


Marco Tarini - Computer Graphics 2025/2026  
Università degli Studi di Milano

Mesh Poligonali:  
Textures & UV mapping 2/2




266

Texture: multirisoluzione (MIP-map)



Piramide di livelli di dettaglio  
detta MIP-map



268

## Tessiture e livelli di MIP-map

- ✓ In quanto immagine raster, una tessitura si presta a semplici strutture multi-risoluzione:
  - ⇒ Basta ridurre la risoluzione dell'immagine per ottenere un «livello» minore, di una piramide di Livello Di Dettaglio (LOD-pyramid)
  - ⇒ Come si riduce la risoluzioni dell'immagine? Vedere corso di image processing (hint: interpolando fra i pixel / i texel)
- ✓ Struttura chiamata (per le texture) «**MIP-map**» :  
E una 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 1x1 (per tessiture quadrate)
- ✓ Nota la convenzione di numerazione:
  - ⇒ (la stessa adottata per tutte le altre piramidi di LOD)
  - ⇒ Il livello 0 (MIP-map-0) è il livello a risoluzione *massima*, cioè originale
  - ⇒ Al crescere del numero di livello, decresce la risoluzione



269

## Tessiture e livelli di MIP-map – perché è utile

- ✓ A differenza delle immagini comuni, le tessiture verranno visualizzate a schermo a scale imprevedibili
  - ⇒ In funzione dall'UV-map della mesh, dalla risoluzione dello schermo, e anche dalla **distanza virtuale** a cui è visualizzata la mesh, etc
  - ⇒ Ad esempio, è possibile che, durante il rendering di una textured mesh, la stessa tessitura di 1024x1024 texel copra un'area a schermo di soli 20x20 pixel in un momento, ma di 1000x1000 pixel in un altro rendering
- ✓ La piramide di MIP-map consente di pre-computare l'aspetto corretto dell'immagine a scale / distanze diverse
  - ⇒ Ad esempio, in una tessitura di colori RGB che rappresenta una scacchiera bianca e nera, i livelli di MIP-map alti mostreranno correttamente che quel pattern, «visto da sufficientemente lontano», si presenta grigio uniforme, e non come un'alternanza di bianco e nero
- ✓ Osservazione:
  - ⇒ A differenza delle LOD pyramid viste per le mesh, i livelli di MIP-map non sono quindi un'ottimizzazione dell'efficienza, ma della qualità visiva
  - ⇒ Sono una forma di *pre-filtraggio delle tessiture*



270

## Tessiture e livelli di MIP-map – costo in memoria

### ✓ Quanti livelli abbiamo?

In tutto abbiamo sono  $1 + \log_2(\text{RISOLUZIONE})$  livelli.

Per es, per una tessitura  $1024 \times 1024$ , avremo i livelli

- ⇒ 0 ( $1024 \times 1024$ ),
- ⇒ 1 ( $512 \times 512$ ),
- ⇒ ...2,3,4,5,6,7,8,9...
- ⇒ 10 ( $1 \times 1$ )
- ⇒ in tutto, 11 livelli, perché  $2^{10} = 1024$

### ✓ Quanto occupa ogni livello?

Ogni livello occupa  $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$  del precedente

- ⇒ sia l'altezza che la larghezza sono la metà,  
quindi il numero di texel è metà x metà = un quarto

### ✓ Quanto occupa la piramide in tutto? Poco!

lo spazio totale occupato da tutti i livelli oltre lo 0 (da 1 a N) è solo  $\frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \frac{1}{256} + \dots \rightarrow \frac{1}{3}$

di quanto occupa la tessitura a piena risoluzione (livello 0)



271

## 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?

✓ Le differenze sono ne...

- ⇒ l'**uso atteso**: una texture è pensata esplicitamente per essere rimappata sulla superficie una mesh (attraverso l'UV-map codificato nella mesh)
- ⇒ il **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 scalari Rosso, Verde, Blu (R,G,B)

✓ Esistono formati file di interscambio specifici per texture (come «.dds»), diversi da quelli per immagini generiche (come «.png» o «.jpeg»), con...

