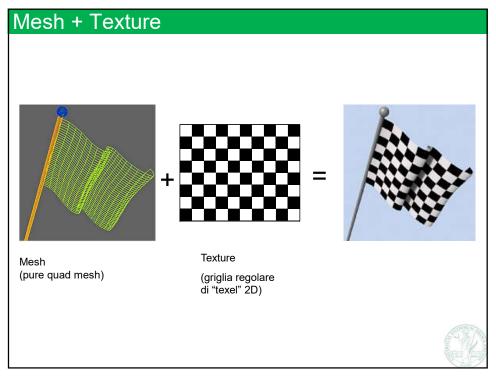


#### Textures

- 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 cm² (sufficiente per rappresentare una statuetta in dettaglio)
  - ⇒ colore: vorremmo 100 campioni per cm² (necessario per riprodurre il dettaglio pittorico sulla statuetta)
- ✓ 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 (su una mesh di triangoli - 1 su mesh di quad)
- ✓ In 2D, le immagini "raster" (griglie 2D regolari di pixel) sono un ottimo modo di campionare segnali in modo molto denso
  - ⇒ es: un'immagine a 100 DPI = 40 dpmc = 1600 campioni («pixel») per cm²
  - ⇒ es: immagine risoluzione 1000x1000 = 1 Milone di pixel
- ✓ Come ottenere un risultato simile per le superfici 3D?

138





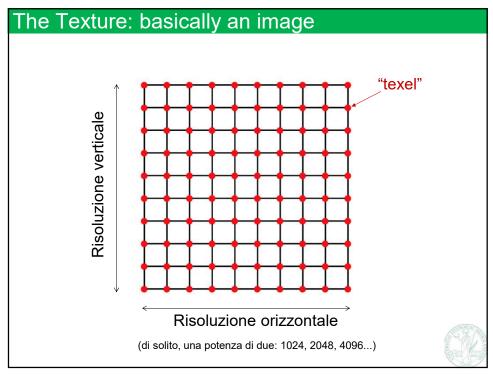
140

# **Texture Mapping**

- ✓ Idea intuitiva: incolliamo un'immagine raster (detta texture, o tessitura) alla superficie di una mesh
  - ⇒ immagine (raster) = grigliato regolare di pixel
  - ⇒ texture = grigliato regolare di texel (texel = texture element)
- ✓ Ogni punto sulla superficie 3D S ← rappresentata dalla mesh viene associato ad un punto sulla rettangolo 2D T
  - $\Rightarrow$  convenzione: le coordinate 2D su T  $\leftarrow$  anche dette coordinate sono indicate da (u,v) a volte da (s,t) o parametriche 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 industiale
    oppure la sua parametrizzazione
    in contesto accademico
- ✓ Domande a cui dobbiamo rispondere:
  - ⇒ come è strutturata una texture
  - ⇒ come memorizzare lo UV-map sulla mesh?
  - ⇒ come ottenerlo per una data superficie? («how do we parametrize» or «how do we uv-map» a mesh)



141



142

#### 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 sostanzialmente un'immagine raster
  - ⇒ Un array 2D di «texel»
- ✓ Come tale, è caratterizzata da:
  - ⇒ Risoluzione orizzontale Rx (di solito, potenza di due  $\leq 2^{13} = 8192$ )
  - ⇒ Risoluzione verticale Ry (idem)
  - ⇒ Numero di canali per texel *Ch* (tipicamente, 1,2 o 4)
  - ⇒ Numero di bit per canale B (8 (spesso), 16, 24 ...)
    - di texel «bit-deptil»

numero

- ⇒ Spazio totale occupato in memoria (espressa in bit) = Rx · Ry · Ch · B
- ✓ In quanto immagine, può essere, per es, «dipinta» da un artista, catturata da foto, photoshopped, etc
  - ⇒ Esistono anche molte tecniche per sintentizzarle automaticamente
  - ⇒ Vedremo uno dei modi più comuni



#### Tessiture e MIPmap level

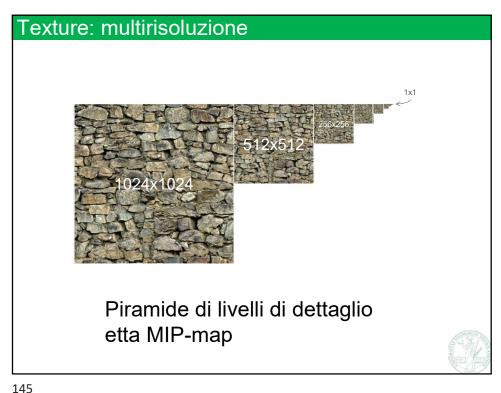
- ✓ In quanto immagine raster, una tessitura si presta a semplici strutture multirisoluzione:
  - ⇒ Basta ridurre la risoluzione dell'immagine per ottenere un «livello» diverso di una piramide di livello di dettaglio
  - ⇒ Come si riduce la risoluzioni dell'immagine? Vedere corso di image processing (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 con metà della risoluzione X e Y
- ⇒ Livello 2: riduzione del livello 1
- ⇒ ..
- ⇒ Livello N: tessitura 1x1 (per tessiture quadrate)
- ⇒ Nota: in tutto sono log₂(RES) livelli
- ⇒ Nota: in tutto, lo spazio tutale occupato da tutti i livelli oltre lo 0 è solo 1/4 + 1/16 +1/64 + 1/256 + ... < 1/3 quello del livello zero



