

Grafica Computazionale

Meshes

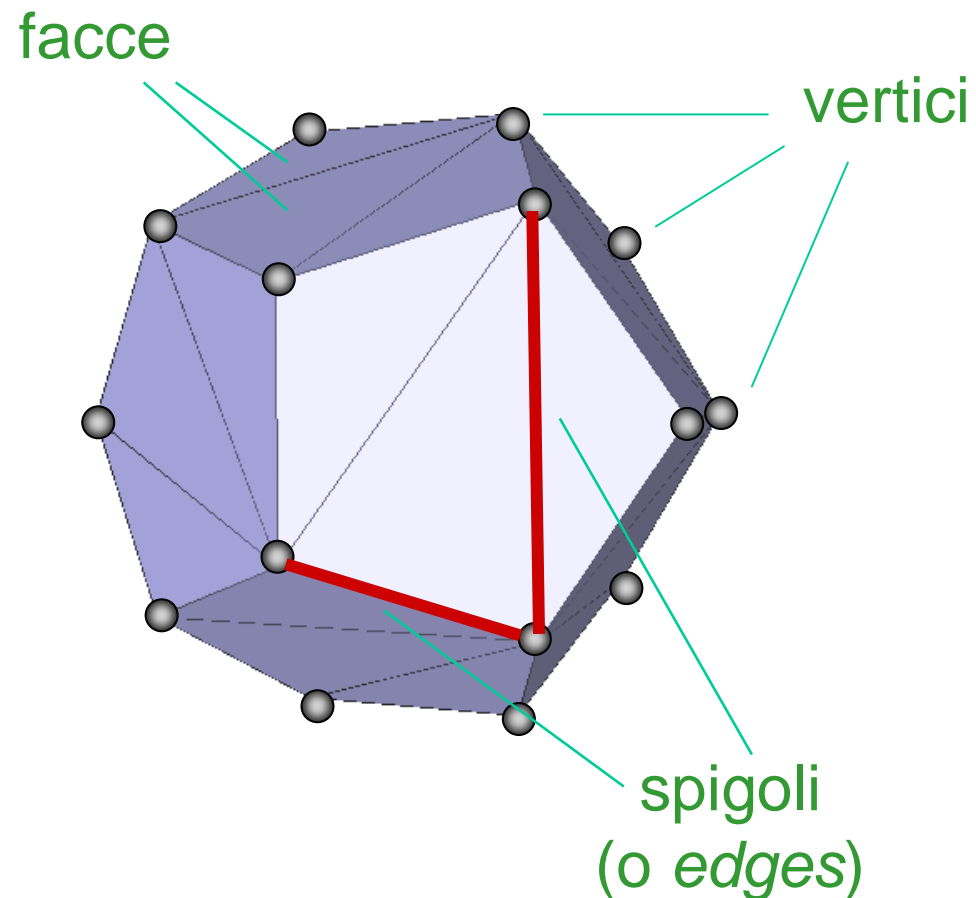
Fabio Ganovelli

fabio.ganovelli@isti.cnr.it

a.a. 2005-2006

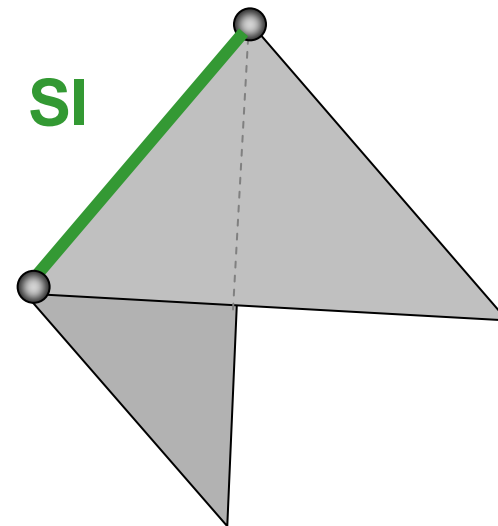
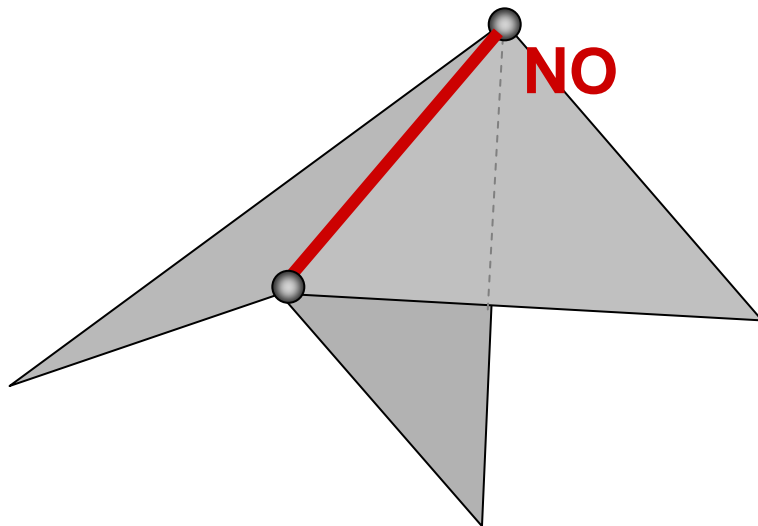
Mesh triangolare (o mesh simpliciale)

- Una mesh è un insieme di triangoli adiacenti



Caratteristiche topologiche di una mesh

- **Two Manifold** ("varietà due") o no