- ⇒ **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 alle GPU, 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:  $1024 \times 1024$ ,  $512 \times 512$ ,  $256 \times 256$ , ... ,  $2 \times 2$ ,  $1 \times 1$ )


✓ Un formato per immagine generica (come «.png») può essere comunque usato come formato di interscambio per texture



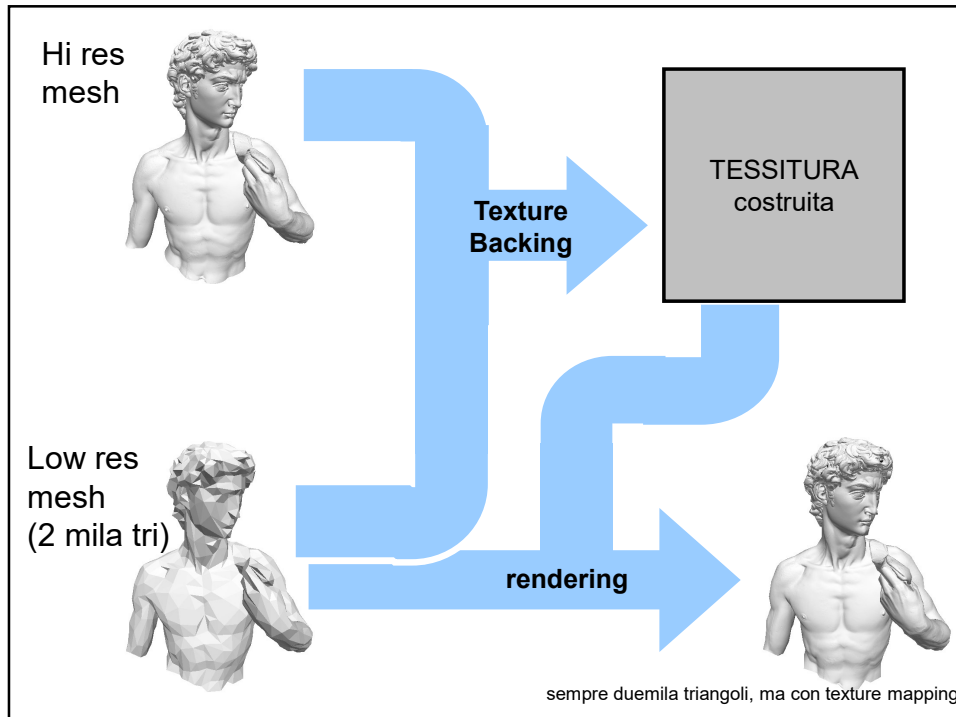
272

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

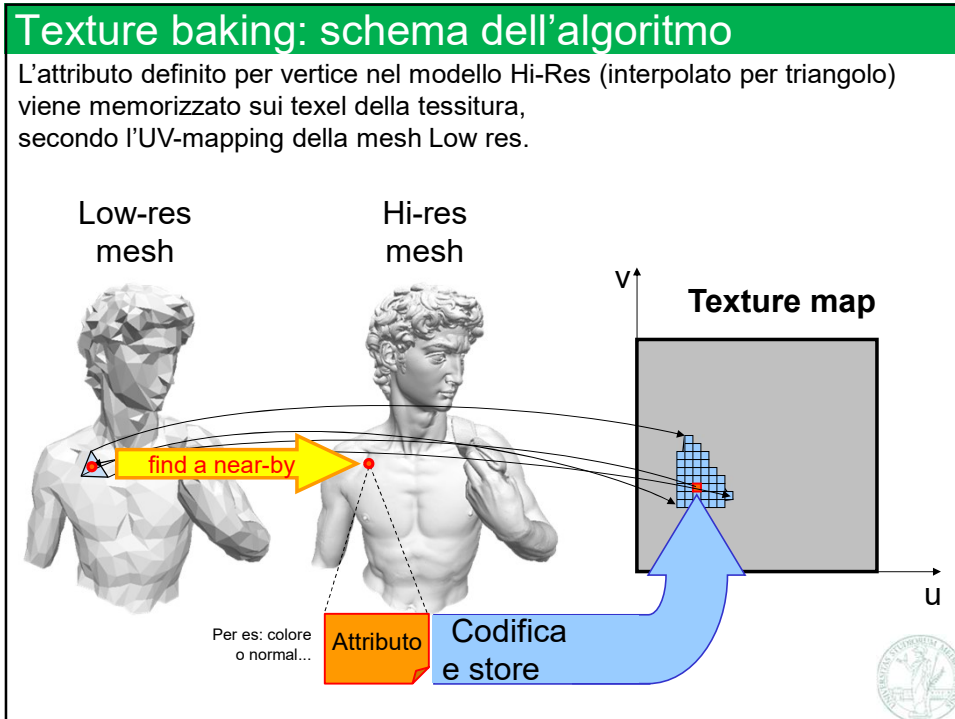
- ✓ Sintesi automatica di una tessitura, per riprodurre un attributo definito su vertici (o facce) di una mesh hi-res
- ✓ Input:
  - ⇒ Mesh Hi-res  $M_H$ , con attributi (per vertice, per faccia...) che rappresentano un segnale che ci interessa riprodurre nella tessitura (colore, normale, materiale...)
  - ⇒ Mesh Low-res  $M_L$  che approssima la stessa forma, dotata di UV-map (cioè «parametrizzata»), ma non ancora di tessitura
- ✓ Output
  - ⇒ una tessitura sintetizzata (baked texture) per  $M_L$  che riproduce su di essa l'attributo definito su  $M_H$
- ✓ Un passaggio comune nel pipeline di creazione dei modelli 3D
  - ⇒ Supportato per es da tutte le suite di modellazione 3D



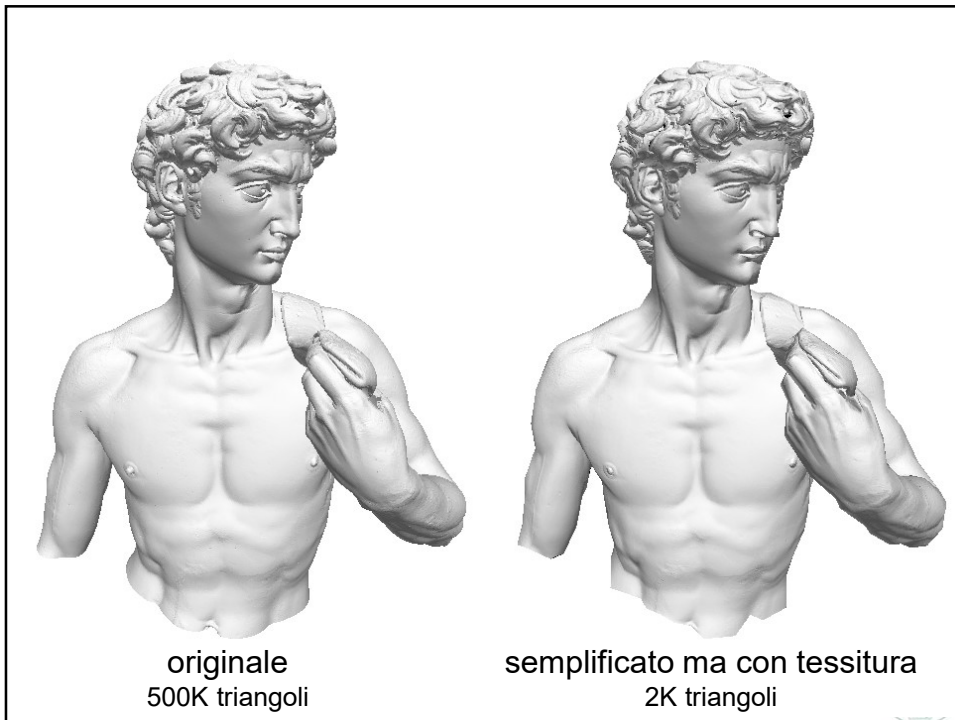
273



274



275



276