144

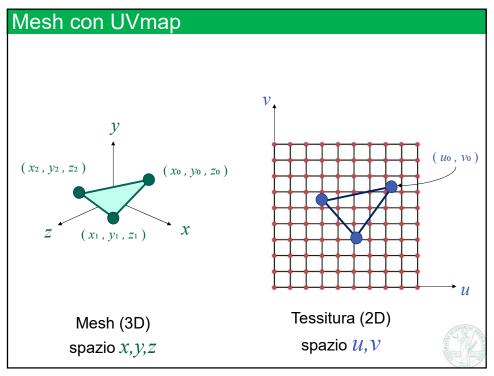


# Tessiture e supporto Hardware

- ✓ Le tessiture sono estremamente utilizzate nella maggior parte dei contesti applicativi in cui si usino mesh poligonali
  - ⇒ Come: games, VR, movies, commercio elettronico, applicazioni architetturali, sci vis, beni culturali... (ma spesso non: CAD, applicaz medicali)
- ✓ Parte della loro diffusione è dovuta al supporto HW:
  - $\Rightarrow$  le GPU (il processore delle schede video) sono progettate per renderizzare mesh triangolari con tessitura associata
  - ⇒ Livello di MIPmap 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 da questo

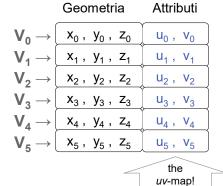


146



#### Mesh con UVmap

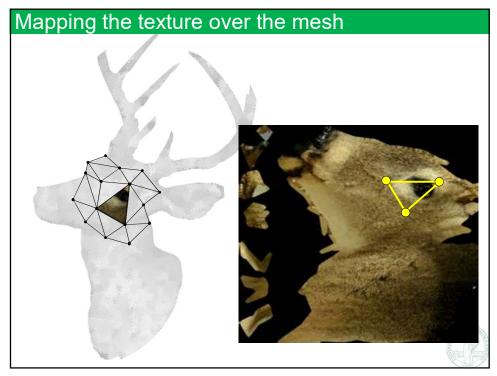
- Le posizioni UV sono memorizzate come attributo per vertice
  - ⇒ Come ogni attributo, si considera interpolato dentro alle facce della mesh (con le coordinate baricentriche)

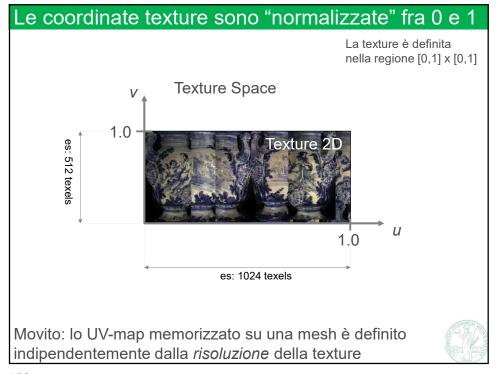


- Conseguenze:
  - ⇒ Ogni triangolo della mesh corrisponde a due triangoli:
     il triangolo T₃ nello spazio in 3D «x,y,z»,
     il triangolo T₂ nello spazio 2D «u,v», dentro al rettangolo della texture
  - ⇒ Ogni punto dentro al triangolo in T<sub>3</sub> è mappato nel punto con le stesse coordinate baricentriche nel triangolo T<sub>2</sub>
  - ⇒ L'intera mesh è mappata sul rettangolo di tessitura

