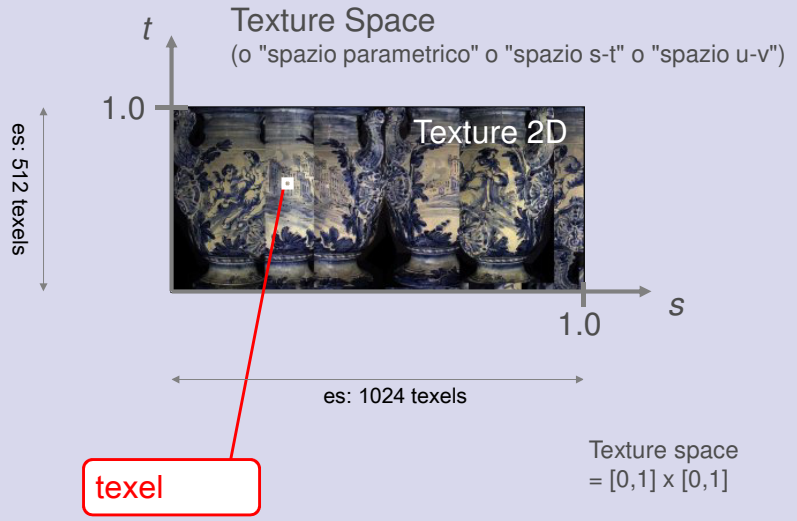
Texture maps assets e Mesh assets

- Non necessariamente 1:1
  - 1:N -- vari texture «sheets» associati ad una mesh
  - N:1 -- più meshes sullo stesso sheet (bene)
- esempio di struttura per :
  - ogni mesh associata a un materiale
  - ogni materiale:
    - 1 sheet di diffuse-map
    - 1 sheet bumpmap (se serve)
    - 1 sheet di alphamap (se serve)
    - 1 vertex + fragment shader
    - vari parametri
      - (es, shininess, …)
  - se parti diverse di mesh associate a tessiture diverse: scomporre oggetto in sottomesh

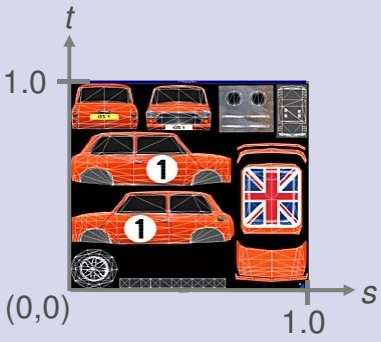
Notazione di spazio texture



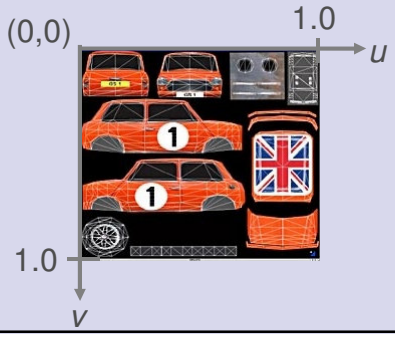
Due notazioni

più diffusa
(in game industry)

s-t
(es OpenGL)

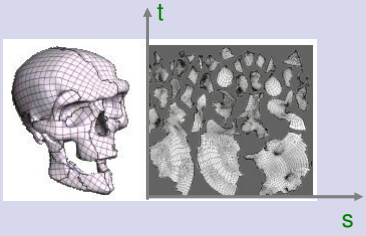


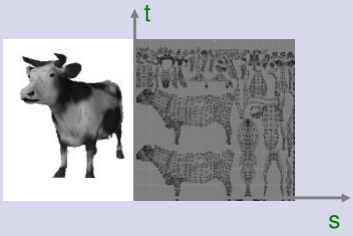
u-v
(es DirectX)



Task di modellazione: "u-v mapping" ("u-v" == "s-t")

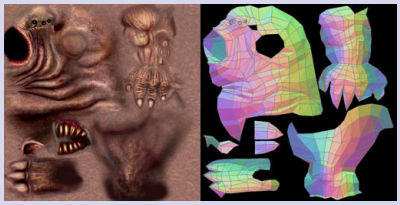
- Assegnare una coppia di coordinate texutres ad ogni vertice della mesh
  - In preprocessing





Task di modellazione: “u-v mapping” (“u-v” == “s-t”)

fatto a mano,
oppure automatizzato



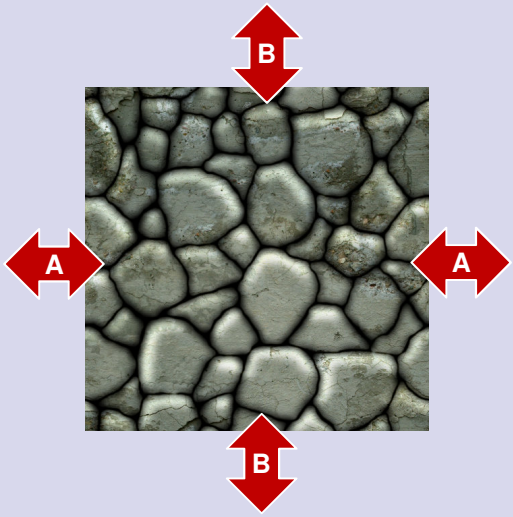
Texture “atlas” (composto di vari “charts”)

