


134

Mesh processing

- ✓ Il **mesh processing** è semplicemente il **geometry processing** quando è eseguito su mesh poligonali (cioè, molto spesso)
- ✓ Alcuni esempi che abbiamo già visto:
 - ⇒ Computo delle **normali** per faccia
 - ⇒ La stima delle **normali** per vertice, che è un esempio del task generale del passaggio da un **attributo** per faccia ad uno per vertice
 - ⇒ Triangolazione, cioè conversione mesh poligonale generica => tri-mesh (come da **pure-quad** mesh a **tri-mesh** tramite **diagonal split**)
 - ⇒ Test per verificare che una mesh sia **ben-orientata** ("ogni **edge** condiviso da due facce deve essere attraversato in due versi opposti da queste due facce") – anche se non abbiamo visto i dettagli su come implementare questo test efficientemente (task di programmaz.)
 - ⇒ La **surface reconstruction** per produrre una mesh poligonale da una nuvola di punti
- ✓ Vediamone ora altri esempi di task
 - ⇒ e in seguito quali **strutture dati** si prestano alla loro implementazione



136

Mesh processing: mesh repair (o cleaning)

- ✓ Molti algoritmi di mesh processing assumono che le mesh su cui lavorano in input siano consistenti. Siano, per es
 - ⇒ **Two-manifold**, e **ben orientate** (assunzione molto frequente)
 - ⇒ **Chiuse** (a volte assunto – più facile non gestire il caso di edge aperti)
 - ⇒ Non abbiano: triangoli degeneri, vertici o facce ripetute, o mancanti etc.
- ✓ Le mesh reali sono spesso invece spesso “difettate” in vario modo
 - ⇒ Specie quelle catturate (ottenute da **3D acquisition**), (o generate da **nuvole di punti** catturate)
 - ⇒ Anche le mesh usate nel **CAD** presentano spesso vari difetti
- ✓ L'approccio standard consiste nel eseguire **mesh repair** come fase preliminare di ogni task di mesh processing
 - ⇒ Eseguire una “pulizia” delle mesh, rimuovendo difetti e inconsistenze
 - ⇒ Solo per info: nella ricerca, un approccio alternativo emergente consiste nel tentare di trovare algoritmi più robusti e “onnivori”, che lavorino con meno assunzioni (mesh processing detto «in-the-wild»)

137

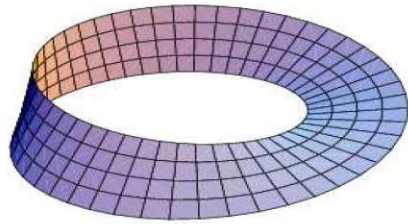
Esempio: orientare consistentemente una mesh

- ✓ Se una mesh non è **ben-orientata**, possiamo “flippare” il verso di alcune facce per renderla tale
- ✓ Idea dell'algoritmo:
 - ⇒ La prima faccia (scelta arbitrariamente) è “ben orientata”.
 - ⇒ Quelle **adiacenti** vengono processate: se non sono consistentemente orientate, allora vengono “flipped” (per es $[1,3,4] \Rightarrow [4,3,1]$, o $[1,4,15,9] \Rightarrow [9,15,4,1]$)
 - ⇒ E così via con le adiacenti, fino a processare tutte le facce
- ✓ Questo procedimento ha sempre successo?
 - ⇒ ...No! Fallisce quando una faccia già processata deve essere di nuovo flippata. Quando succede, significa che era *impossibile* orientare quella mesh consistentemente
 - ⇒ La mesh si dice allora non **orientabile**
 - ⇒ E' raro, ma è possibile!

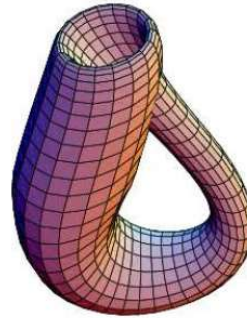
138

Orientare consistentemente una mesh

✓ esempi notevoli di mesh **non orientabili**:



Una mesh che modella un
nastro di Möbius
(non orientabile, aperta)



Bottiglia di Klein
(non orientabile, chiusa)



139

Geometry Processing: mesh cleaning / reparing

- ✓ Mesh cleaning (o reparing):
l'insieme di task per la rimozione dei difetti della di una mesh, cioè
 - ⇒ le situazioni non two-manifold,
 - ⇒ buchi (per es un poligono mancante)
 - ⇒ Orientamento non consistente delle facce
(se non è possibile farlo, ma mesh non è «ben orientabile»)
 - ⇒ Auto-intersezioni
 - ⇒ Replicazioni di vertici
 - ⇒ Facce degeneri
- ✓ Molte categorie di mesh, come le mesh scansionate (acquisizione 3D), presentano spesso molti di questi difetti
- ✓ Spesso, è un preprocessing necessario per consentire l'applicazione di vari altri procedimenti di geometry processing



140

Mesh reparing

Polygon Mesh Repairing: An Application Perspective 15.5

Fig. 1. Illustration of the various types of flaws and defects that can occur in polygon meshes.

