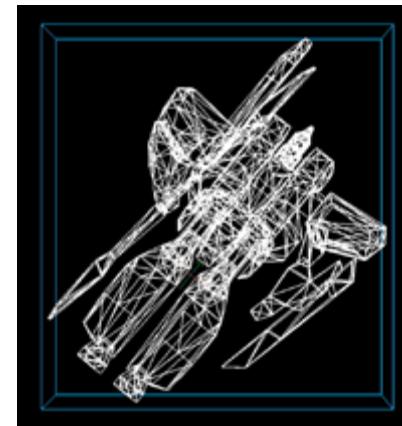
Calcolo del bounding-box

Bounding-box= Parallelepipedo che contiene l'oggetto

facile da calcolare rispetto agli assi coordinati

AABB – axis aligned boundig box

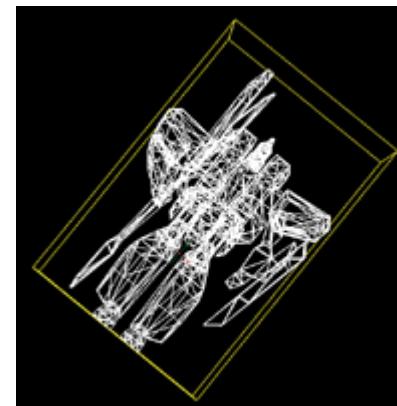
- si itera sui vertici per trovare il max e il min di tutte le x, le y e le z
- se l'oggetto ruota si deve ricalcolare



OOBB – object oriented bounding box

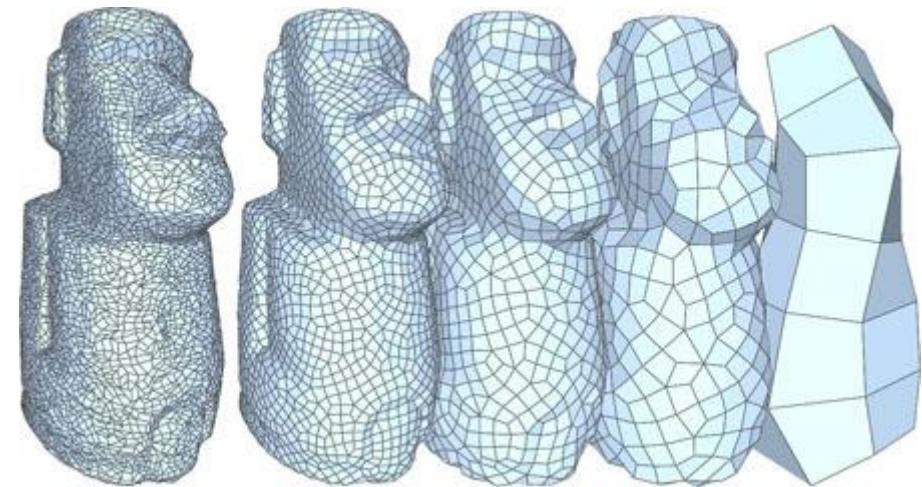
bounding box allineato con gli assi principali dell'oggetto

- calcolo più complesso



Mesh – Semplificazione

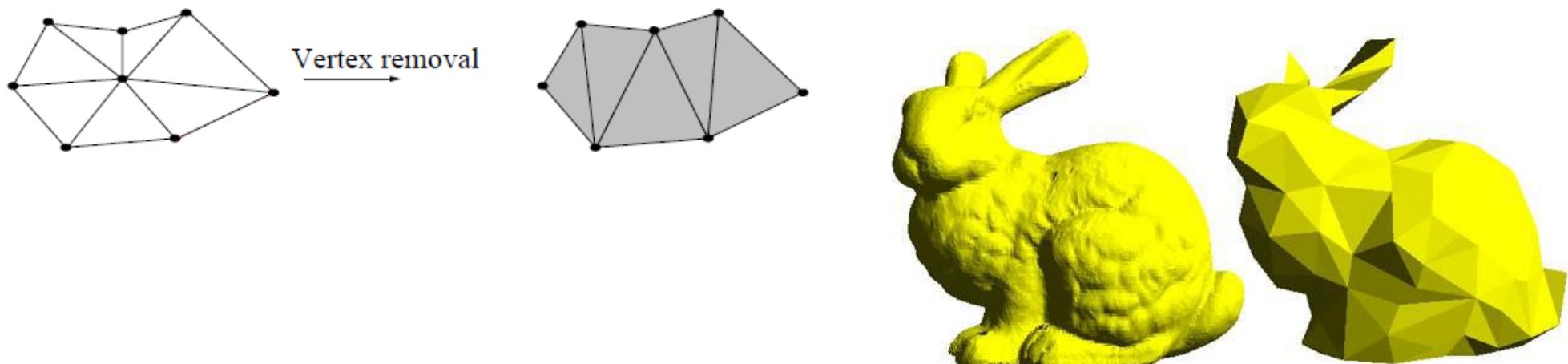
- talvolta è necessario semplificare un mesh.
- tipicamente si applica algoritmo incrementale che rimuove un vertice alla volta e ripara il vuoto lasciato.
- si vuole comunque mantenere una buona approssimazione del mesh originale.
- vi sono due categorie di algoritmi:
 - *decimazione dei vertici*.
 - *contrazione iterativa degli spigoli*.