 - se si ogni **edge** è condiviso da al max due **facce**
 - two manifold = bene
 - non two manifold = male
 - molti algoritmi su mesh richiedono che la mesh sia two-manifold

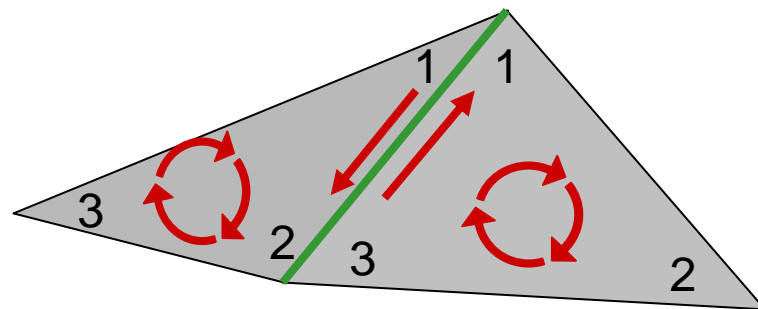


Caratteristiche topologiche di una mesh

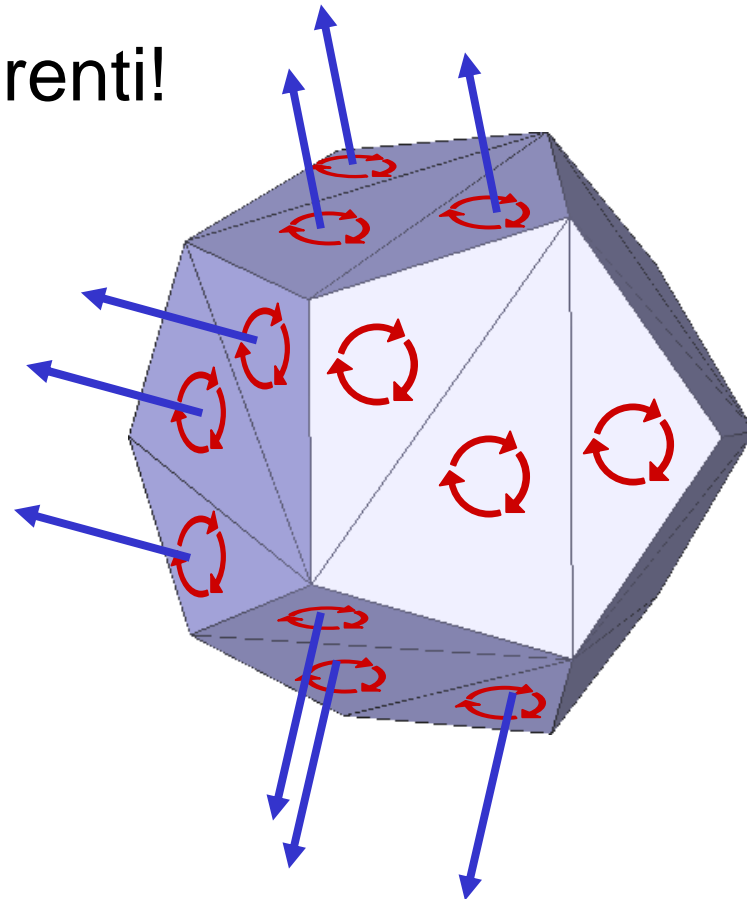
- Chiusa o aperta
 - se chiusa, ogni edge è condiviso proprio due facce

