


Marco Tarini - Computer Graphics 2024/2025
Università degli Studi di Milano

Mesh Poligonali:
Textures & UV mapping 1/2

The slide features a green header with the author's name and university. Below it, a white rounded rectangle contains the title 'Mesh Poligonali: Textures & UV mapping 1/2'. The main content area shows a wireframe teapot on the left and a textured vase on the right. The background is a light gray with a faint watermark of the University of Milan logo, which includes the text 'STUDIORUM' and 'DIOL'.

229

Texture: multirisoluzione (MIP-map)

The diagram shows a pyramid of texture levels for a stone wall. The largest level is labeled '1024x1024'. The next level down is '512x512'. The third level is '256x256'. The smallest level is '1x1', indicated by an arrow pointing to the top-right corner of the pyramid. The texture is a stone wall.

Piramide di livelli di dettaglio
detta MIP-map



230

Tessiture e MIPmap level

- ✓ In quanto immagine raster, una tessitura si presta a semplici strutture multirisoluzione:
 - ⇒ Basta ridurre la risoluzione dell'immagine per ottenere un «livello» diverso di una piramide di livello di dettaglio
 - ⇒ Come si riduce la risoluzioni dell'immagine? Vedere corso di image processing (hint: interpolando fra i pixel / i texel)
- ✓ Struttura chiamata «**MIP-map**» (per le texture):
Una piramide di livelli di dettaglio:
 - ⇒ Livello 0: la tessitura originale, risoluzione massima
 - ⇒ Livello 1: una tessitura ridotta ad $\frac{1}{2}$ della risoluzione in entrambe le dim
 - ⇒ Livello 2: riduzione del livello 1 (come sopra)
 - ⇒ ...
 - ⇒ Livello N: tessitura 1×1 (per tessiture quadrate)
 - ⇒ Nota: in tutto sono $\log_2(\text{RES})$ livelli,
 - ⇒ Nota: ogni livello occupa $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ del precedente
 - ⇒ Nota: in tutto, lo spazio totale occupato da tutti i livelli oltre lo 0 è solo $\frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \frac{1}{256} + \dots \rightarrow \frac{1}{3}$ quello del livello 0



231

Tessiture e supporto Hardware

- ✓ Le tessiture sono estremamente utilizzate nella maggior parte dei contesti applicativi in cui si usino mesh poligonali
 - ⇒ Come: games, VR, movies, commercio elettronico, applicazioni architettonali, sci vis, beni culturali...
 - (ma spesso non: CAD, applicaz medicali)
- ✓ Parte della loro diffusione è dovuta al supporto HW:
 - ⇒ le GPU (il processore grafico) sono progettate proprio per renderizzare mesh triangolari con tessitura associata
 - ⇒ Livello di MIP-map compresi!
 - ⇒ I limiti e le caratteristiche tipiche di una tessitura (num di canali ≤ 4 , risoluzione massima, bit depth, risoluzione come potenza di 2) sono imposti dalle caratteristiche dell'HW