Task di modellazione: “u-v mapping” (“u-v” == “s-t”)

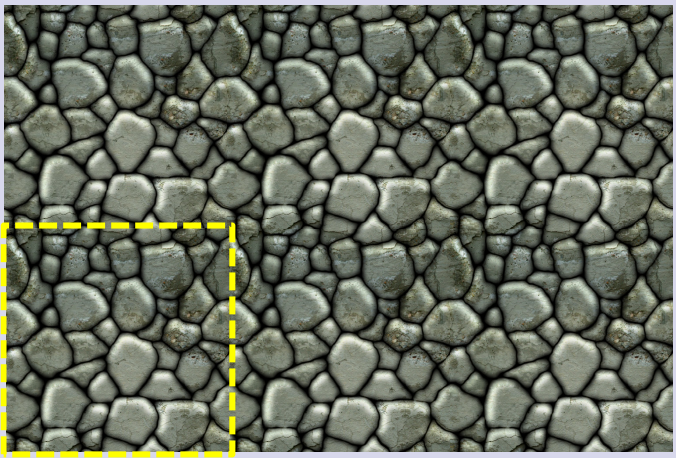
DEMO!

- strategie pratiche:
  - 1. selezionare edge di taglio
    …o…
    1. assegnare facce a charts
    - decidere dove sono i “texture seams”
  - 2. unfolding
    - minimizzare “distorsione”
  - 3. packing dei charts
    - minimizzare spazi vuoti
    - assegnare aree secondo necessita’
      (es, parti importanti → maggiore spazio tessitura)

Tileable Textures

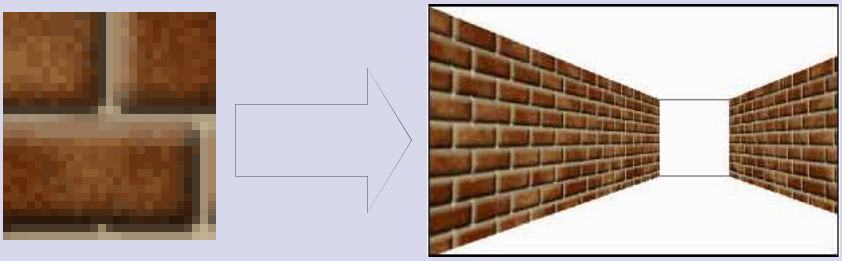


Tileable textures



Tileable textures

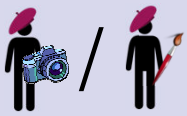
- Tipico utilizzo:



Molto efficiente in spazio!

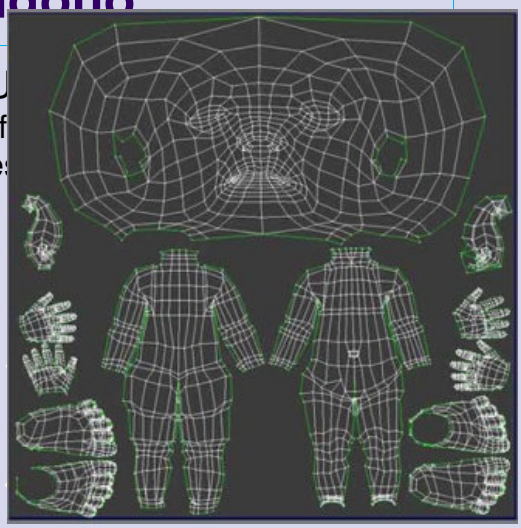
RGB maps: come si ottengono

- Image first, then U
  - e.g. immagine da f
  - e.g. tileable image



2D painter

3D modeller

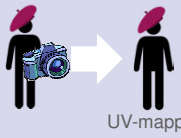


Texture maps come assets

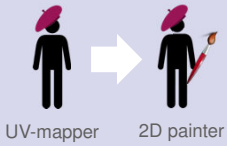
- Vari texture «sheets» associati ad una mesh
  - o anche: più meshes sullo stesso sheet (bene)
- tipica struttura dati:
  - mesh divisa in sottomesh
  - ogni sottomesh associata a un materiale
  - ogni materiale:
    - 1 sheet di diffuse-map
    - 1 sheet bumpmap (se serve)
    - 1 sheet di alphamap (se serve)
    - 1 vertex + fragment shader
    - vari parametri
      - (es, shininess, …)

RGB maps: come si ottengono

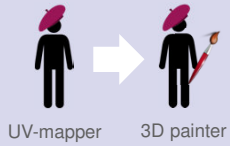
- Image *first, then* UV-mapping
  - e.g. immagine da fotografie
  - e.g. tileable images



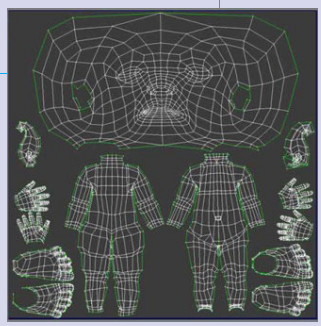
- UV-mapping *first, then* paint 2D
  - paint with 2D app (e.g. photoshop)



- UV-mapping *first, then* paint 3D
  - paint within 3D modelling software,
  - *or:* 1. export 2D rendering,
    2. paint over with e.g. photoshop,
    3. reimport images
    4. goto 1