Caratteristiche topologiche di una mesh

- **Orientabile, non orientabile**
 - è possibile assegnare un orientamento ad ogni faccia coerentemente?
 - orientabile = normali coerenti!



senso opposto, edge coerente

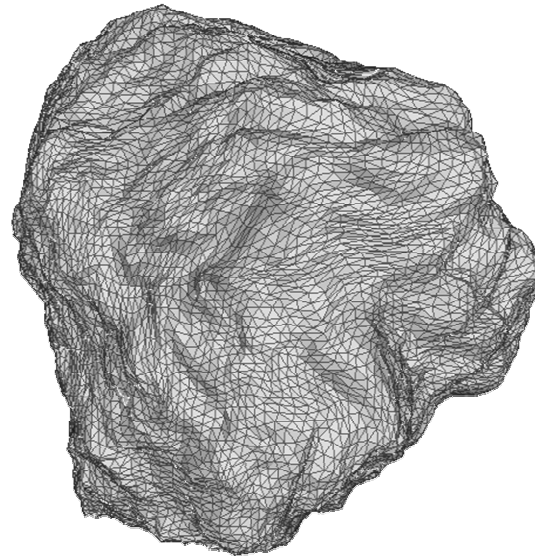
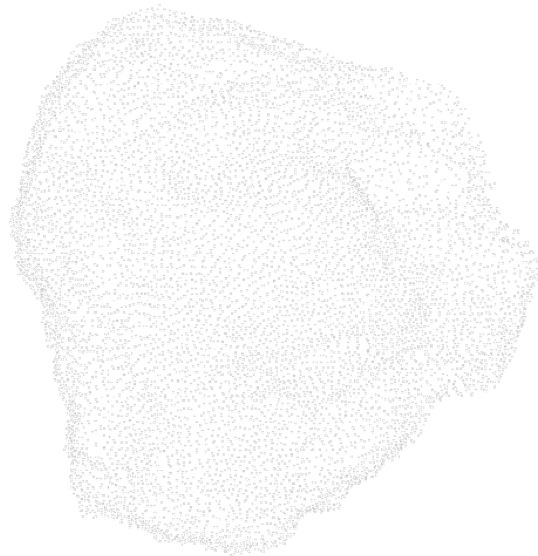


Come definisco una mesh?

- Una mesh è un insieme di triangoli adiacenti
- Come la definisco?
- Modo diretto:
 - un vettore di triangoli
 - e per ogni triangolo tre vertici
 - e per ogni vertice tre coordinate
- Piuttosto poco efficiente

Come definisco una mesh?

- Modo **indexed**
 - Lista ordinata di vertici
 - per ogni vertice la posizione
 - Lista ordinata di facce
 - per ogni faccia, 3 indici di vertici



E gli attributi?

