


Marco Tarini - Computer Graphics 2023/2024  
Università degli Studi di Milano



3D Models:  
meshes (5/5)  
Textures & UV mapping



204

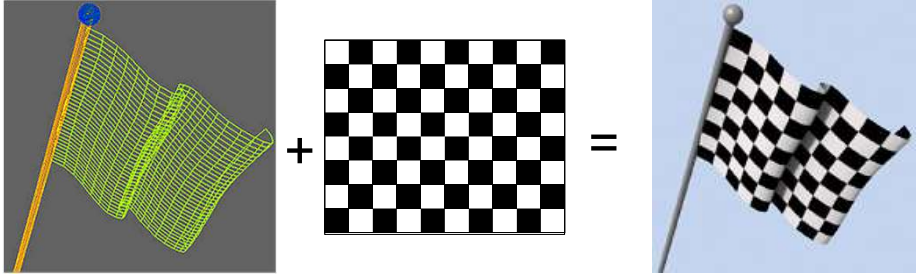
## Textures

- ✓ Spesso vorremmo poter campionare un *attributo* (un segnale: come colore, normale, lucentezza, temperatura etc) sulla superficie in modo *molto più denso* di quanto campioniamo la stessa superficie per catturare la sua forma geometrica.
- ✓ Per es:
  - ⇒ forma: ci bastano 10 campioni per cm<sup>2</sup> (sufficiente per rappresentare una statuetta in dettaglio)
  - ⇒ colore: vorremmo 100 campioni per cm<sup>2</sup> (per es, necessario per riprodurre il dettaglio pittorico su un vaso)
- ✓ Memorizzando gli **attributi per vertice** su di una mesh poligonale, ho:
  - 1 campione di segnale = 1 campione di forma = 1 vertice ☹
  - ⇒ e invece attributi per faccia, guadagno solo un fattore 2 in media (se è una mesh di triangoli – fattore 1 se è una mesh di quad)
- ✓ In 2D, le immagini “raster” (griglie 2D regolari di pixel) sono un ottimo modo di campionare segnali molto densamente
  - ⇒ es: un’immagine a 100 DPI (DPI = dots per inch<sup>2</sup>) = 40 dpcm (dots per cm<sup>2</sup>) = 40x40 (1600) campioni (cioè «pixel») per cm<sup>2</sup>
  - ⇒ es: immagine risoluzione 1000x1000 = 1 Milone di pixel = 1 «MegaPixel»
- ✓ Come ottenere una simile densità di campioni su superfici 3D?



207


### Mesh + Texture



The diagram illustrates the process of texturing a mesh. On the left, a 3D wireframe mesh of a flag is shown against a dark background. In the center, a 2D checkerboard texture is displayed. An equals sign follows, leading to the final rendered image of a checkered flag waving against a light blue sky.

Mesh  
(pure quad mesh)

Texture  
(griglia regolare di "texel" 2D)



208

### Mesh + Texture



The diagram illustrates the process of texturing a mesh. On the left, a 3D white mesh of a classical vase is shown. In the center, a texture consisting of a patchwork of images of classical vases is displayed. An equals sign follows, leading to the final rendered image of the vase with the classical vase texture applied.

Tri-Mesh

Texture  
(in questo caso:  
un patchwork di foto)



209

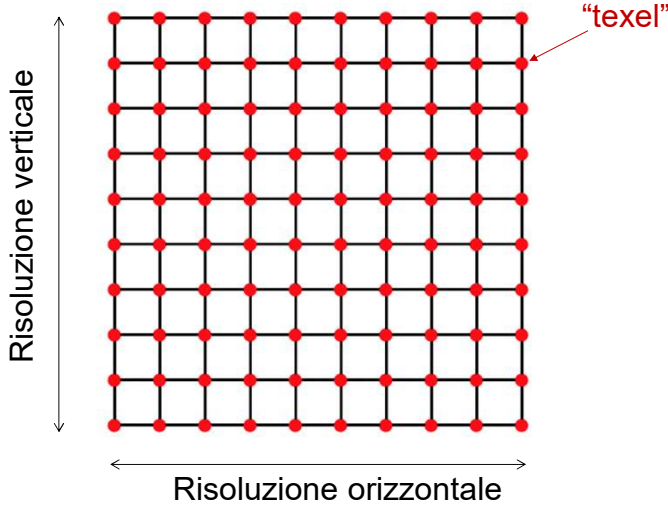
## Texture Mapping

- ✓ Idea intuitiva: incolliamo un'immagine raster (detta **texture**, o **tessitura**) alla superficie di una mesh
  - ⇒ immagine raster = grigliato regolare di **pixel** (pixel = **p**icture **e**lement)
  - ⇒ texture = grigliato regolare di **texel** (texel = **t**exture **e**lement)
- ✓ Ogni punto sulla superficie 3D **S** ← rappresentata dalla mesh viene associato ad un punto su un rettangolo 2D **T**
  - ⇒ convenzione: le coordinate 2D su **T** ← anche dette coordinate texture o parametriche sono indicate da  $(u, v)$  – a volte da  $(s, t)$  per distinguerle dalle coordinate 3D su **S** che sono  $(x, y, z)$
  - ⇒ questo mapping da **S** a **T** è detto **UV-map** o **UV-mapping** della mesh, ← in contesto industriale oppure la sua **parametrizzazione** ← in contesto accademico
- ✓ Domande a cui dobbiamo rispondere:
  - ⇒ come è memorizzata una texture
  - ⇒ come memorizzare lo UV-map sulla mesh?
  - ⇒ come viene costruito un UV-map per una data mesh? («how do we **parametrize**» or «how do we **uv-map**» a mesh)