232

Texture = immagine rasterizzata? Le differenze

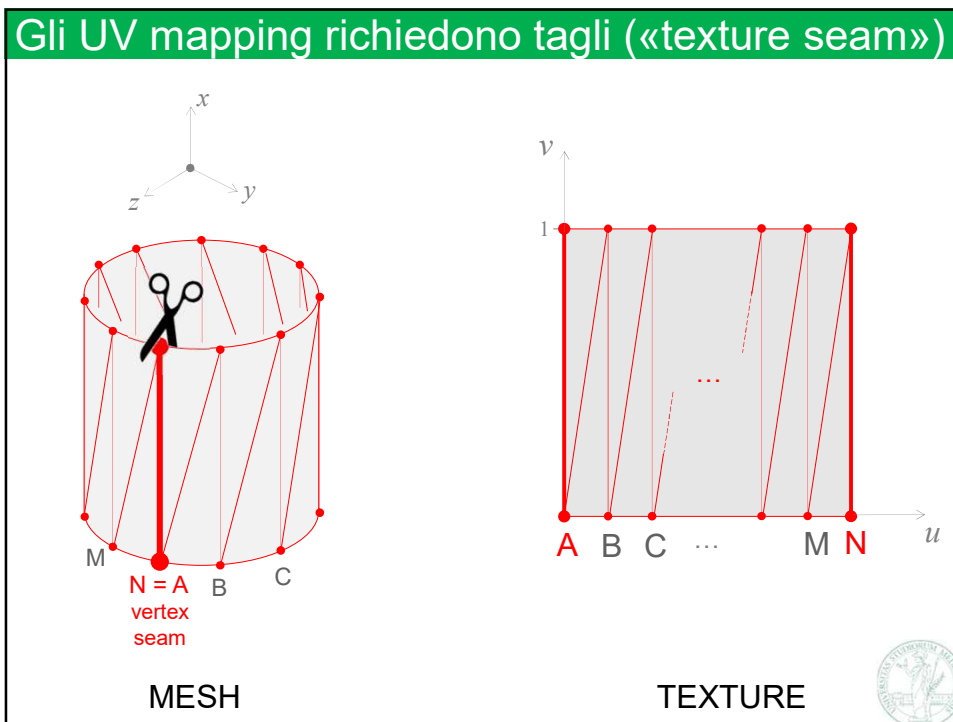
- ✓ Una texture 2D è del tutto analoga ad un'immagine rasterizzata. Perché usare un termine separato per identificare questo tipo di immagine?
- ✓ Diversi in...
 - ⇒ **uso atteso**: una texture è pensata esplicitamente per essere rimappata su una mesh (attraverso un UV-map)
 - ⇒ **contenuto**: i texel di una texture campionano qualsiasi segnale sulla superficie (normale, colore, proprietà del **materiale** (per es, coefficiente di lucidità...) etc), laddove i pixel di un'immagine sono campioni di **colore**, di solito rappresentati da una tripla di valori R,G,B
- ✓ Esistono formati file specifici per texture (come dds), diversi da quelli per immagini generiche (come png o jpeg), in termini di...
 - ⇒ **numero di canali**: una texture standard ha 4 canali (a volte: 1 o 2), un'immagine tipicamente ne ha 3
 - ⇒ **risoluzione**: per motivi storici inerenti al rendering, una texture ha risoluzione per lato come potenza di 2 (es: 512, 1024, 2048, 4096)
 - ⇒ **profondità di texel**: ogni canale di una texture ha un numero di bit per canale variabile da 4 a 16, molti formati per immagine prevedono solo 8bit per canale
 - ⇒ **compressione**: una tessitura deve essere random-accessible anche da compressa. Esistono *schemi di compressione per texture*, diversi dagli schemi usati per le immagini (che non hanno bisogno di essere random-accessible)
 - ⇒ **multi-risoluzione**: una tessitura spesso comprende un tipo di piramide di livelli di dettaglio detta «MIP-map»: ogni livello successivo dimezza la risoluzione del precedente (es: 1024x1024, 512x512, 256x256, ... , 2x2, 1x1)
- ✓ Un formato per immagine generica (come png) può essere comunque usato come formato di interscambio per texture

233

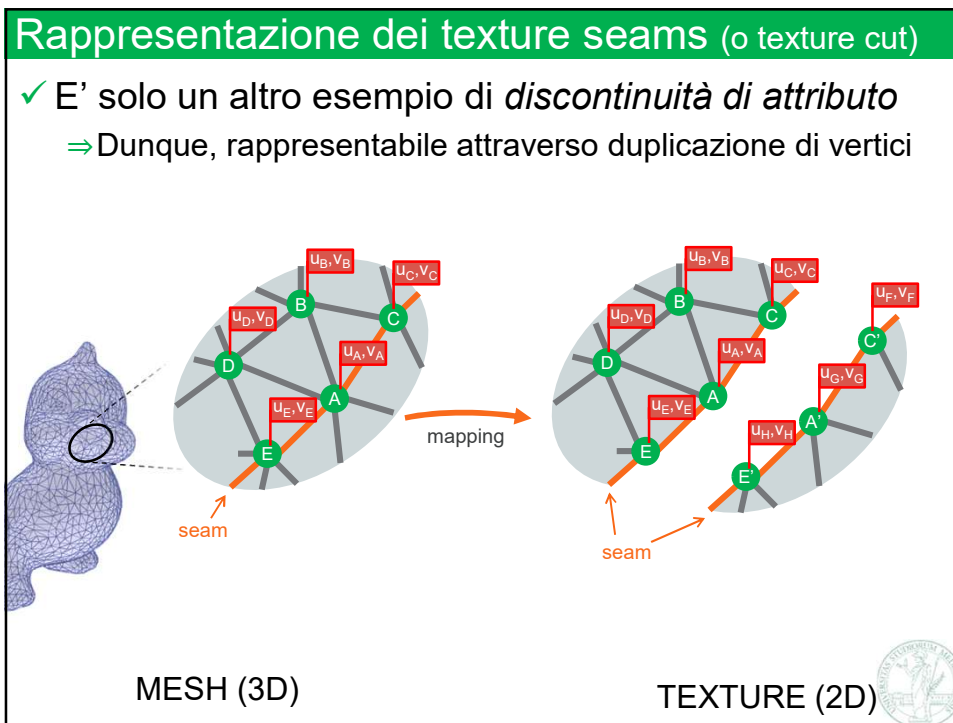
Mesh parametrization (task di geometry processing)