Mesh – Semplificazione

Decimazione dei vertici

- Si selezionano vertici “poco importanti”, basandosi su euristiche locali, si rimuovono e si ri-triangola il vuoto risultante.
- Ad ogni passo di rimozione si deve determinare il vertice candidato ad essere rimosso, in base a criteri euristici.
- Un criterio è quello di diradare le zone a bassa curvatura.



Mesh – Semplificazione

Contrazione iterativa di spigoli

- Agli spigoli viene associato un costo della contrazione dello spigolo stesso.
- Ad ogni iterazione viene eliminato per contrazione lo spigolo di costo minore, ed i costi dei vicini vengono aggiornati.
- I vari metodi differiscono per la metrica di costo impiegata.
- Una metrica semplice:

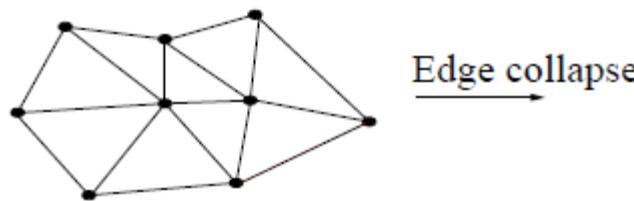
$$\epsilon(\overline{P_1 P_2}) = \frac{\|P_2 - P_1\|}{1 + n_l \cdot n_r}$$

dove P_1 e P_2 sono i vertici dello spigolo e n_l e n_r sono i versori normali delle due facce incidenti sullo spigolo. Seleziona spigoli corti in zone a bassa curvatura (dove l'angolo tra le due facce incidenti allo spigolo è piccolo).