- Posso definirli:
 - per **vertice**
 - scrivo un attributo dopo ogni vertice
 - per **faccia**
 - scrivo un attributo dopo ogni faccia
 - per **wedge** (vertice di faccia)
 - scrivo tre attributi dopo ogni faccia
- Attributi più comuni:
 - Colore, normale, coordinate texture

Esempio: formato PLY

- E' un formato digitale per mesh
- Può essere in **binario**, o in **ASCII** (testo)
 - binario: più compatto e veloce da leggere
 - ascii: umanamente leggibile con un editore di testo
- In ogni caso, comincia con un header in ASCII

Esempio: formato PLY

- Esempio:

cubo.ply

```
ply
format ascii 1.0
comment proprio un cubetto
element vertex 8
property float x
property float y
property float z
element face 12
property list uchar int vertex_indices
end_header
```

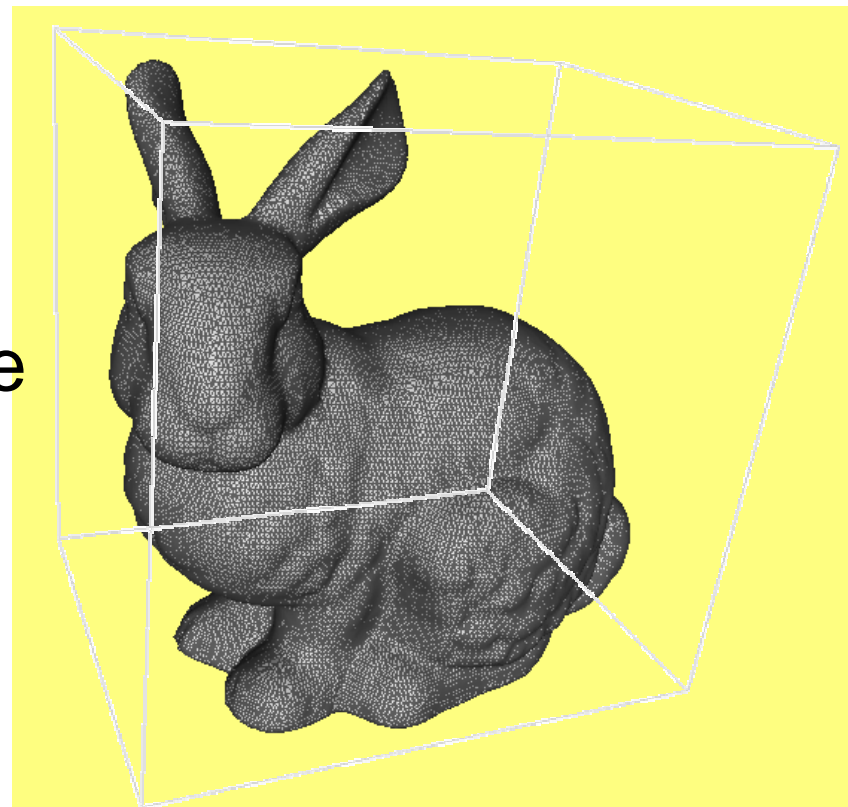
Esempio: formato OFF

- Esempio:

	# facce	# edges		LetteraL.ply	
	OFF			1 5 1	
# vertici	12	10	40	0 5 1	
	0 0 0		← indice 0	4 3 2 1 0	prima faccia: 4 vertici: con indici 3, 2, 1 e 0
x,y,z	3 0 0		← indice 1	4 5 4 3 0	
2ndo vert	3 1 0		← indice 2	4 6 7 8 9	
	1 1 0		← indice 3	4 6 9 10 11	
	1 5 0			4 0 1 7 6	
	0 5 0			4 1 2 8 7	
	0 0 1			4 2 3 9 8	
	3 0 1			4 3 4 10 9	
	3 1 1			4 4 5 11 10	
	1 1 1			4 5 0 6 11	

Mesh: task comuni

- Data una mesh:
 - magari appena caricata
- trovare il **bounding box**
 - utile ad esempio per translare e scalare l'oggetto opportunamente
 - come si fa?
 - (si itera sui vertici: trovare il max e il min di tutte le x, le y e le z)