- ✓ **UV-mapping di una mesh, o mesh parametrization**:
il task di creare texture seams e assegnare coordinate UV ad ogni vertice
 - ⇒ A volte, (raro) la mesh viene creata congiuntamente al suo UV map
 - ⇒ Più spesso, la mesh creata (es. acquisita, o modellata, o estratta da un altro tipo di dato 3D) sprovvista di un UV-map
- ✓ Task non semplice
 - ⇒ Automatizzazione: tema molto studiato nel Geometry Processing
 - ⇒ Artisti digitali intervengono manualmente con software di 3D modelling (es Blender, Maya...)
- ✓ Un «buon» UV map deve rispondere a molti criteri:
 - ⇒ Inietività: ogni punto della tessitura può essere mappato in al più un punto della mesh (cioè: nessun triangolo è sovrapposto in spazio UV)
 - ⇒ Bassa distorsione: ogni triangolo di mesh T_3 deve essere mappato in un triangolo di tessitura T_2 di *forma simile* e di *area proporzionale*
 - ⇒ Buon coverage della tessitura: le parti di tessitura non coperte da nessun triangolo rappresentano uno spreco di memoria e vanno minimizzate

234



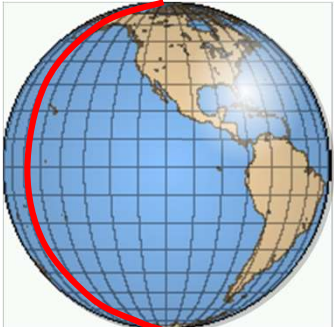
235




236

Tagli (cuts) o texture "seams"


✓ Sempre necessari se la mesh è chiusa



two-manifold chiuso
in 3D



spazio parametrico 2D



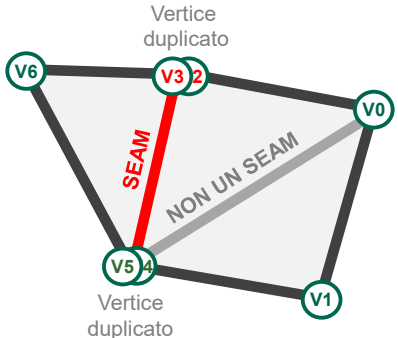
237

Texture seams (o anche texture "cuts")

✓ I seams sono scomodi ma necessari!


	X	Y	Z	U	V	...
V0	p_x0	p_y0	p_z0	$u0$	$v0$...
V1	p_x1	p_y1	p_z1	$u1$	$v1$...
V2	p_x2	p_y2	p_z2	$u2$	$v2$...
V3	p_x2	p_y2	p_z2	$u3$	$v3$...
V4	p_x3	p_y3	p_z3	$u4$	$v4$...
V5	p_x3	p_y3	p_z3	$u5$	$v5$...
V6	p_x4	p_y4	p_z4	$u6$	$v6$...

GEOMETRIA + ATTRIBUTI



Tri:	Wedge 1:	Wedge 2:	Wedge 3:
T0	0	1	4
T1	4	2	0
T2	5	3	6


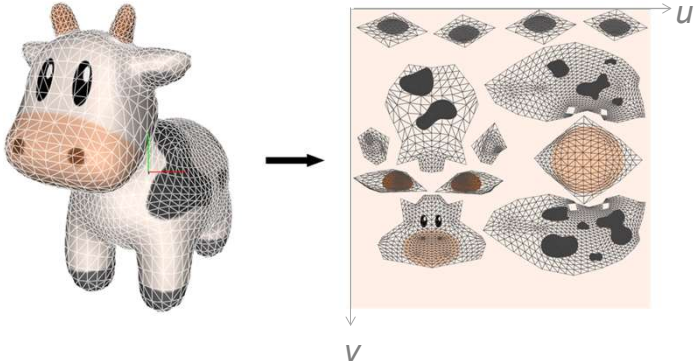
CONNETTIVITA'



238

Texture Atlas

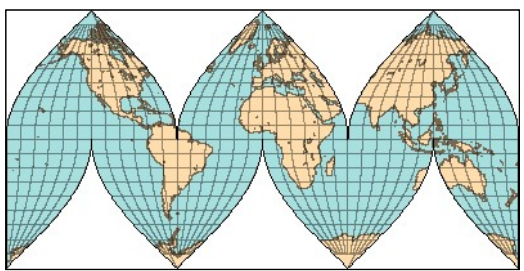
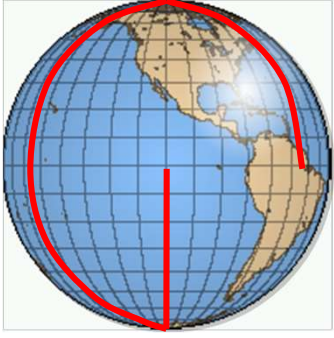
- Tipo di UV map in cui mesh viene divisa in zone (patches), ciascuna rimappata in un "isola" della tessitura.
- (il nome proviene dall'analogia con un atlante di mappe)