210

## Una tessitura (texture)



(di solito, sono potenze di due: 1024, 2048, 4096...)



211

## Tessiture e texel

- ✓ Ogni pixel della tessitura, chiamato texel, memorizza valori qualsiasi che variano sulla superficie
  - ⇒ Tipicamente: colori, normali, valori di trasparenza, coefficienti di lucidità – o altri modi di descrivere un «materiale» ...
- ✓ La tessitura è del tutto analoga ad un'immagine raster
  - ⇒ Cioè un array 2D (una «matrice») di «texel» (texture elements)
- ✓ Come tale, è caratterizzata da:
  - ⇒ Una risoluzione orizzontale  $R_x$  (di solito, potenza di due  $\leq 2^{13} = 8192$ )
  - ⇒ Una risoluzione verticale  $R_y$  (idem)
  - ⇒ Numero di canali per texel  $Ch$  (tipicamente, 1,2 o 4)
  - ⇒ Numero di bit per canale  $B$  (8 (spesso), 16, 24 ...)
  - ⇒ Spazio totale occupato in memoria (espressa in *bit*) =  $R_x \cdot R_y \cdot Ch \cdot B$   
numero di texel    «bit-depth»
- ✓ In quanto immagine, può essere, per es, «dipinta» da un artista, catturata da foto, photoshopped, etc
  - ⇒ Esistono anche molte tecniche per sintetizzarle automaticamente
  - ⇒ Vedremo fra poco uno dei metodi più utilizzati



212

## 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»:  
Piramide di livelli di dettaglio per tessiture:
  - ⇒ 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 1x1 (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



213

### Texture: multirisoluzione




Piramide di livelli di dettaglio  
detta MIP-map



214

### 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 delle schede video) sono progettate per renderizzare mesh triangolari con tessitura associata
  - ⇒ Livello di MIP-map compresi!
  - ⇒ I limiti e le caratteristiche tipiche di una tessitura (num di canali  $\leq 4$ , risoluzione massima, bit depth, risoluzione come potenza di 2) sono imposti da questo



215

## 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

216

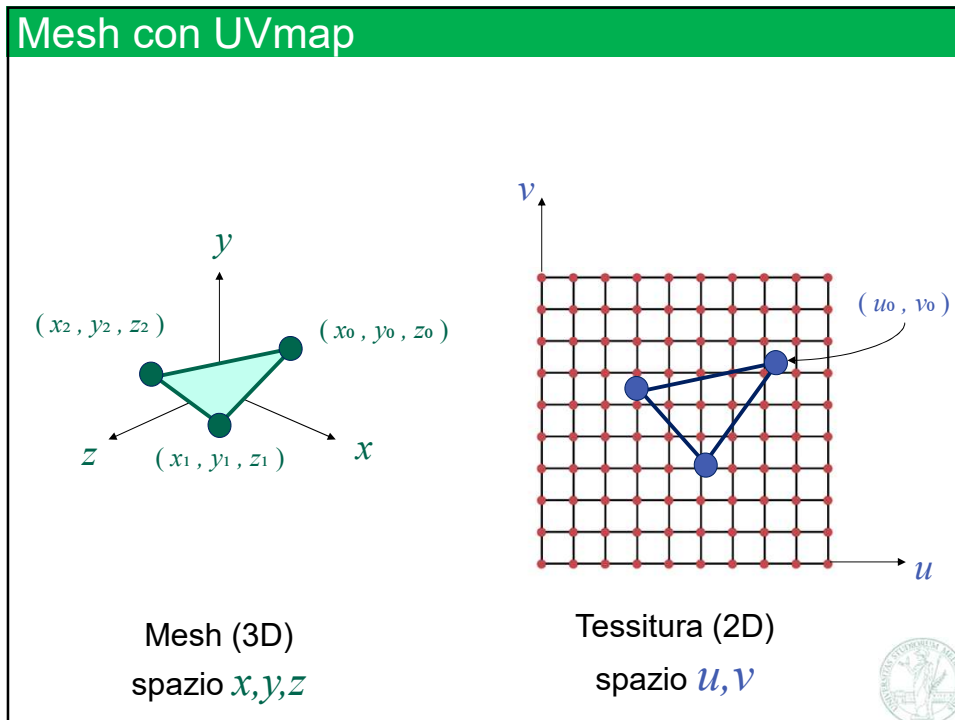
## Mesh con UVmap

- ✓ **Le posizioni UV sono memorizzate come attributo per vertice**
  - ⇒ Come ogni attributo, lo si considera interpolato dentro alle facce della mesh (attraverso le coordinate baricentriche)