RGB maps: come si ottengono

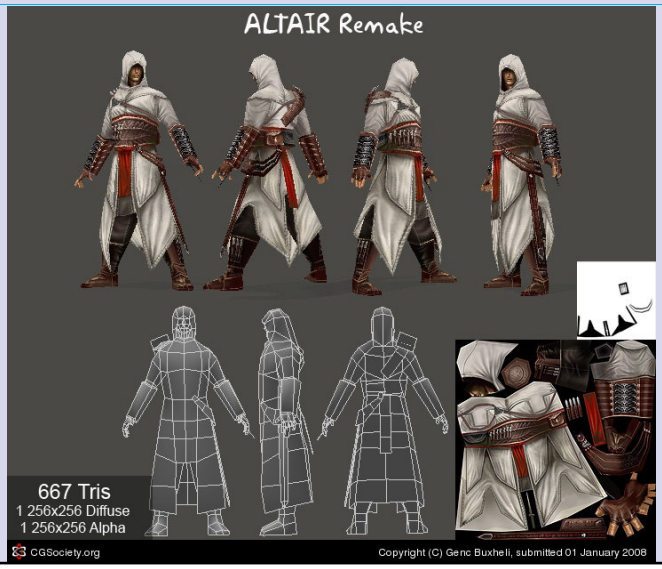


…or:

- *first* Paint 3D
  - on hi-res model,
  - “paint” on vertex attributes
  - e.g. with Z bursh…
- *then* coarsen
  - build / autobuild final low-poly version
- *then* UV-map
  - the low-poly model
  - must be a 1:1 mapping!
- *then* auto-texture
  - auto build texture

more about this later…

Alpha mapping (texels = lvl trasparenza)

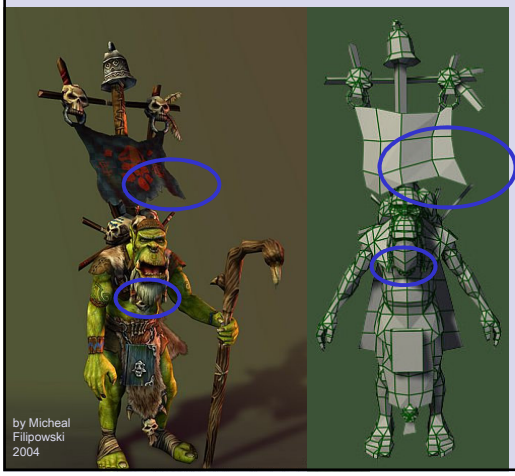


Alpha map

RGB map

Alpha mapping (texels = lvl trasparenza)

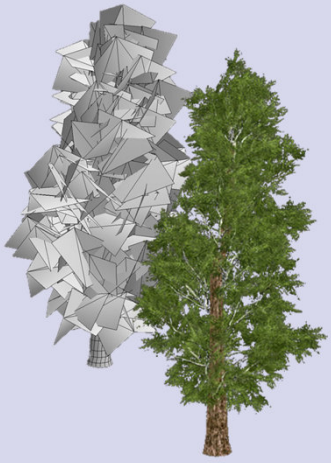
- es: drappi, barba...



tessitura

Texture mapping e Alpha Test

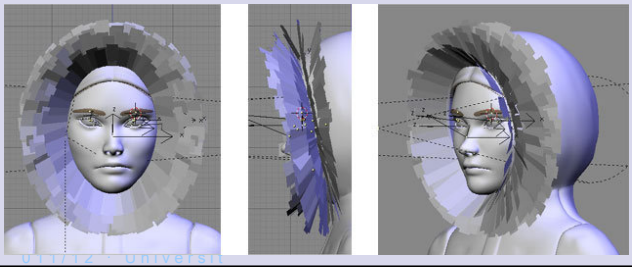
- es: alberi, foliage



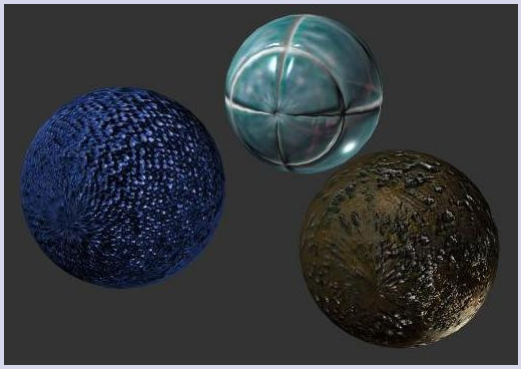
Texture mapping e Alpha Test

- Es: pelo, pellicce

tessitura
(ripetuta)