Mesh: task comuni

- Data una mesh:
 - magari appena caricata
- trovare le **normali per vertice**
 - come si fa?
 - 1 azzerare tutte le norm x vertice
 - 2 iterare ogni faccia:
 - trovare normale
 - normalizzare normale
 - aggiungere normale al vertice
 - 3 iterare ogni vertice:
 - normalizzarlo

Task più difficili

- Bounding sphere
- Stripification
- Parametrizzazione
- Simplificazione automatica
- Detail recovery
- ...

Task più difficili

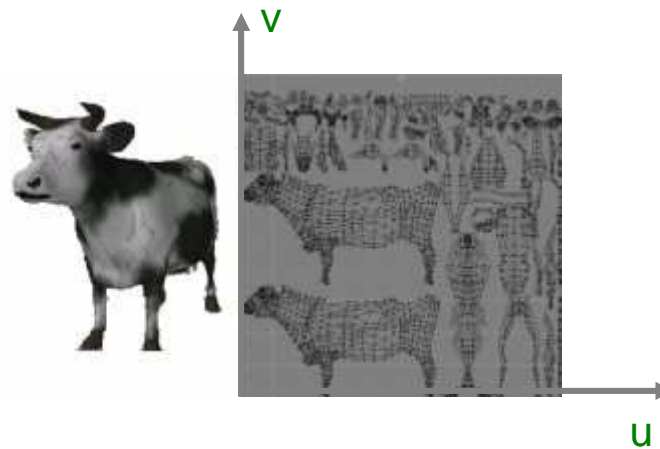
- Stripification
 - suddividere i triangoli in **triangle strips**



Task più difficili

- Parametrizzazione

- assegnare una coppia di coordinate texture ad ogni vertice
- ci sono **seams (cuciture)**
 - replicare i vertici
 - memorizzare le text coord per **wedge**

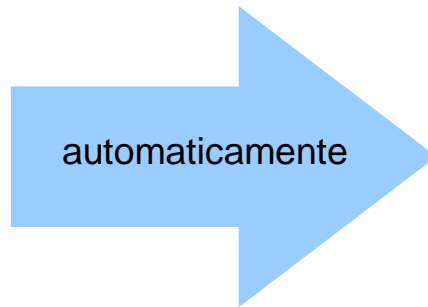


Task più difficili

- Semplificazione automatica
 - parametri:
 - un errore massimo
 - o un numero di facce obiettivo



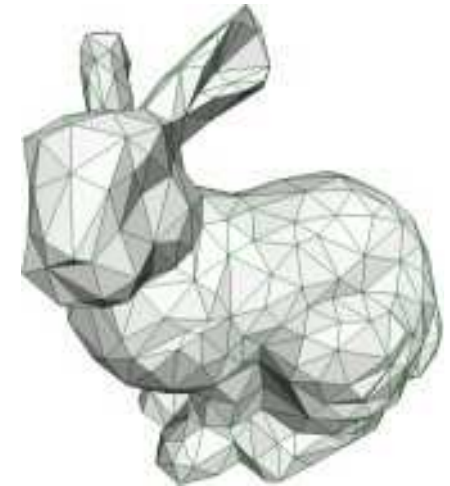
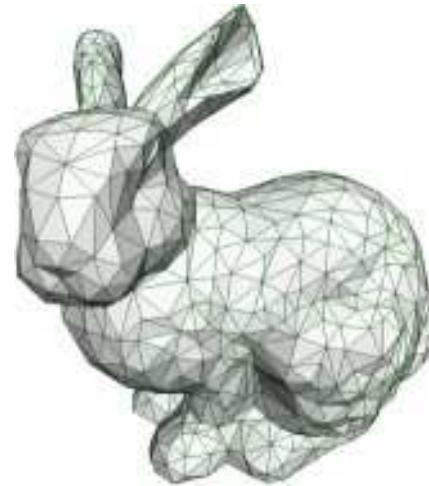
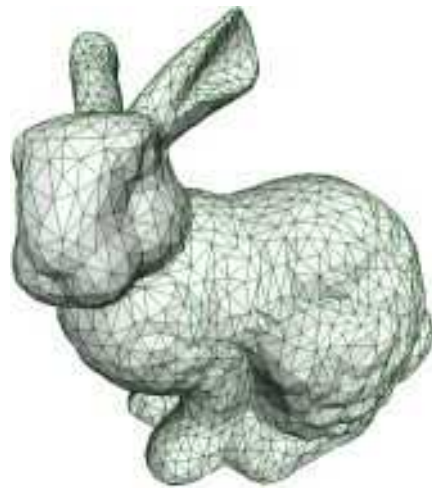
mesh originale
500K triangoli