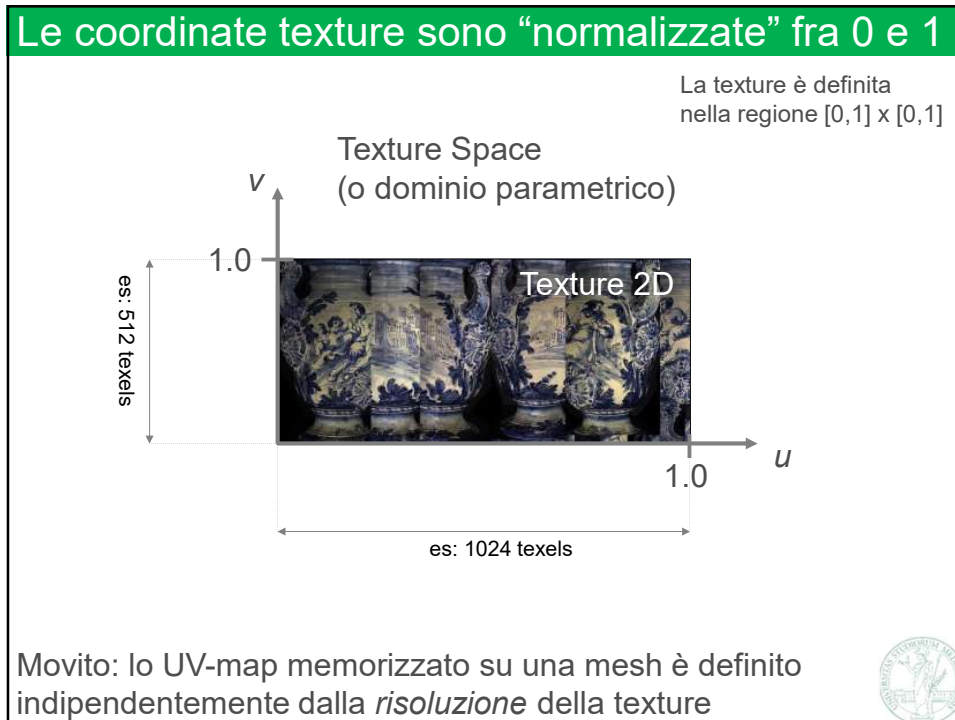
	Geometria	Attributi
$V_0 \rightarrow$	$x_0, y_0, z_0$	$u_0, v_0$
$V_1 \rightarrow$	$x_1, y_1, z_1$	$u_1, v_1$
$V_2 \rightarrow$	$x_2, y_2, z_2$	$u_2, v_2$
$V_3 \rightarrow$	$x_3, y_3, z_3$	$u_3, v_3$
$V_4 \rightarrow$	$x_4, y_4, z_4$	$u_4, v_4$
$V_5 \rightarrow$	$x_5, y_5, z_5$	$u_5, v_5$

- ✓ **Conseguenze:**
  - ⇒ Ogni triangolo della mesh corrisponde a due triangoli: il triangolo  $T_3$  nello spazio in 3D «x,y,z», il triangolo  $T_2$  nello spazio 2D «u,v», dentro al rettangolo della texture
  - ⇒ Ogni punto dentro al triangolo in  $T_3$  è mappato nel punto con le stesse coordinate baricentriche nel triangolo  $T_2$  (come tutti gli altri attributi per vertice, «u,v» è interpolato nelle facce)
  - ⇒ L'intera mesh è embedded sul «dominio parametrico» 2D