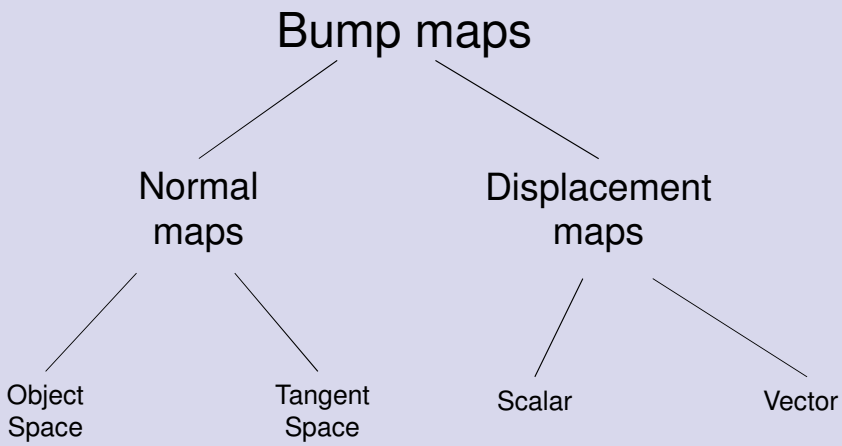


Bump-Mapping (see demo)



stessa geometria (una sfera)
bumpmaps diverse

Bump maps: Categorie



Bump maps: Categorie

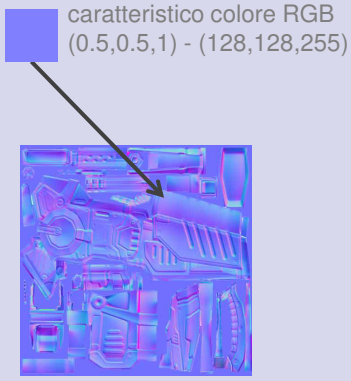
- Bump map:
  - qualunque tessitura che codifica dettagli hi-freq ("meso-struttura") su una sup low-res

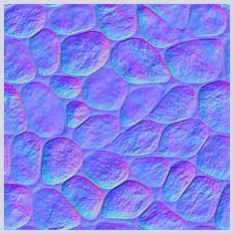
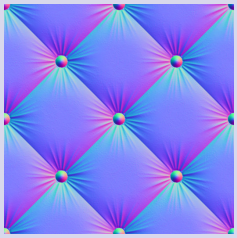
caso più comune, di "bump map"

- Normal Map:
  - Dettagli codificati memorizzando le normali della sup hi-freq
  - Modificano il lighting
  - In quale spazio (in che base vettoriale)?
    - Tangent Space: (spazio TBN)
      Riutilizzabili su più superfici indipendentemente dall'oreintamento
      Richiede direzioni Tangenti-Bitangenti (e normali) def su superifcie
    - Object Space:
      Solo per UV-mapping 1:1
- Displacement Map
  - Dettagli codificati memorizzando le differenze fra low-res e hi-freq
  - Come vettori, oppure come scalari (distanza lungo la normale)
  - Usati per re-tasselation, o per effetto parallasse (*parallax mapping*)

Normal maps

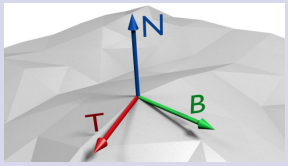
- Memorizzati come immagini
  - RGB => XYZ
    (come?)



(Tangent Space) Normal maps

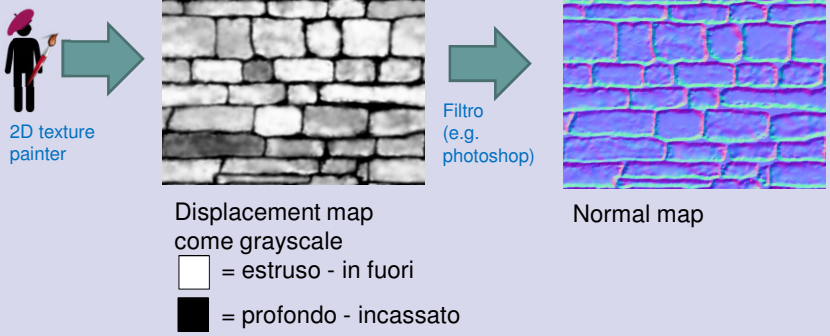
- Espressi in *spazio tangente*
  - Memorizzare direzioni tangenti
    Tangente e Bi-tangente
  - Nuovi attributi x vertice!
  - Computati automaticamente
    a partire da UV mapping (come?)



Normal maps

[DEMO]

- Come si ottengono (1/3):

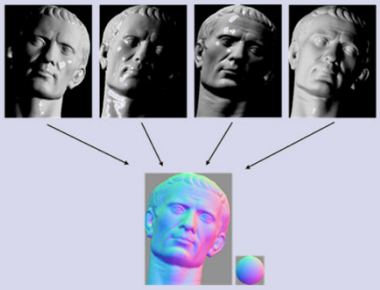


Displacement map come grayscale
= estruso - in fuori
= profondo - incassato

Normal map

Normal maps

- Come si ottengono (2/3): Photometric Stereo



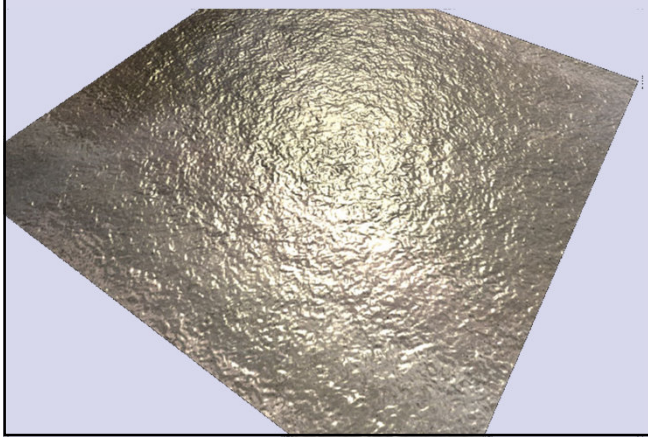
  - da: N immagini (N>=4) reali
    - Stesso punto di vista
    - Illuminazione diversa
      - (e possibilmente controllata e nota)
  - a Normal Map