mesh semplificata
2K triangles

Semplificazione automatica

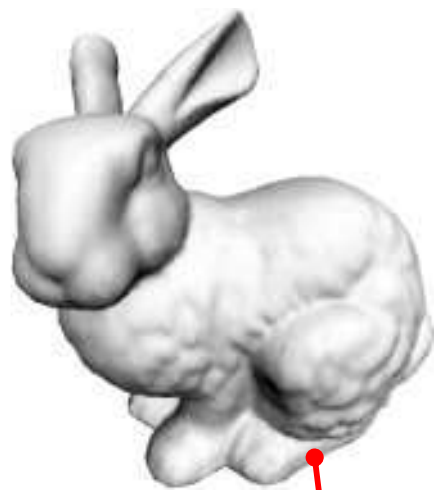
performance



quality

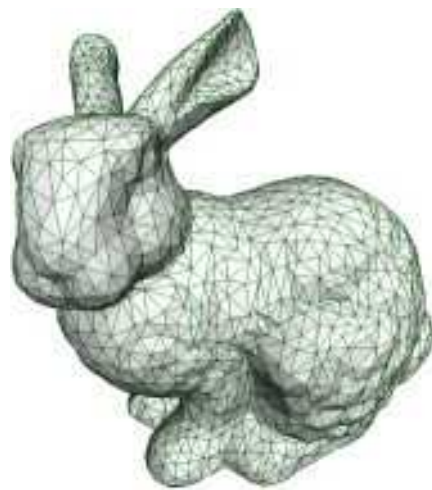
Semplificazione automatica

Una piramide di Livelli di Dettaglio

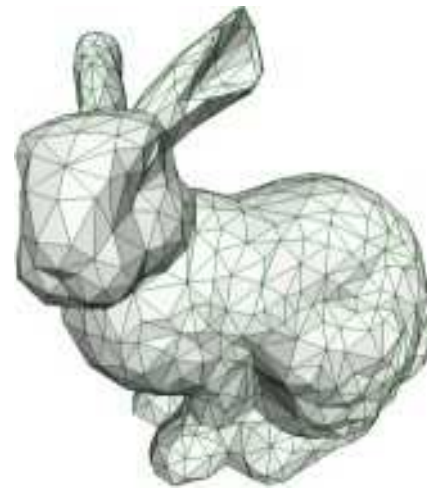


LOD 1

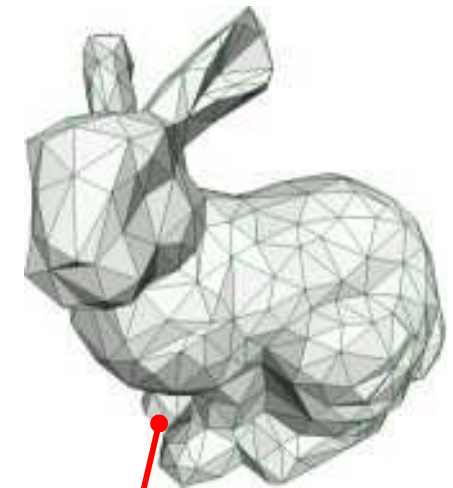
usare quando
visto da vicino



LOD 2



LOD 3



LOD 4

usare quando
visto da lontano

Detail preservation (o texture for geometry)