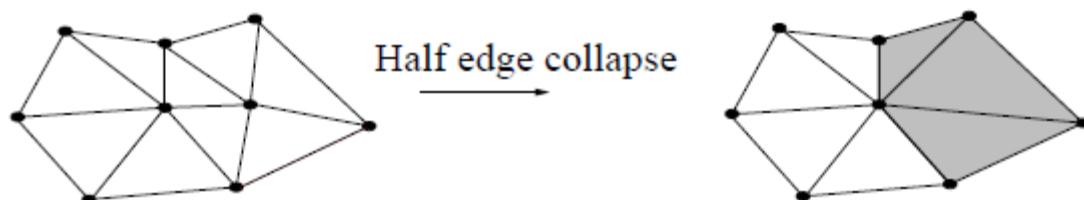
Mesh – Semplificazione

Contrazione iterativa di spigoli

- tecniche di rimozione degli spigoli



si crea un nuovo vertice



uno dei due vertici dello spigolo rimosso viene mantenuto

Mesh – Semplificazione

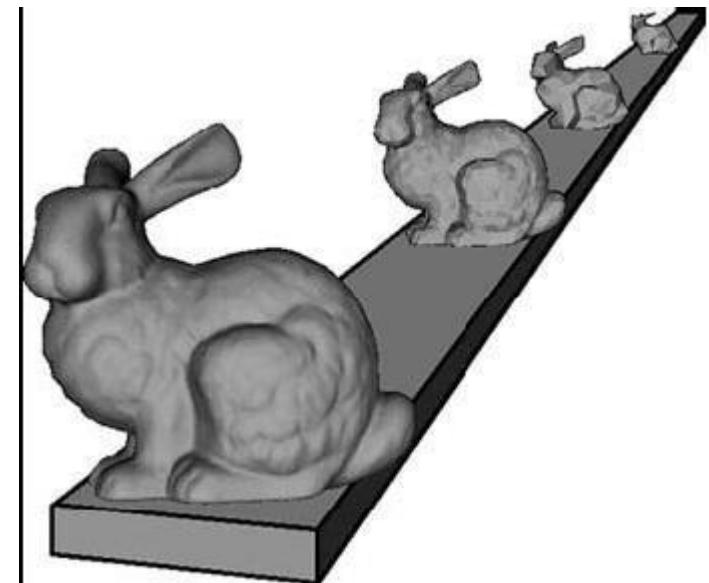
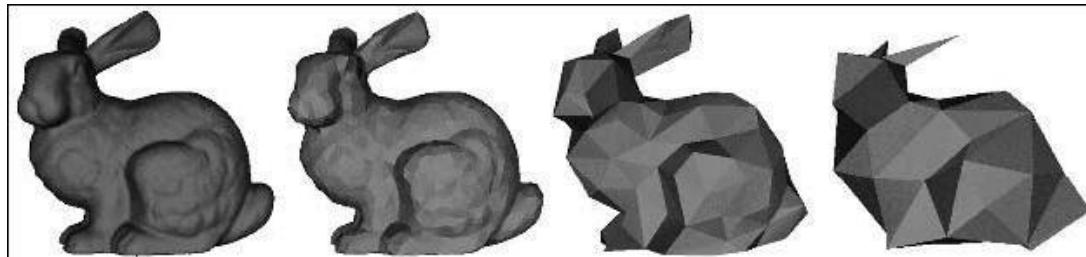
Applicazione

- ***Level of detail (LOD)*** per aumentare l'efficienza del sottosistema geometrico

Si costruiscono molteplici rappresentazioni dell'oggetto a diverse risoluzioni del mesh.

Si utilizzano versioni ad alta risoluzione se l'oggetto è vicino all'osservatore.

Oggetti lontani vengono disegnati a bassa risoluzione.



Mesh – Semplificazione

Impostori – Billboard

- spesso si ricorre a dei “trucchi” per evitare di sovraccaricare la GPU con un numero eccessivo di oggetti
- si rappresenta l’oggetto da inserire nella scena come una semplice immagine 2D anziché rappresentarlo come mesh poligonale 3D
- Tale immagine viene usata come texture per un poligono (in genere un rettangolo)
- Se si posiziona tale poligono in modo opportuno, ruotandolo rispetto all’osservatore, l’oggetto sembrerà inserito geometricamente nella scena
- pertanto occorre che il poligono non venga mai visto di taglio, deve essere sempre orientato frontalmente rispetto all’osservatore

Esempio di rendering
di alberi “impostori”
sul terreno

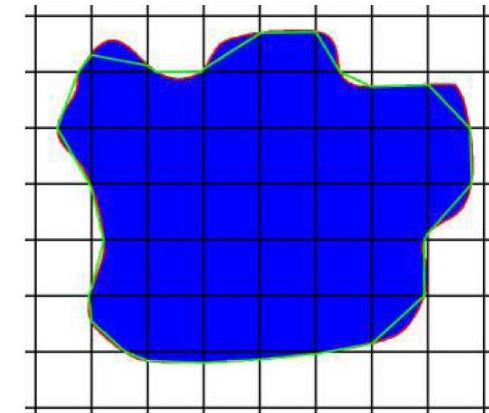
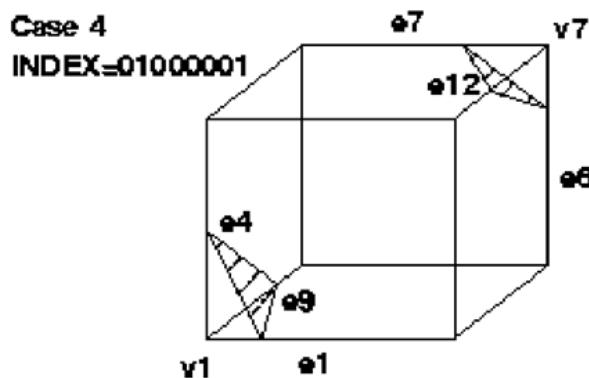
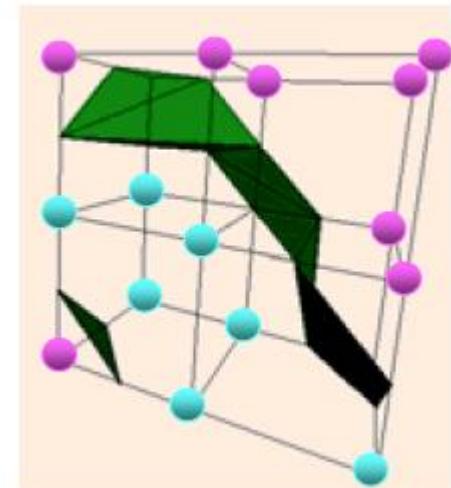