Normal maps

- Come si ottengono (3/3):
  - Detail recovery
    - da:
      - 1) mesh Hi-Res
      - 2) mesh Low-res + UV mapping (senza ripetizioni)
    - a:
      - Normal map per 2
        (che mimica il dettaglio presente in 1)
  - More about this later…

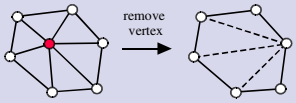
Normal maps

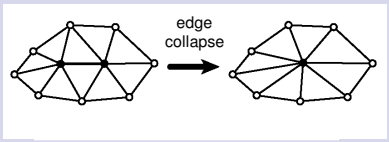
- Come si ottengono (bonus):
  - Proceduralmente

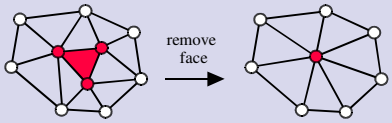


Semplificazione automatica aka. “poly reduction”

- Strategie completamente diverse
  - Approcci iterativi
    - repeat
      - compi l'azione di semplificazione atomica meno costosa (in termini di errore aggiunto)
      - aggiorna costi
    - until (obiettivo raggiunto)
      - es: numero faccie, errore

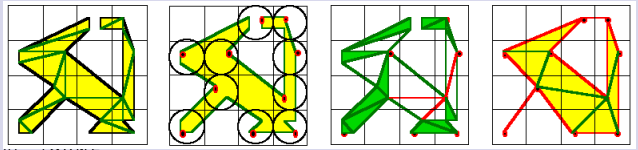






Semplificazione automatica

- Strategie completamente diverse
  - Vertex clustering:
    - dividi i vertici originali in una griglia regolare
    - "collassa" in un solo vertice tutti quelli nella stessa casella
    - togli i triangoli che hanno solo 1 o 2 vertici diversi
  - Approssimazione dipende da dimensione griglia

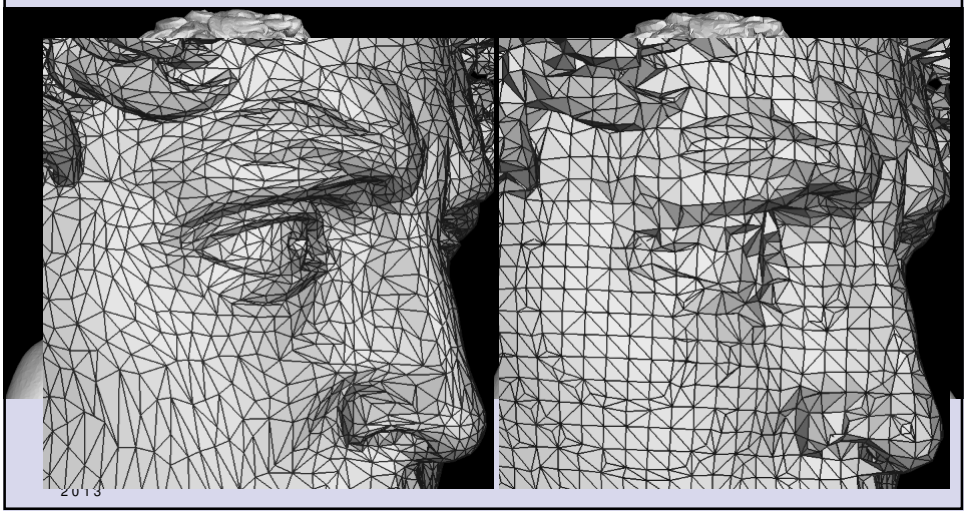


Marco Tari... [GAME -DEV] · Verona · 2013

Semplificazione automatica

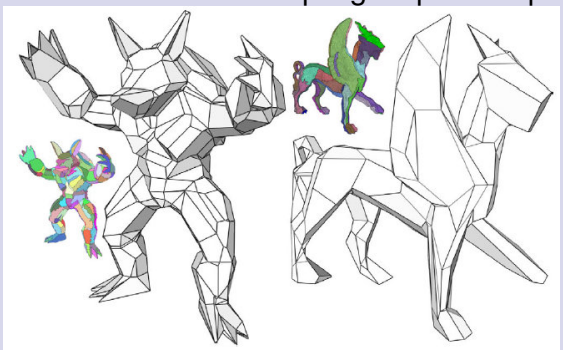
Simplificazione ad edge collapse
(adattva)
215k faces

Semplificazione a clustering
(non adattiva)
235k faces



Semplificazione automatica

- Strategie completamente diverse
  - Fitting di piani
    - sostituire molte facce con poligoni planari quando i loro