239


Tagli (cuts) o texture "seams"

✓ Più tagli \Rightarrow meno distorsione (di solito)

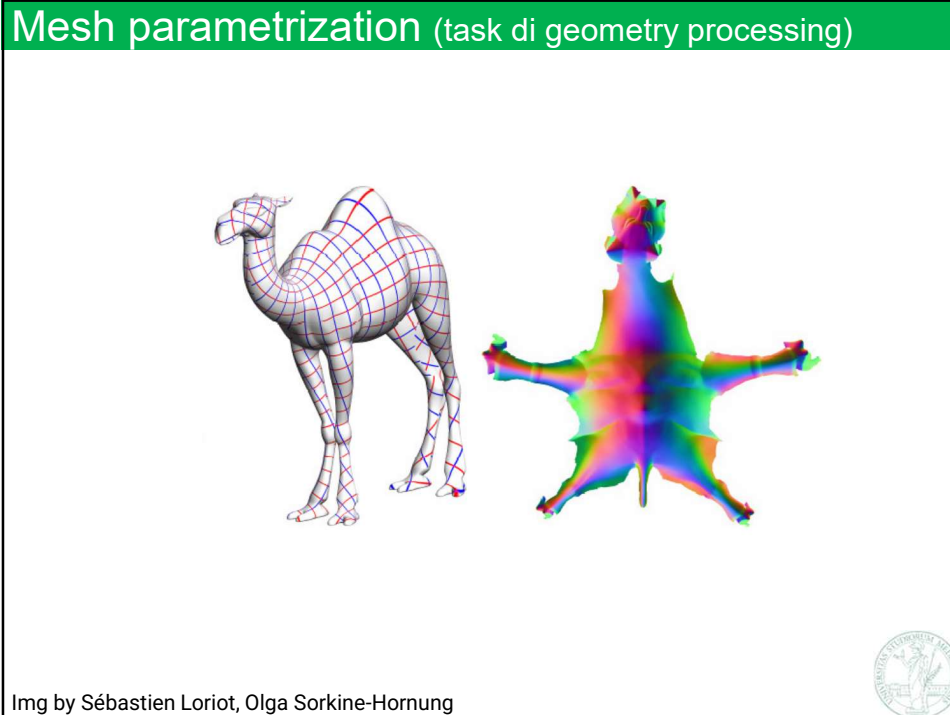


two-manifold chiuso
in 3D

spazio parametrico 2D




240



241

Mesh parametrization (task di geometry processing)

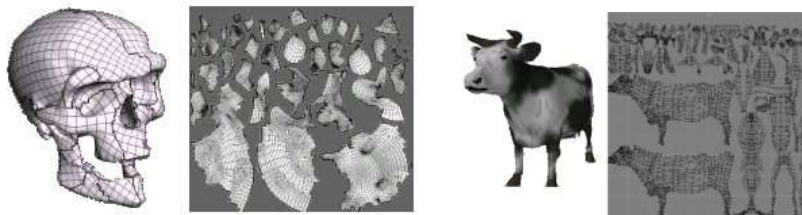
- ✓ Il task:
 - ⇒ Data: una mesh (two-manifold) di input
 - ⇒ Produrre: il suo UV-map (cioè un assegnamento delle texture coordinates u-v per ogni vertice)
- ✓ Richiede di risolvere sottoproblemi come:
 - ⇒ Dove introdurre i tagli (i texture seams, o texture cuts)
 - ⇒ Come stendere («unwrap») la superficie sul piano
 - ⇒ Nelle tessiture ad atlante (atlas): come comporre le varie isole nel rettangolo della tessitura («packing»)
- ✓ Task chiamato anche:
 - ⇒ **UV-mapping** di una mesh (specie in ambiente industriale – che sia fatto ad opera di un algoritmo automatico O di un artista digitale umano)
 - ⇒ Automatic Mesh **parametrization** (in ambiente accademico)



242

Geometry processing: mesh parametrization

- ✓ Task: costruire automaticamente un UV-map per una mesh data
 - ⇒ problema difficile da risolvere in modo soddisfacente (soprattutto la scelta sul posizionamento/numero dei tagli)
 - ⇒ un gran numero di approcci algoritmici diversi
 - ⇒ rimane problema aperto: gli UV-map prodotti da artisti digitali sono spesso di qualità migliore