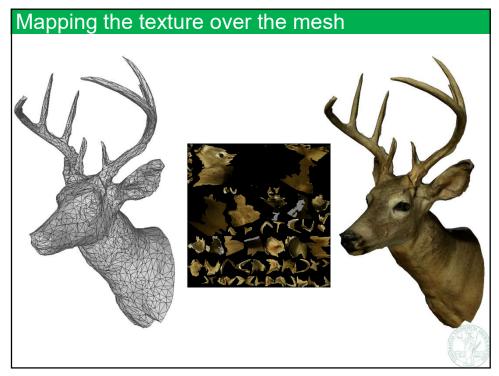


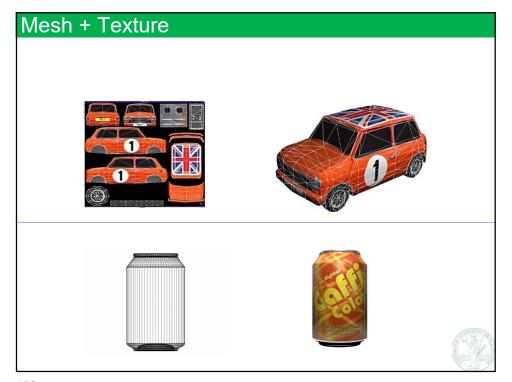
148





150





153



# Costruire un UV-map di una mesh

- ✓ UV-mapping di una mesh, o mesh parametrization assegnamento di coordinate UV per ogni suo vertice
  - ⇒ A volte, (raro) la mesh viene creata congiuntamente al suo UV map
  - ⇒ Più spesso, la mesh viene create (es. acquisita, o modellata) ed è sprovvista di un UV-map per una mesh che ne sia
- ✓ 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 (no triangoli sovrapposti in spazio UV)
  - ⇒ Bassa distorsione: ogni triangolo di mesh T₃ deve essere mappato in un triangolo di tessitura T₂ di forma simile e di area proporzionale
  - ⇒ Buon coverage: le parti di tessitura non coperte da nessun triangolo rappresentano uno spreco di memoria e vanno minimizzate
  - ⇒ Limitato numero di texture seams (see later)

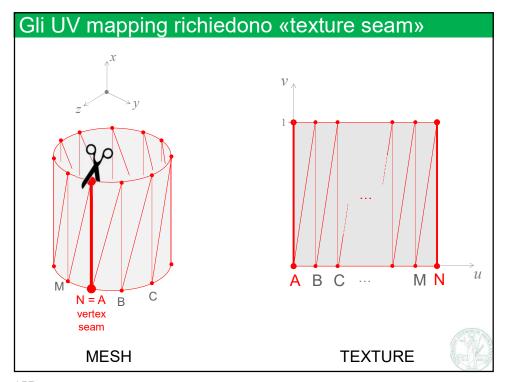
155

## Texture seams (o cut)

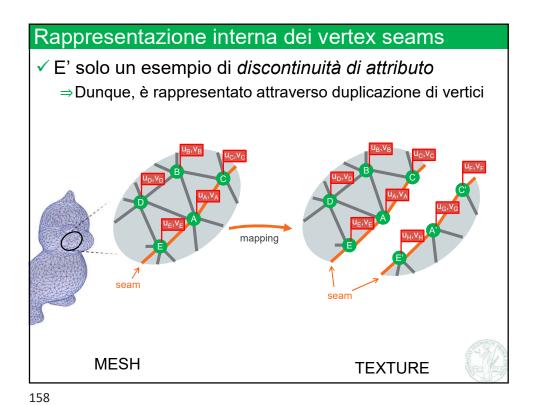
- ✓ Nozione intuitiva:
  - ⇒il task di mesh parametrization ricorda quello di sbucciare una mela e disporre la buccia dentro un rettangolo
  - ⇒ Simile: come costruire un planisfero, cioè la mappa (2D) della superficie del pianeta (sfera in 3D) (problema storicamente molto studiato: es. vedi: http://vcg.isti.cnr.it/~tarini/spinnableworldmaps/
  - ⇒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



156

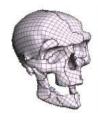


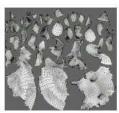
157



## 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





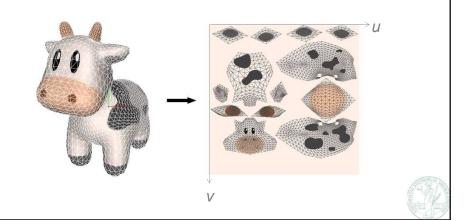




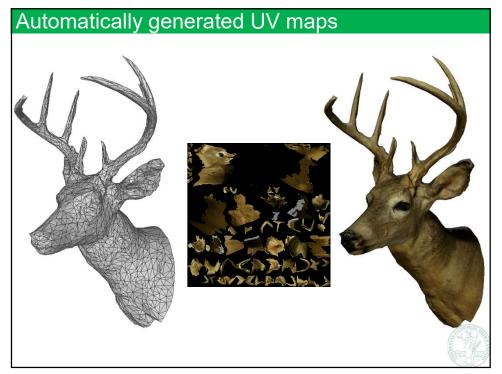
159

# Texture Atlas

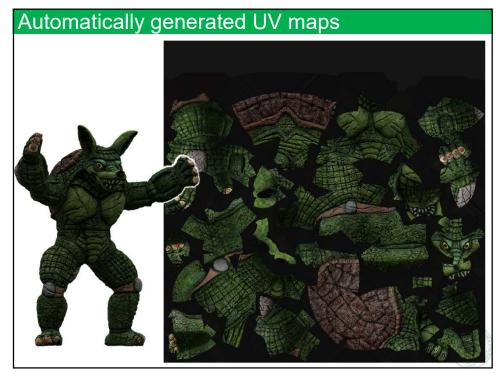
- Spesso la mesh viene divisa in zone, ciascuna parametrizzata in un "isola" di tessitura.
- Lo UV-map di questo tipo si dice «texture atlas» (un atlante di mappe)