Mesh – Marching cubes

Algoritmo “marching cubes” (Lorensen e Cline, 1987) per la generazione di un mesh di triangoli a partire da una rappresentazione implicita o volumetrica con voxel (enumerazione per occupazione spaziale)

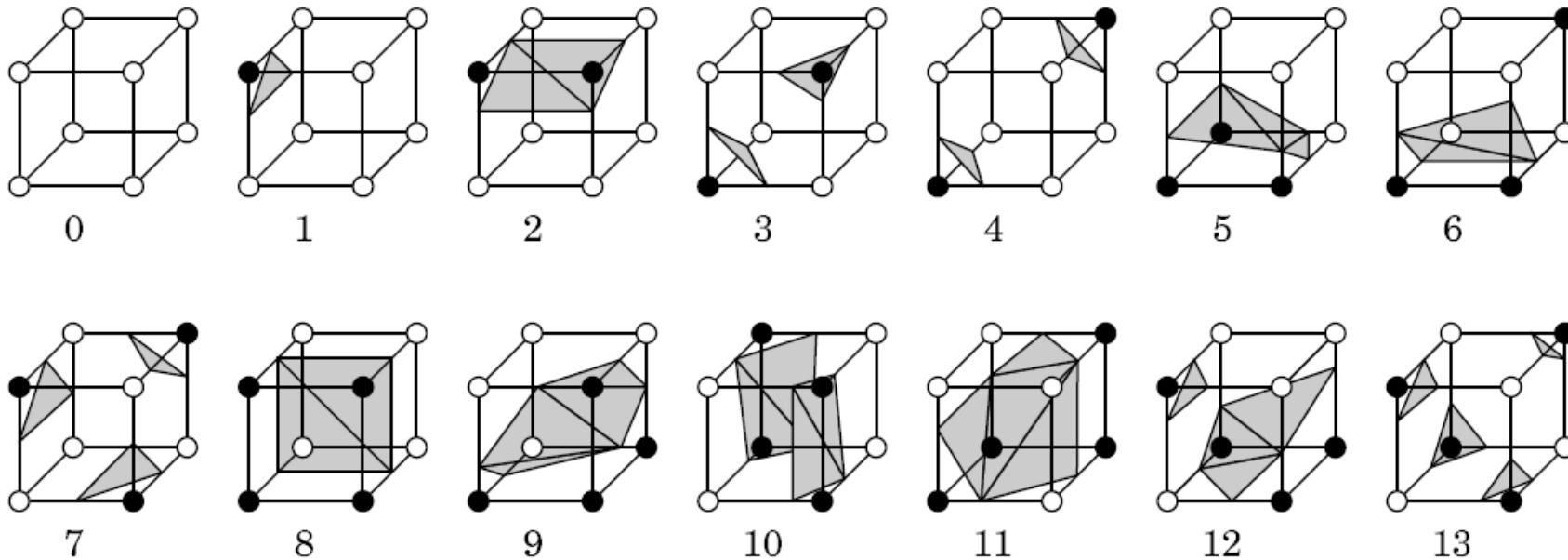
L’algoritmo “marcia” da un voxel all’altro calcolando quali vertici di ciascun voxel sono interni o esterni alla superficie del solido.

Ciascun cubo viene indicizzato con un numero binario di 8 cifre, classificato come appartenente ad una categoria (vedi prossima slide) e triangolarizzato