217

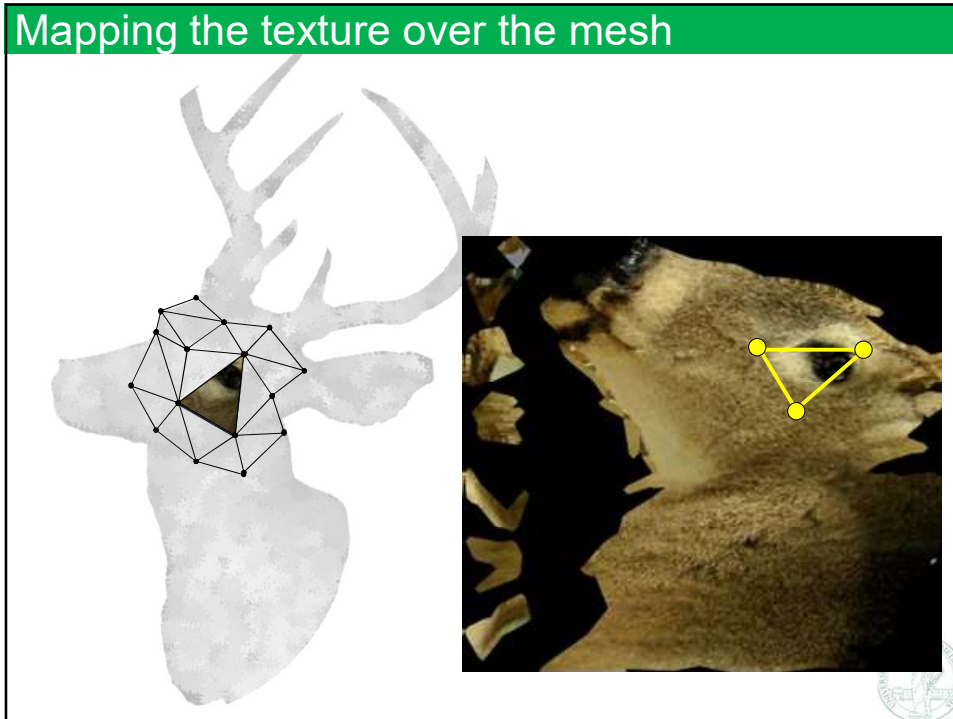


218



219



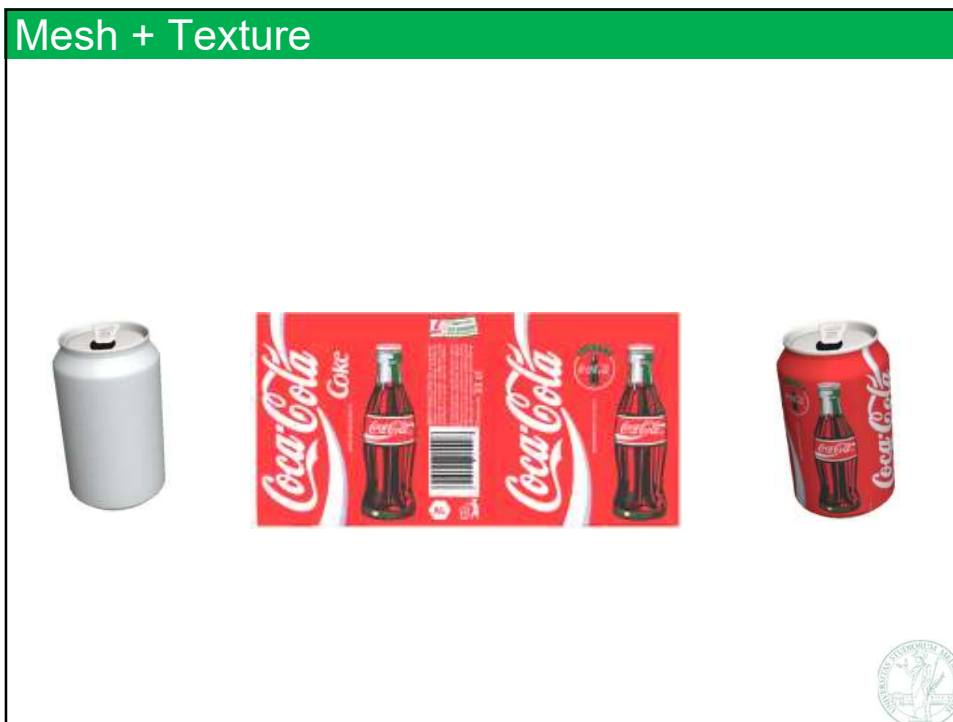


220

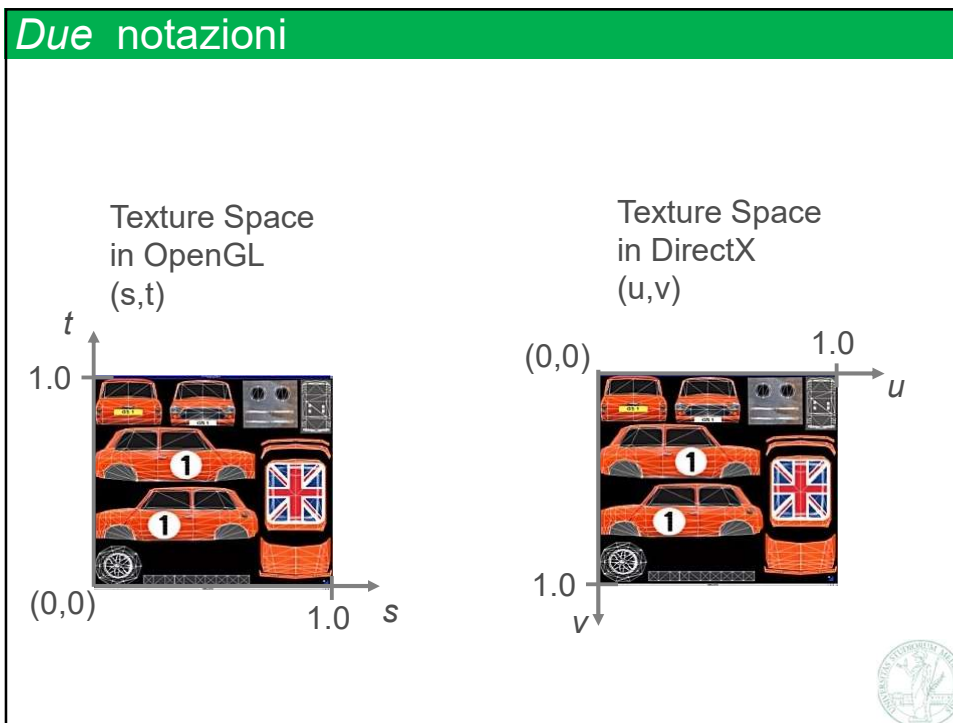


221

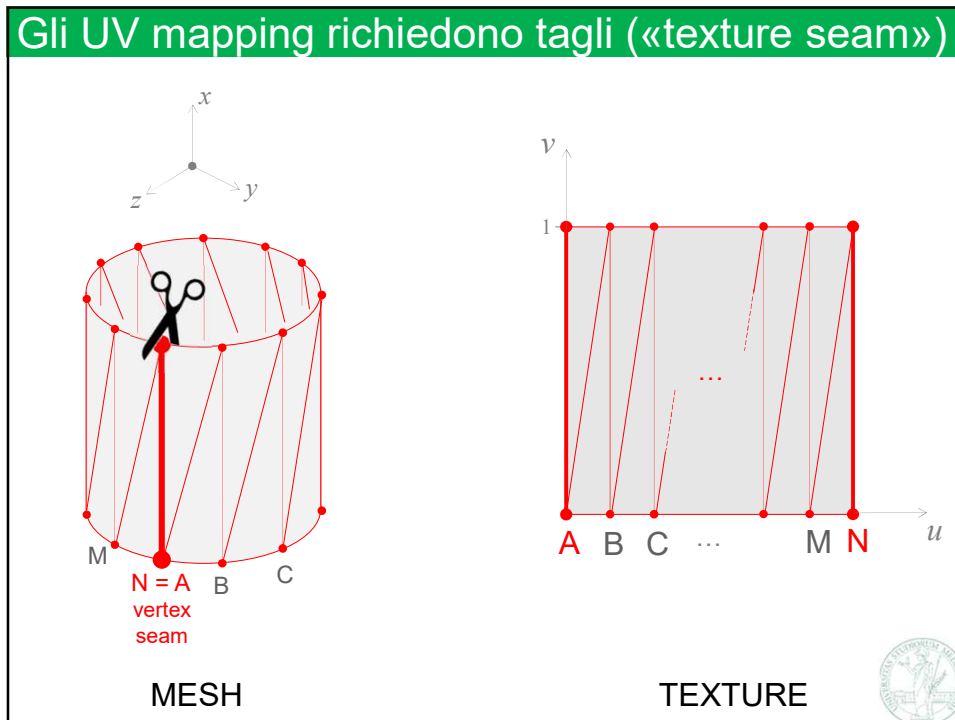




222



223



225

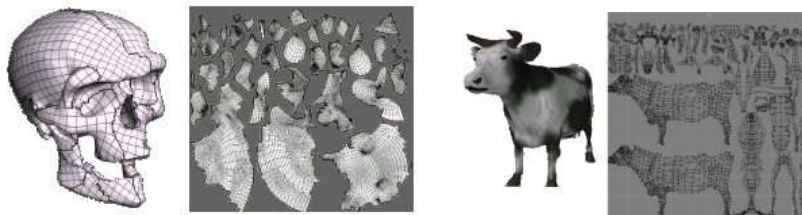
### Costruire un UV-map per una mesh

- ✓ **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:**
  - ⇒ Iniettività: 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

226

## 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 automaticamente sono spesso di scarsa qualità



227

## 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, un UV-map richiede di introdurre dei tagli
  - ⇒ detti «texture seams» o cuts
  - ⇒ questi tagli consentono di aprire la mesh per stenderla (unwrap) su un piano