162



163

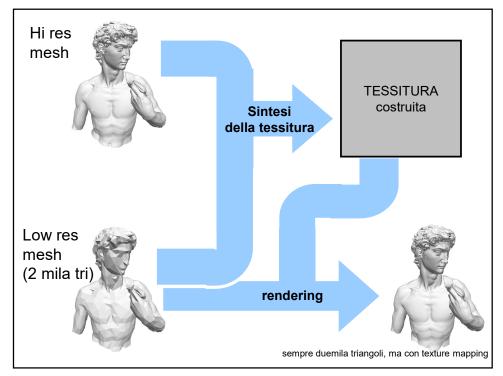


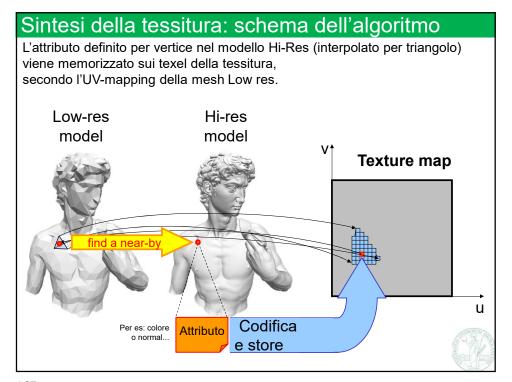
# Campionare una texture da mesh hi-res

- ✓ Input:
  - ⇒ Mesh Hi-res M0, con attributi che ci interessa riprodurre
  - ⇒ Mesh Low-res M1, parametrizzata (con UV map)
- ✓ Output
  - ⇒ sintetizzare una tessitura per M1 che riproduca l'attributo definito su M0
- Un passaggio comune nel pipeline di creazione delle mesh
  - ⇒Supportato per es da tutte le suit di modellazione 3D

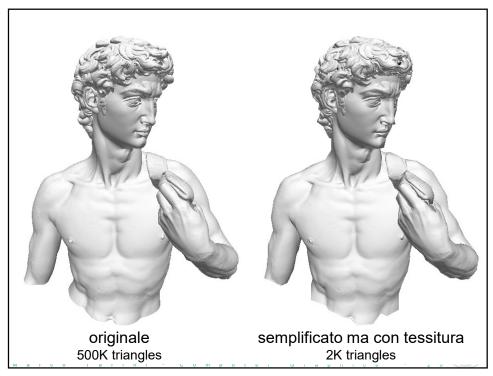


165





167



#### Textured Mesh VS Hi-resolution meshes

Hi-res mesh (con attributi)

- densità di campionamento segnale = densità di campionamento geometria
- aumentare i vertici richiede di incrementare anche la connettività (e quindi di procesare più triangoli durante il rendering)
- ✓ non richiede vertex seams (se il segnale è continuo)
- ✓ normale & posizione per vertex: la forma è catturata per intero, compreso nei suoi dettagli (questo permette di tenerne conto in tutti i tipi di processing)

Low-res mesh con texture

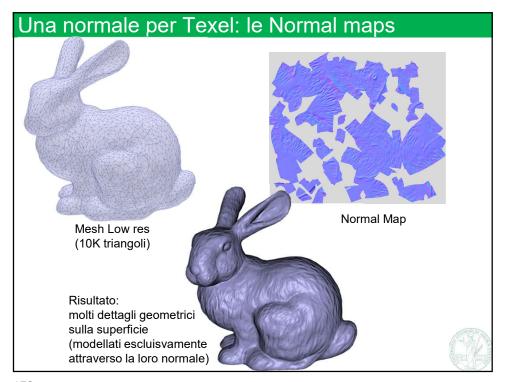
- risoluzione texture (colore, etc) indipendete da risoluzione mesh (geometria)
- ✓ incrementare i campioni di tessitura ha un costo piccolo
- costo extra: UV-map sulla mesh. Va costruito (difficile); va mantenuto in memoria; richiede di introdurrealcuni vertex seams.
- ✓ in totale, meno onerosa in termini di spazio (in memoria), e tempo (rendering)
- ✓ normal-map: i dettagli di forma sono solo riprodotti visualmente attraverso il lighting

169

# 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?

171



172