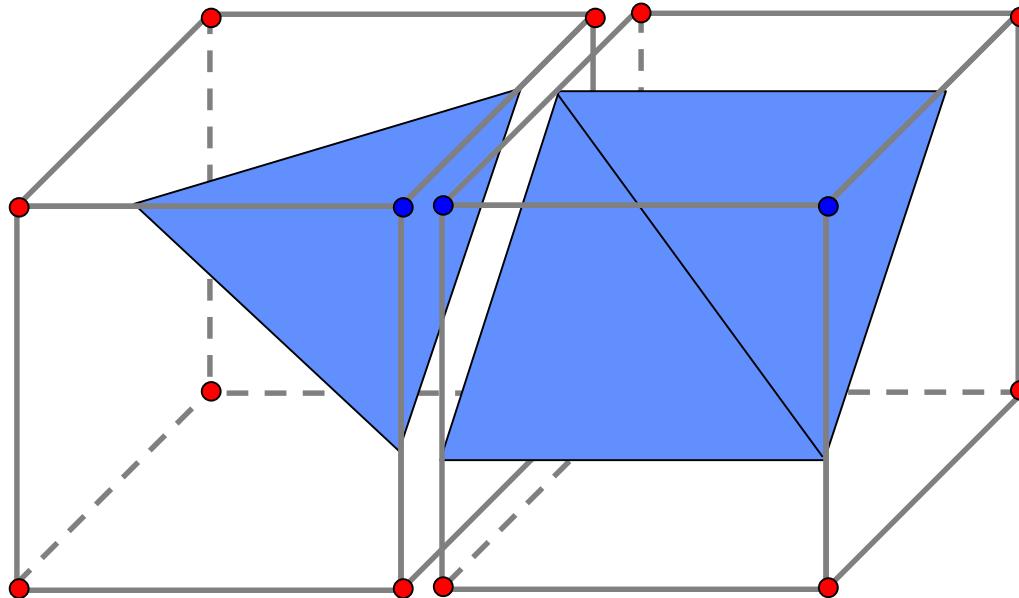
Mesh – Marching cubes

L'algoritmo di estrazione di triangoli utilizza una look-up table predefinita che contiene tutte le possibili categorie (vantaggioso dal punto di vista computazionale). Ci sono 256 possibili casi, riconducibili a 14 per simmetria



Nell'implementazione più semplice i vertici dei triangoli del mesh vengono collocati nel mezzo degli spigoli, altrimenti sono interpolati

Mesh – Marching cubes

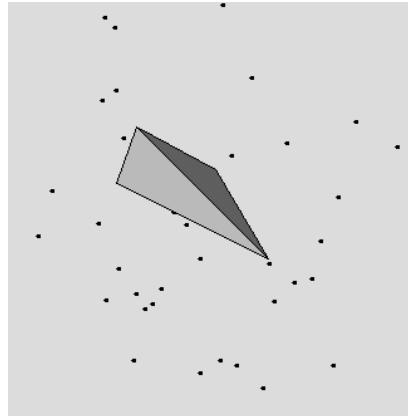


Mesh – Convex Hull

Il **Convex Hull** (inviluppo convesso) di un insieme N di punti 3D è il più piccolo insieme convesso che li contiene

Algoritmo incrementale K.L. Clarkson and P.W. Shor in 1989 complessità $O(n^2)$

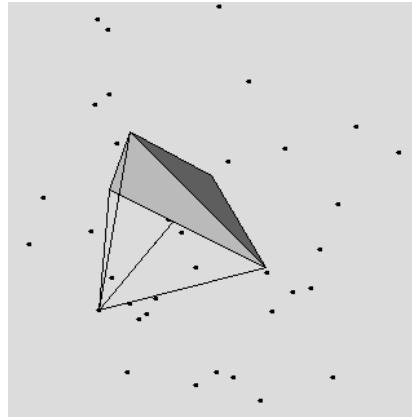
1. *Si costruisce il tetraedro con i primi 4 punti di N*



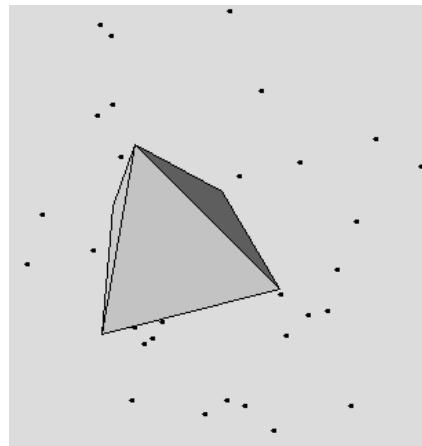
2. *Si eliminano i punti che cadono all'interno del convex hull corrente. Si sceglie uno dei punti p rimanenti e si determinano le facce visibili da p , come se in p ci fosse una luce che illumina il mesh*

Mesh – Convex Hull

3. *Si determinano gli spigoli dell' "orizzonte", cioè gli spigoli che separano una faccia visibile da una faccia non visibile da p .*
4. *Per ogni spigolo dell'orizzonte si costruisce una nuova faccia (triangolo) connettendo uno spigolo dell'orizzonte con p*



5. *Si eliminano tutte le facce che erano visibili e si torna al punto 2 se ci sono altri punti da elaborare*



Mesh – Convex Hull