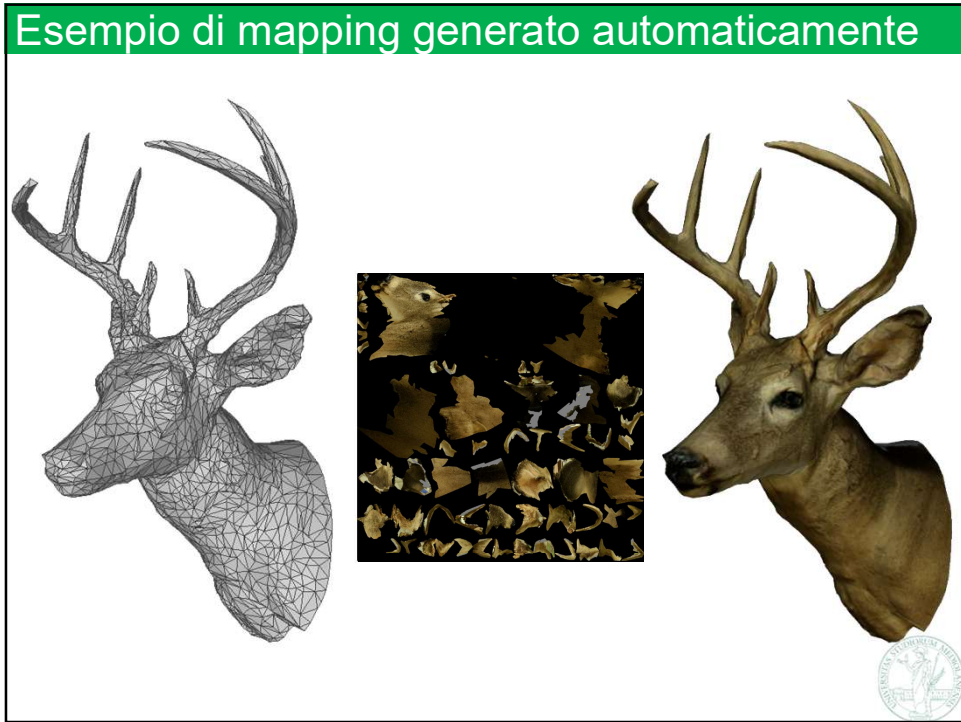
243

Texture seams (o cut)

- ✓ Nozione intuitiva:
 - ⇒ il task di mesh parametrization ricorda quello di sbucciare una mela e disporre la buccia dentro un rettangolo
 - ⇒ Simile: come costruire un planisfero, cioè la mappa (2D) della superficie del pianeta (sfera in 3D) (che è un problema storicamente molto studiato: es. vedi: <http://vcg.isti.cnr.it/~tarini/spinnableworldmaps/>)
 - ⇒ La differenza è che la mesh ha una forma e una topologia arbitraria, piuttosto che sferica
- ✓ Come questa intuizione suggerisce, definire un UV-map richiede di introdurre i tagli adatti
 - ⇒ questi tagli consentono di aprire la mesh per stenderla (unwrap) su un piano