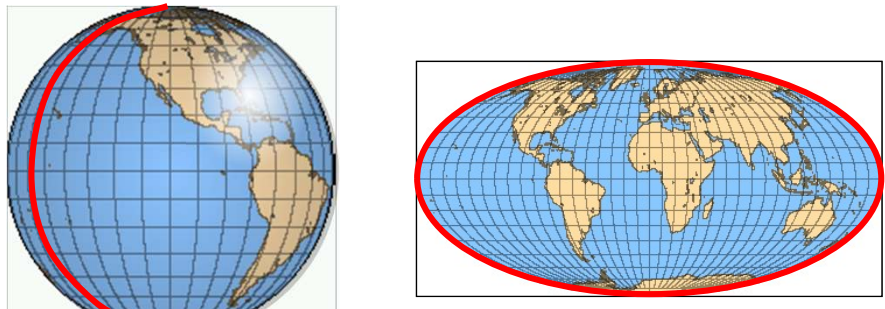


228




### Tagli (cuts) o texture "seams"

✓ Necessari se la mesh è chiusa



two-manifold chiuso  
in 3D

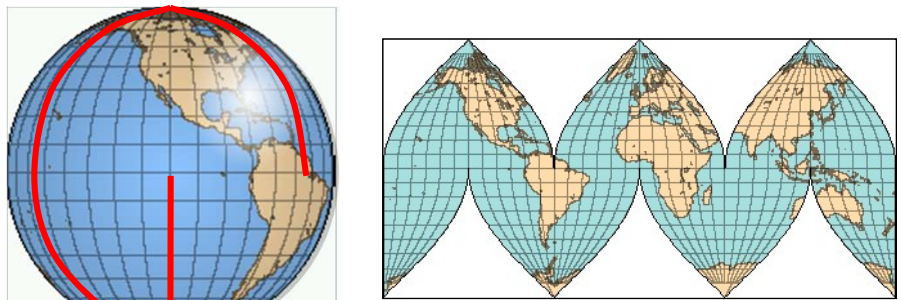
spazio parametrico 2D



231


### Tagli (cuts) o texture "seams"

✓ Più tagli  $\Rightarrow$  meno distorsione (di solito)



two-manifold chiuso  
in 3D

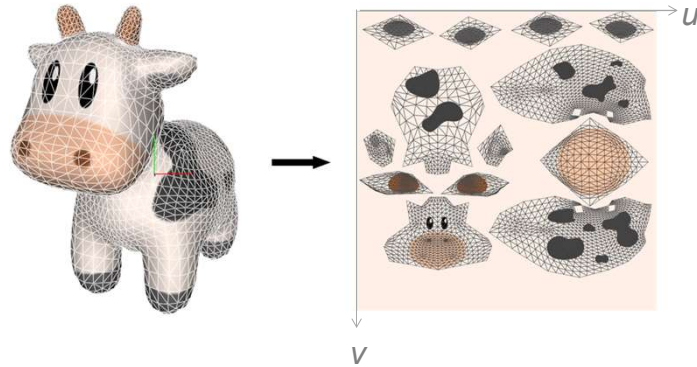
spazio parametrico 2D



232

## Texture Atlas

- Approccio in cui mesh viene divisa in zone (patches), ciascuna parametrizzata in un "isola" di tessitura.
- Lo UV-map di questo tipo si dice «texture atlas» (analogia con un atlante di mappe)