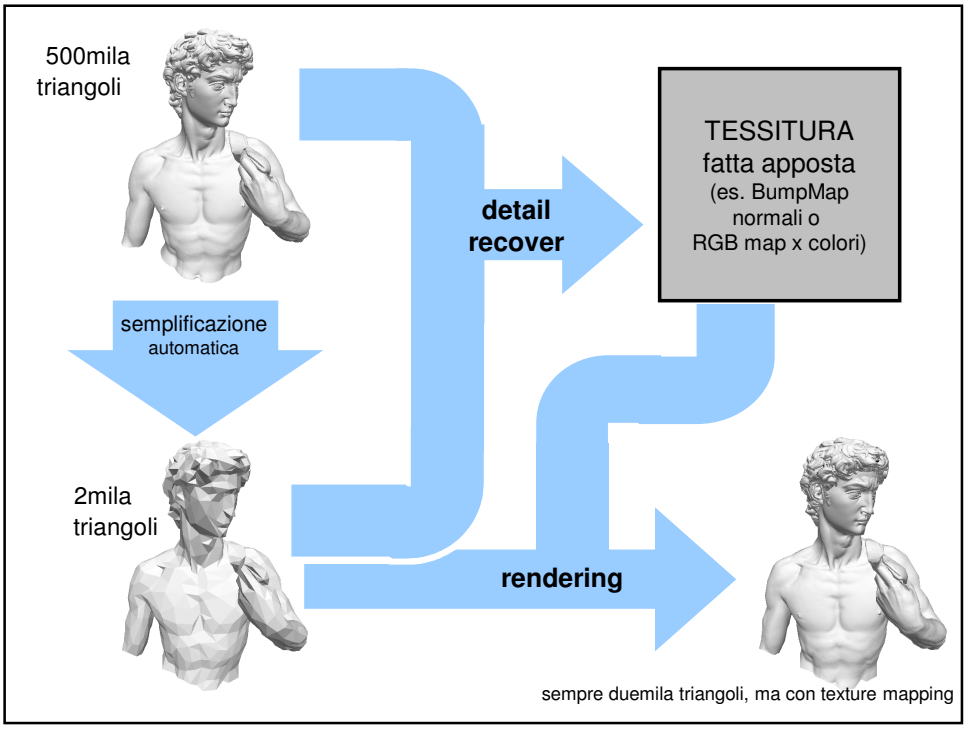


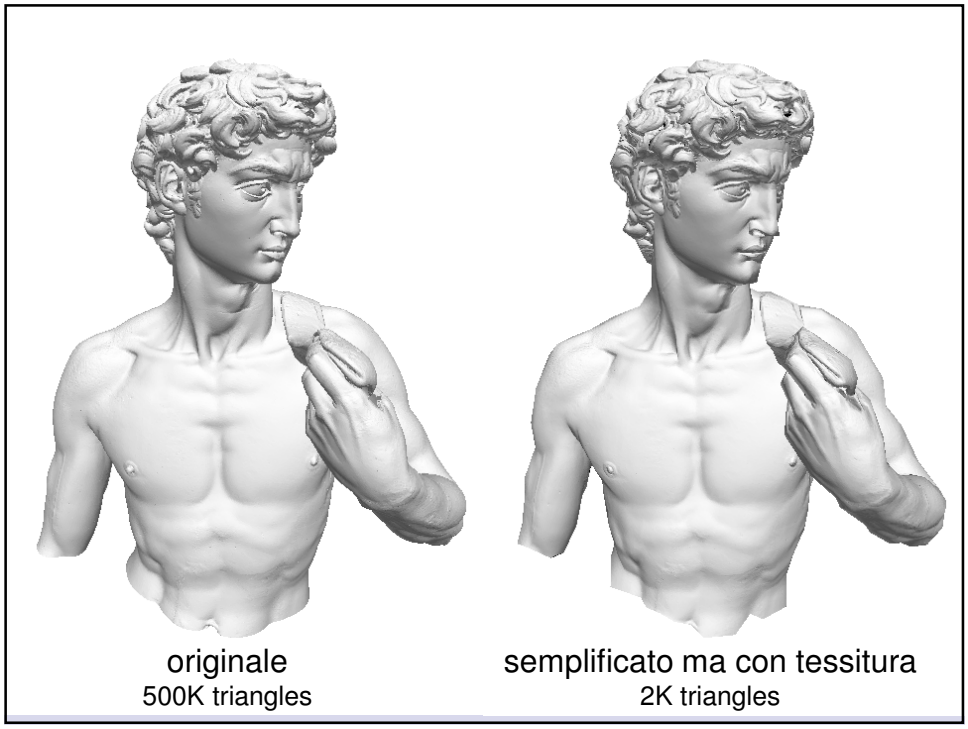
Cohen-Steiner, Alliez, Desbrun (SIGGR04)