244



245



246

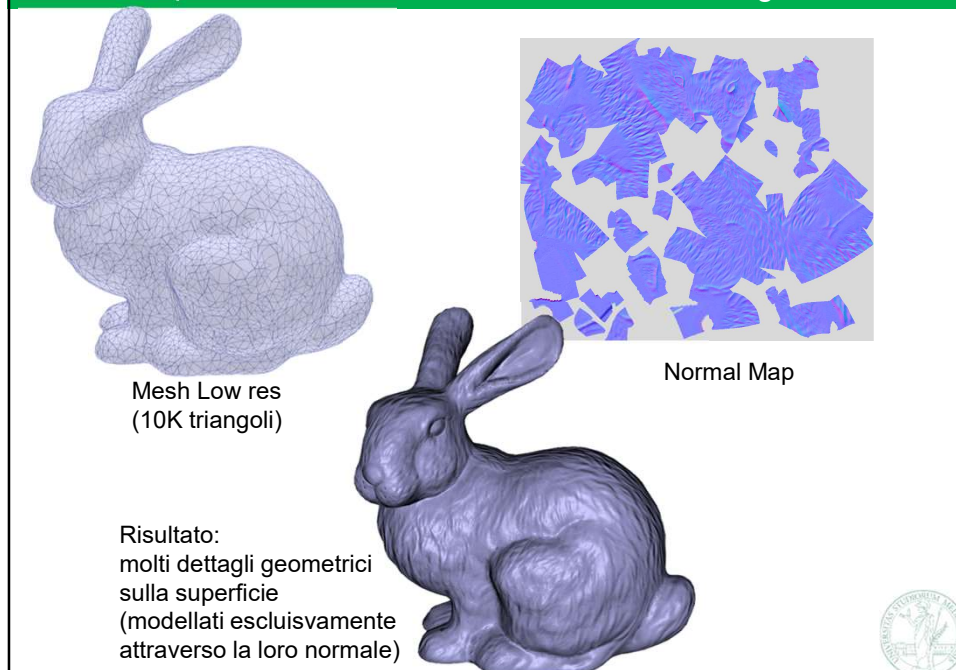
Una normale per Texel: le Normal maps

- ✓ Una **normal-map** è una texture che campiona le normali della superficie
 - ⇒ in questo modo, contribuisce a specificare la forma geometrica dell'oggetto
 - ⇒ (introducendo però una approssimazione)
 - ⇒ tipicamente, la normal map riproduce dettagli geometrici minuti, ad alta frequenza (come gli avvallamenti su una buccia di arancia), mentre la mesh riproduce la forma generale dell'oggetto (come la forma sferica dell'arancia)
- ✓ Uno stesso oggetto può essere rappresentato, con simile accuratezza e una resa simile, attraverso ...
 - ⇒ una mesh ad alta risoluzione, oppure
 - ⇒ una mesh a bassa risoluzione ma provvista tessitura per dettagli «ad alta frequenza» (come una normal-map e/o una tessitura colore)
- ✓ Quali vantaggi e svantaggi?



247

Normal map: tessitura che codifica una normale in ogni texel



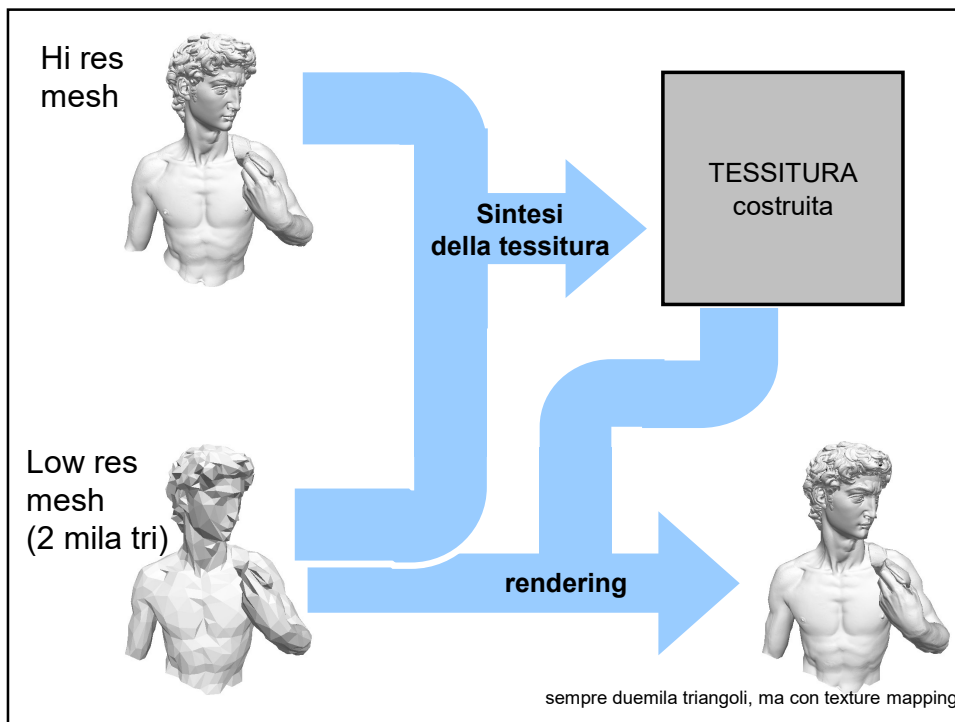
248

Textured Mesh VS Hi-resolution meshes	
Hi-res mesh (con attributi) <ul style="list-style-type: none">✓ densità di campionamento segnale = densità di campionamento geometria✓ incrementare il numero di vertici (cioè la risoluzione della mesh) ha un costo elevato: coinvolge anche la connettività, richiede di processare più triangoli✓ non richiede <i>vertex seams</i> (se il segnale è continuo)✓ la forma della superficie è modellata esplicitamente, fin nei suoi dettagli più minuti (questo permette di tenerne conto in tutti i tipi di processing)	Low-res mesh con texture <ul style="list-style-type: none">✓ risoluzione texture (colore, etc) indipendente da risoluzione mesh (geometria)✓ incrementare il numero di texel (cioè la risoluzione della tessitura) ha un costo piccolo✓ costo extra: UV-map sulla mesh. Va costruito (difficile); va mantenuto in memoria; richiede l'introduzione di vertex seams.✓ ma, in totale, meno onerosa in termini di spazio (memoria), e tempo (di rendering)✓ normal-map: i dettagli di forma sono riprodotti solo visualmente attraverso il lighting