233

## Esempio di mapping generato automaticamente



234





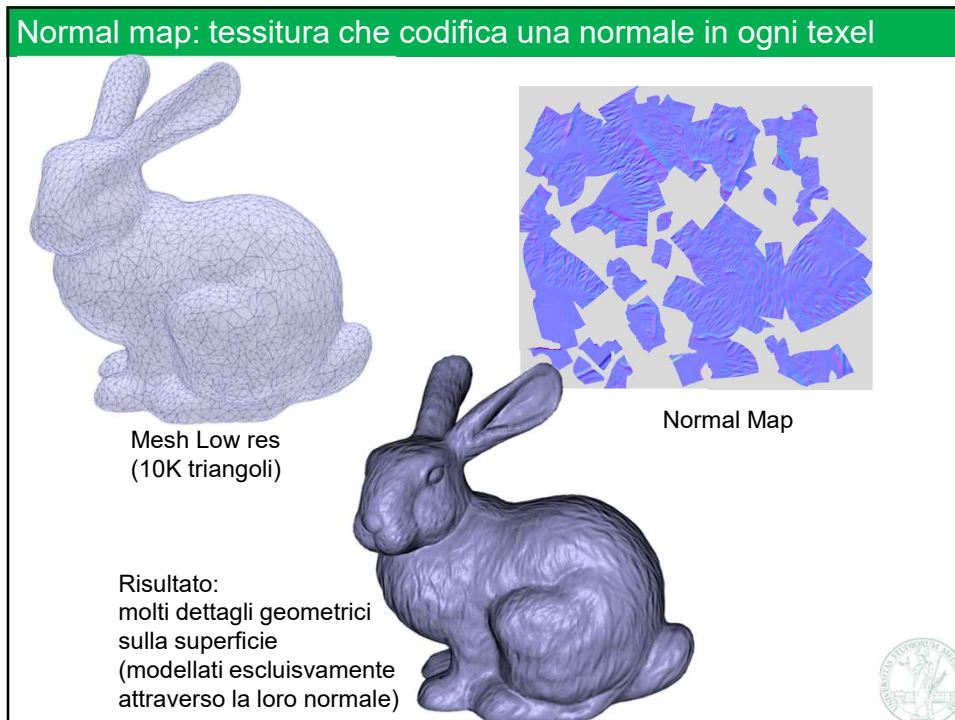
235

### Una normale per Texel: le Normal maps

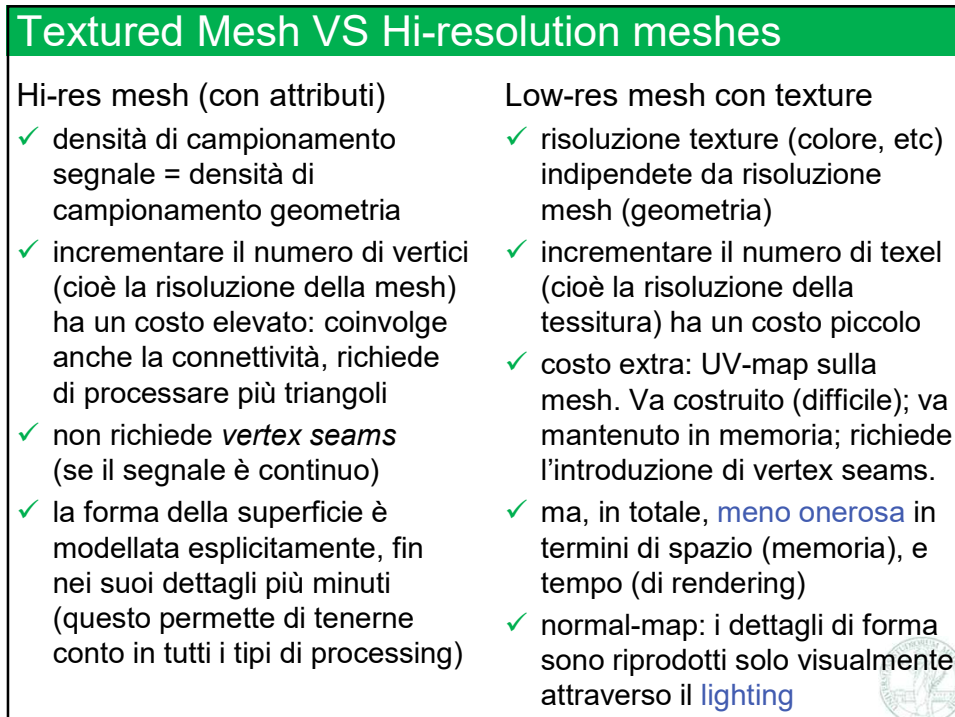
- ✓ Una **normal-map** è una texture che campiona le normali della superficie
  - ⇒ in questo modo, contribuisce a specificare la forma geometrica dell'oggetto
  - ⇒ (introducendo però una approssimazione)
  - ⇒ tipicamente, la normal map riproduce dettagli geometrici minuti, ad alta frequenza (come 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?