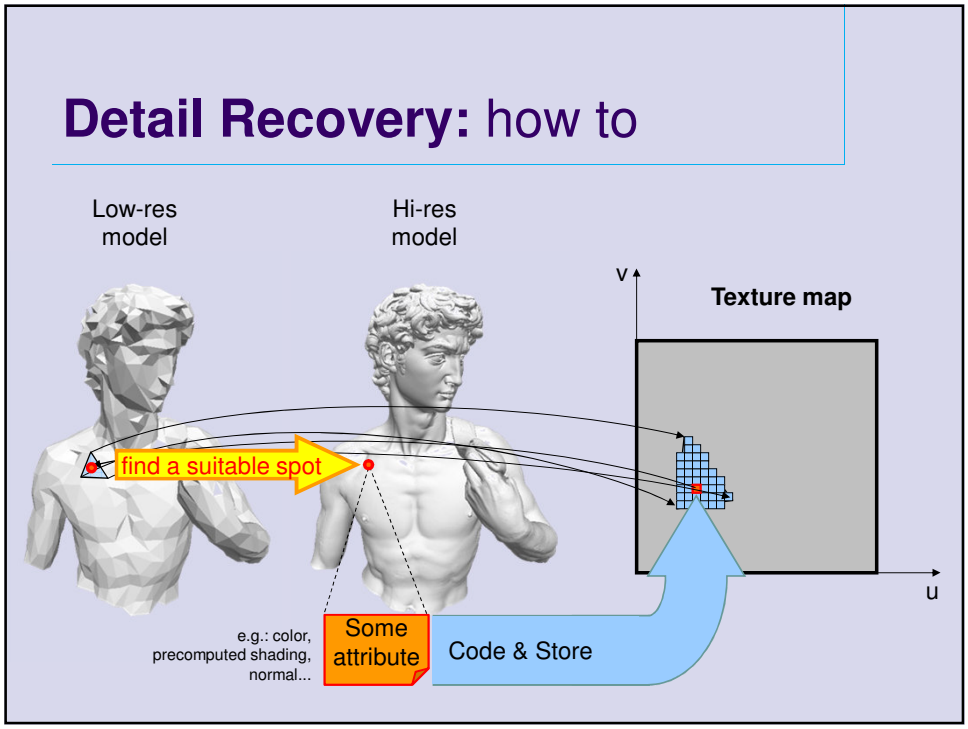
Detail preservation / recovery (o "texture for geometry")

- Idea:
  - data
    - una mesh A low res, uv-mapped
    - una mesh B hir res
  - sintetizzare una tessitura per A
  - per ripristinare il dettaglio hi-freq presente in B

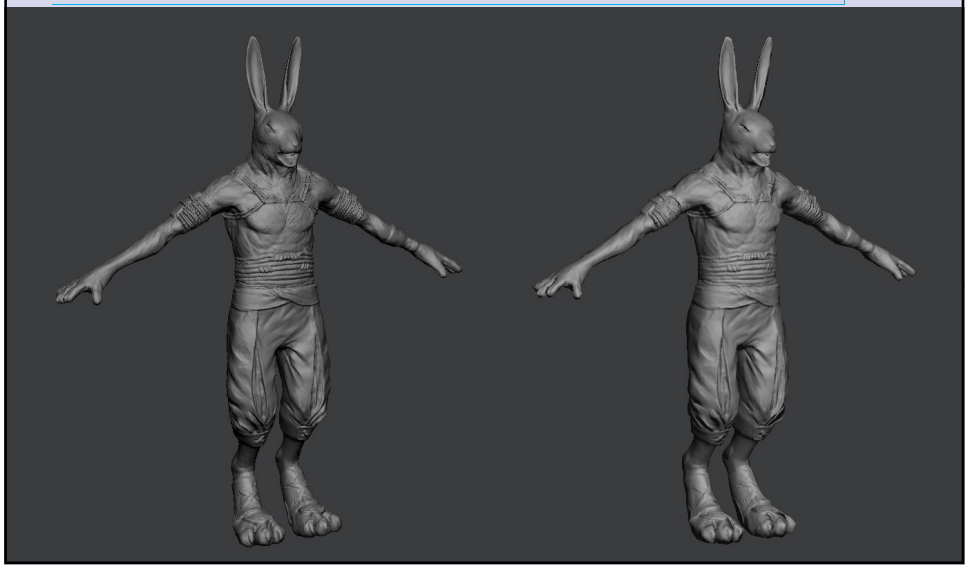
es: A ottenuto da B tramite semplificazione automatica