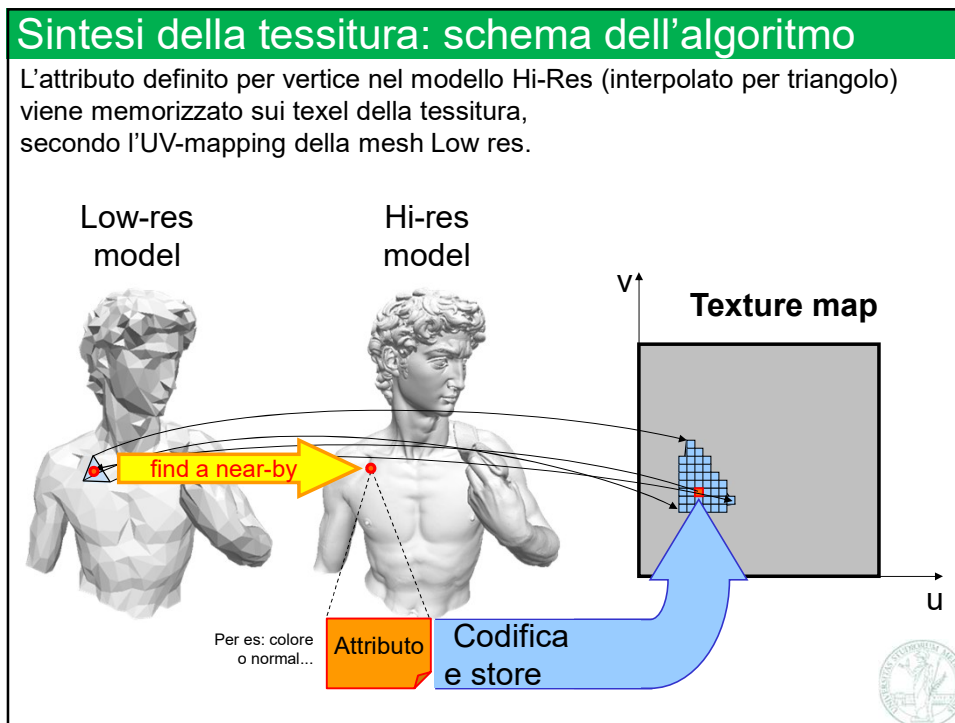
249

Texture «baking» (traduzione letterale: cuocere al forno)
<ul style="list-style-type: none">✓ Sintesi automatica di una tessitura, per riprodurre un attributo definito su una mesh hi-res✓ Input:<ul style="list-style-type: none">⇒ Mesh Hi-res M0, con attributi (per vertice, per faccia...) che rappresentano un segnale che ci interessa riprodurre nella tessitura (colore, normale, materiale...)⇒ Mesh Low-res M1 che approssima la stessa forma, dotata di UV-map (cioè «parametrizzata»), ma non ancora di tessitura✓ Output<ul style="list-style-type: none">⇒ una tessitura sintetizzata (baked texture) per M1 che riproduce su di essa l'attributo definito su M0✓ Un passaggio comune nel pipeline di creazione dei modelli 3D<ul style="list-style-type: none">⇒ Supportato per es da tutte le suite di modellazione 3D