236




237



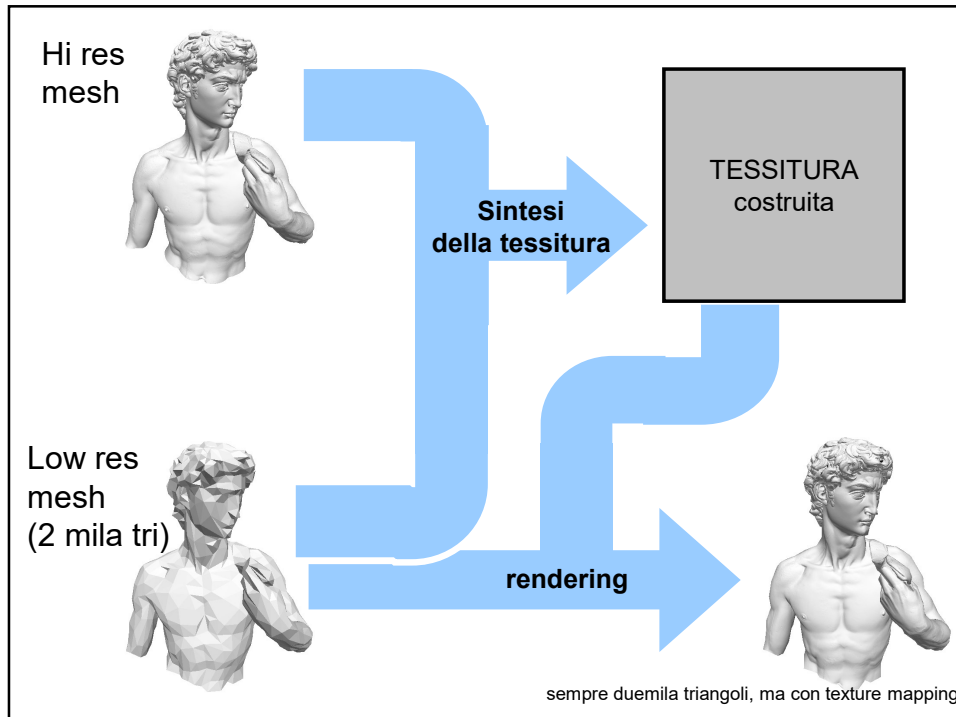
238

**Texture «baking»** (traduzione letterale: cotta al forno)

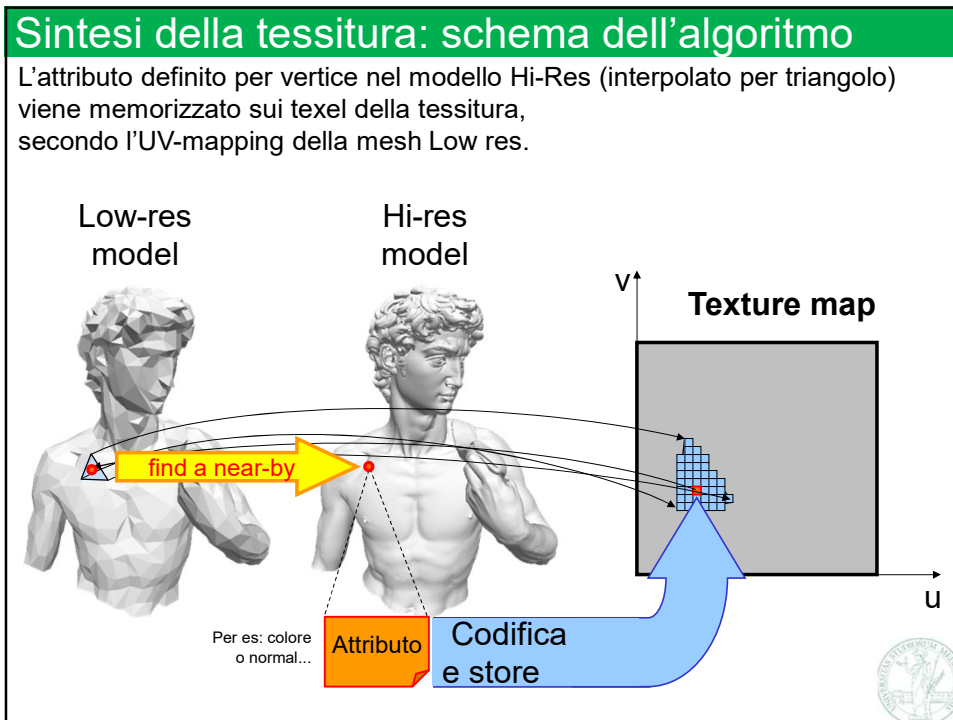
- ✓ Sintesi automatica di una tessitura, per riprodurre un attributo definito su una mesh hi-res
- ✓ Input:
  - ⇒ Mesh Hi-res M0, con attributi (per vertice, per faccia...) che rappresentano un segnale che ci interessa riprodurre nella tessitura (colore, normale, materiale...)
  - ⇒ Mesh Low-res M1 che approssima la stessa forma, dotata di UV-map (cioè «parametrizzata») ma non di tessitura
- ✓ Output
  - ⇒ 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
  - ⇒ Supportato per es da tutte le suite di modellazione 3D



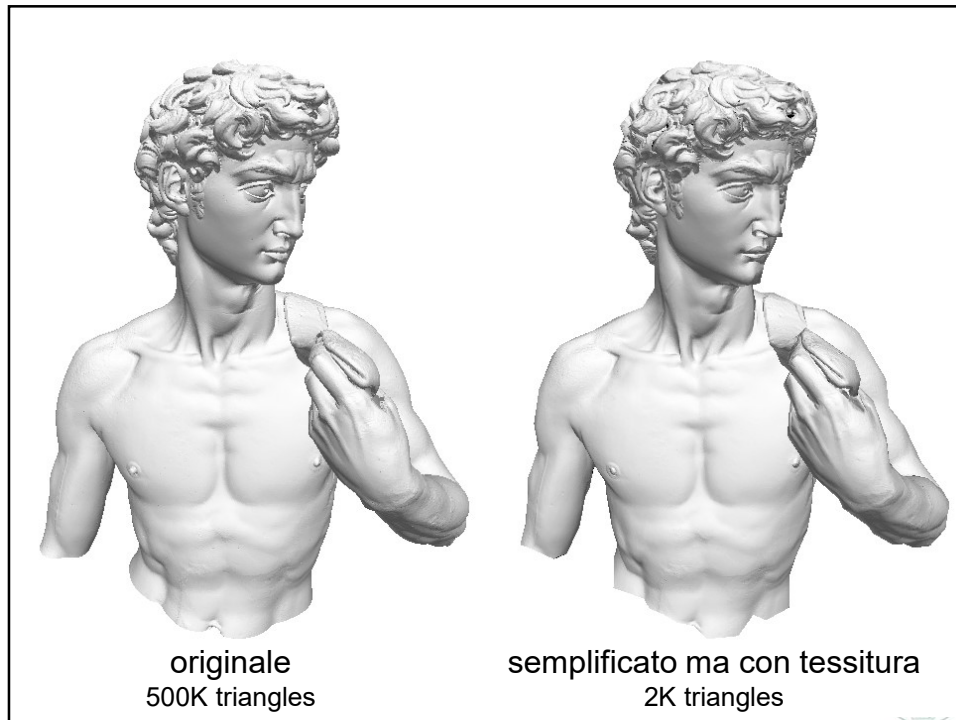
239



240



241



242