- Idea:
 - sintetizzare una tessitura
 - per ripristinare il dettaglio perso durante la semplificazione

500mila
triangoli



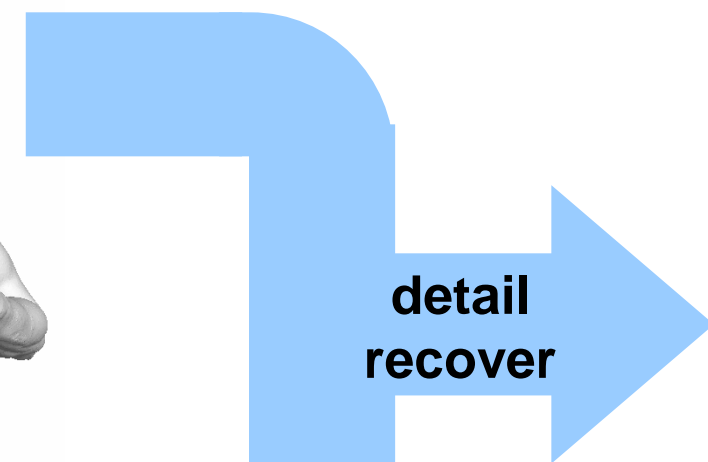
semplificazione
automatica



2mila
triangoli



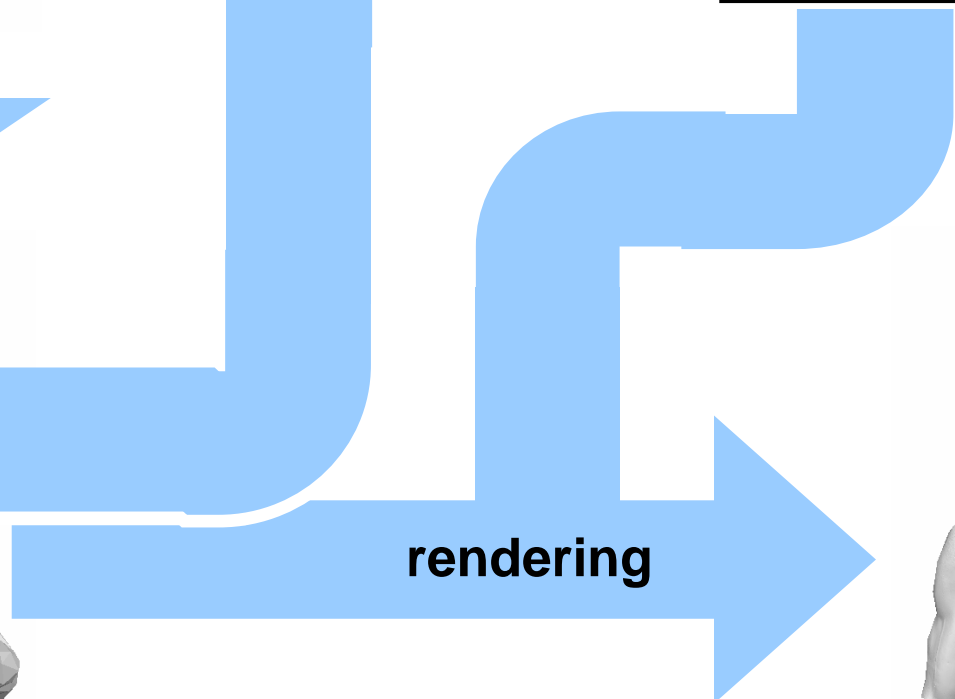
**detail
recover**



TESSITURA
fatta apposta



rendering



sempre duemila triangoli, ma con texture mapping



originale
500K triangles



semplificato ma con tessitura
2K triangles