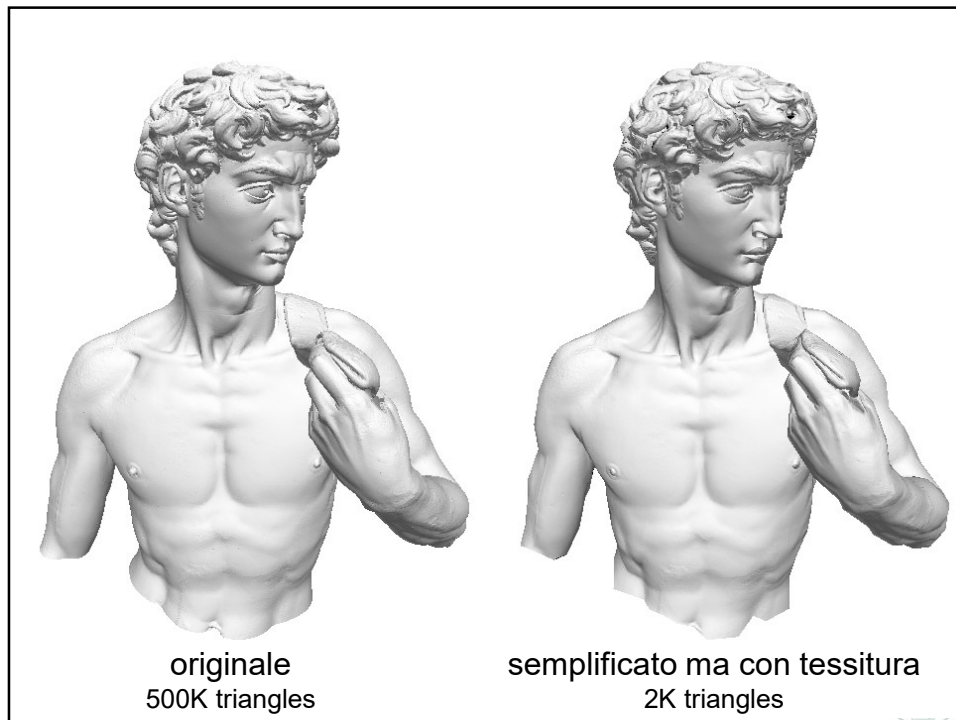
250



251



252



253

Osservazione: geometria e illuminazione

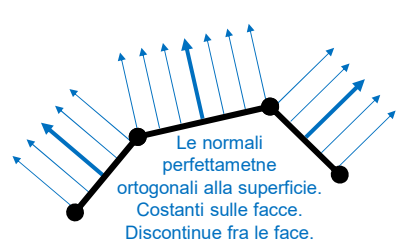
- ✓ Come ci eravamo già resi conto nel caso dello smooth shading, nel rendering l'aspetto percepito delle superfici è dominato dall'illuminazione (lighting, il chiaroscuro cioè), che è determinata dalle normali, maggiormente che non dall'effettiva forma geometrica rappresentata dai nostri modelli
- ✓ Ad esempio:
 - ⇒ Un oggetto renderizzato con flat-shading (cioè usando normali definite per faccia) rivela la sua reale natura **piatta**, dato che le normali sono costanti, ma...
 - ⇒ visualizzato con smooth shading (cioè normali definite per vertice ed interpolate nelle facce) ci appare **curvo**, dato che le normali variano con continuità
 - ⇒ Questo ci consente di visualizzare superfici (che appaiono) curve anche usando mesh triangolari, che sono per definizione piatte «a tratti» (in ogni faccia)
- ✓ Anche nel caso nelle normal-map, il lighting è molto efficace nel suggerire una forma complessa codificata delle normali memorizzate nella tessitura
 - ⇒ Piuttosto che la forma effettiva, ben più povera di dettaglio!
 - ⇒ Ottenere lo stesso dettaglio geometrico attraverso un incremento della risoluzione della mesh sarebbe molto più oneroso (memoria, tempo di rendering)
- ✓ Si tratta, in entrambi i casi, di un'illusione
 - ⇒ rivelata, per es, dalla sagoma, o silhouette
 - ⇒ Vedremo nelle prossime lezioni strutture dati capaci di rappresentare superfici **effettivamente** curve!



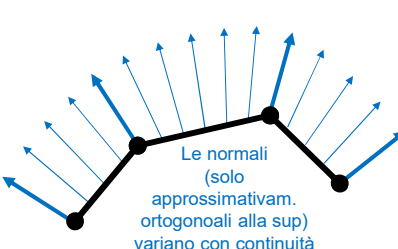
254

Superficie "effettiva" della mesh, e normali utilizzate:

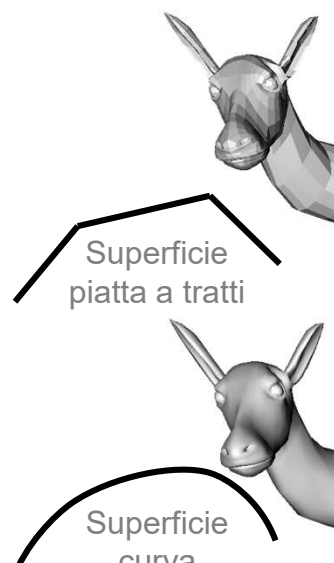
Flat shading
(normali per faccia)



Smooth shading
(normali per vertice)



Appare come:



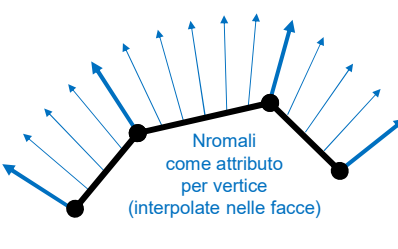
Superficie piatta a tratti

Superficie curva

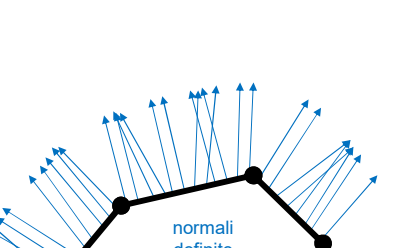
255

Superficie "effettiva" della mesh, e normali utilizzate:

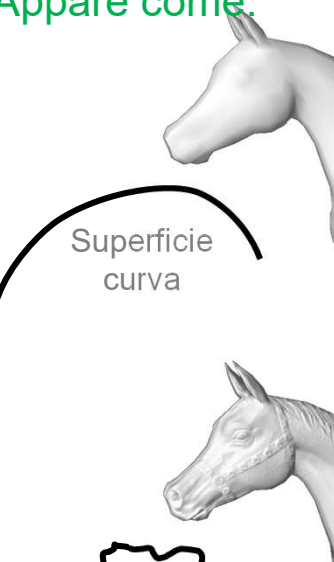
Senza texture



Con normal map



Appare come:



Superficie curva

Superficie dettagliata

256