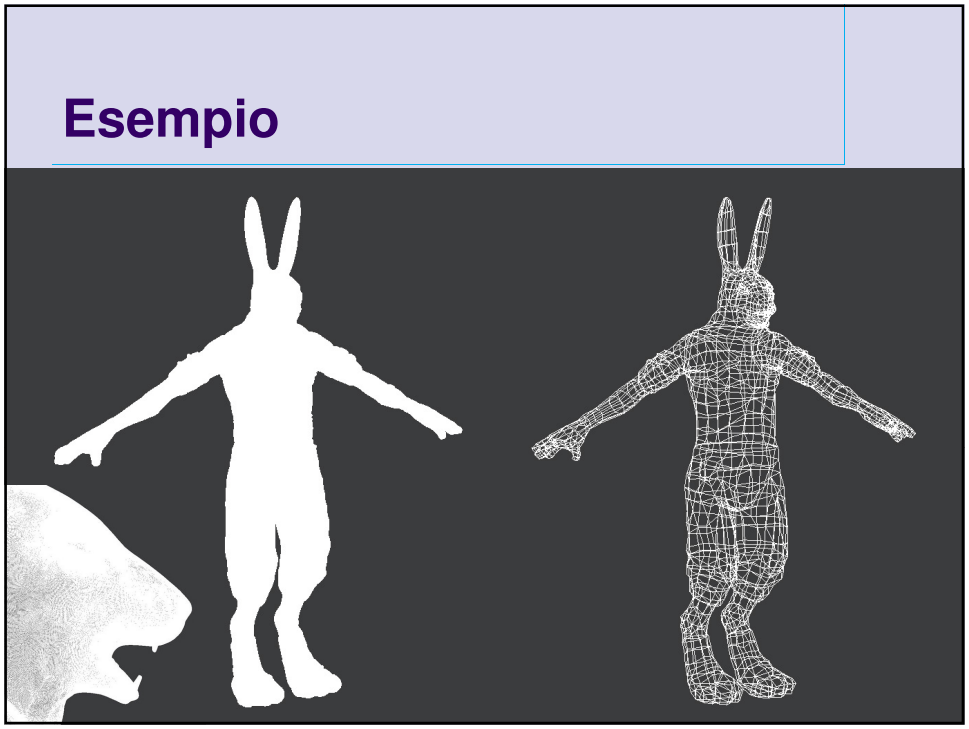

originale
500K triangles
semplificato ma con tessitura
2K triangles

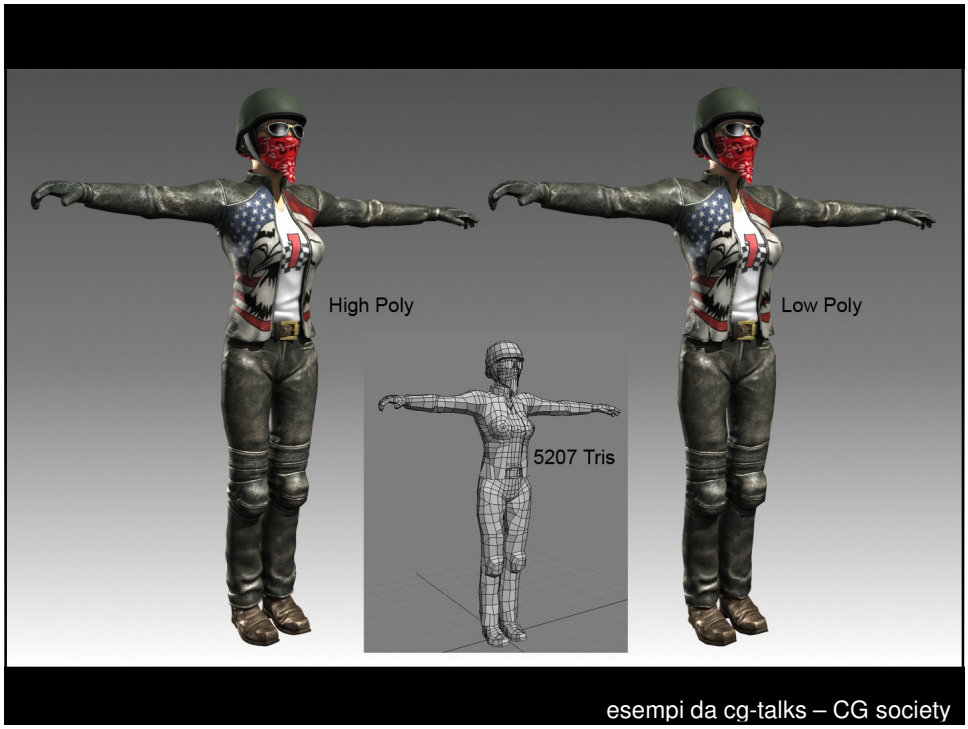

Detail Recovery: how to
Low-res model
Hi-res model
Texture map
v
u
find a suitable spot
e.g.: color, precomputed shading, normal...
Some attribute
Code & Store

Esempio



Esempio




High Poly
5207 Tris
Low Poly
esempi da cg-talks – CG society


TEMPAR
GAME RESOLUTION
REAL TIME CGFX SHADED MODEL
2X 1024 NORM
2X 1024 SPEC
2X 1024 COL
7743 TRIS
High Res Model
Untexured Game Res
Normal mapped Game Res
Final Game Res
esempi da cg-talks – CG society

Mesh: task tipici nella game industry

- Poly reduction / Retopology
  - e.g. LOD construction
- Light baking
  - Precomputazione Luce
  - e.g.: Ambient Occlusion
- U-V mapping
  - parametrizzazione
- Texturing
  - creazione tessiture di vario tipo
- Rigging / Skinning / Animation
  - more about this next time

Una classe di tool utili: attribute transfer

- Attribute transfer
  - Da mesh A a mesh B
  - Retargeting di:
    - Animazioni, UV-mapping, tessiture, …

Solid textures

Solid Textures

- Tessitura volumetrica voxelizzata
- 1 texel == 1 voxel
  - E.g. ogni voxel un colore RGB → solid RGB textures
- Come tutte le tessiture:
  - In video RAM
  - Accesso veloce durante rendering
  - MIP mapping, filtering in accesso…
- Modellano colore di tutto il volume
  - superficie + interno
  - utile, per es, per fratture
- Nota: nessun bisogno di UV-mapping!
  - tessitura indicizzata con geometria mesh
- Solito problema: spazio ram
  - Cubico con risoluzione
  - Soluz: tessiture procedurali?