Collezione
di software a
<http://meshrepair.org/>

Una pagina che descrive dal survey:
"Polygon Mesh Repairing: An Application Perspective"
Attene, Campen, Kobbelt, 2012

141

Semplificazione automatica di una mesh

- ✓ Da: mesh hi-res
a: mesh low-res (detta anche low-poly mesh)
- ✓ Procedimento chiamato anche
 - ⇒ «Mesh coarsening»
 - ⇒ «Poly-reduction» (soprattutto in ambiente industriale)
 - ⇒ «Mesh decimation»
- ✓ Perché è utile: la risoluzione di una mesh deve essere adatta al contesto applicativo che la usa
 - ⇒ Molti dei procedimenti su una mesh (rendering compreso!) hanno una *complessità lineare* col numero di elementi: maggiore la risoluzione della mesh, più lento l'algoritmo
- ✓ Osservazione:
 - ⇒ in una nuvola di punti, bastava scegliere un sottoinsieme
 - ⇒ per mesh, è più complicato: non posso rimuovere elementi senza danneggiare le proprietà della mesh (es. chiusura)

142

Semplificazione automatica di una mesh: obiettivo

- ✓ Obiettivo: ottenere un buon bilancio fra:
 - ⇒ costo (risoluzione della mesh risultante, cioè numero di elementi residui)
 - ⇒ qualità (errore geometrico introdotto rispetto alla mesh originale)
- ✓ Q: come definisco l'errore introdotto?
 - ⇒ A: misuro la *distanza geometrica* fra le due superfici descritte da mesh originale e mesh semplificata
 - ⇒ cioè assumiamo:
mesh originale = «ground truth»
- ✓ Q: come definisco la distanza fra due superfici?
 - ⇒ def matematica: la risposta esula da questo corso
 - ⇒ per approfondire: cercare «hausdorff distance»
- ✓ Q: come la posso calcolare?
 - ⇒ un task di geometry processing – ma la risposta esula da questo corso
 - ⇒ per approfondire: google search for
«Metro: measuring error on simplified surfaces»

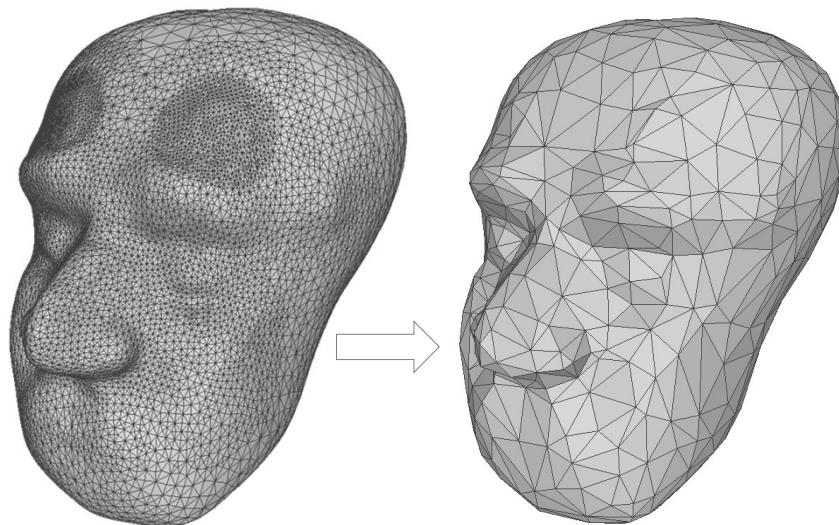


143

Geometry Processing: mesh simplification

Input: 19K facce

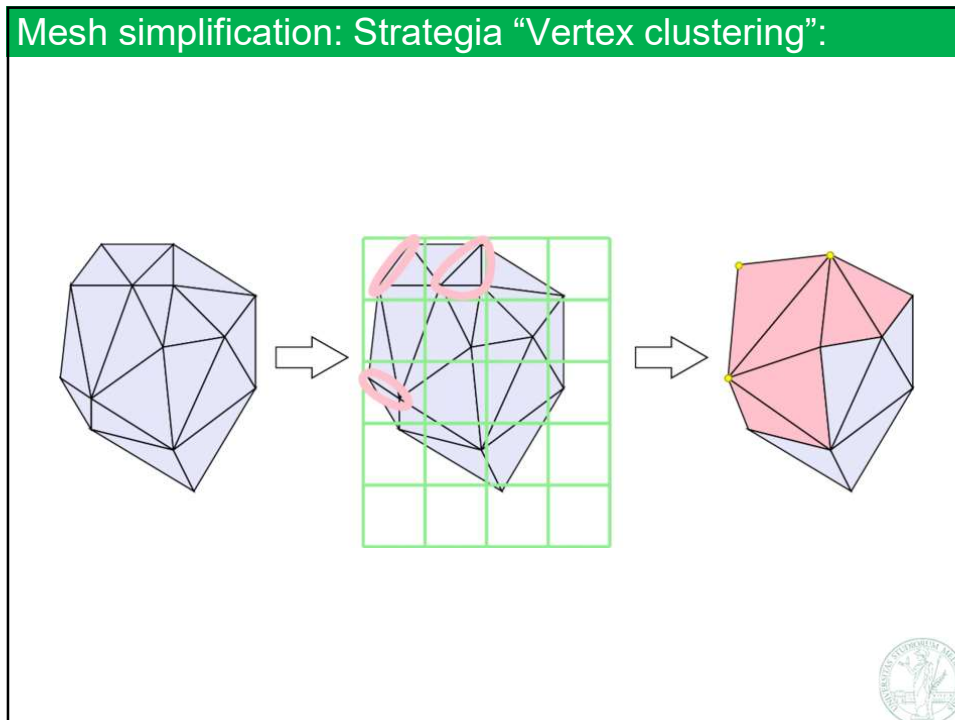
Output: 1200 facce



(esempio eseguito con MeshLab)



145



Mesh simplification: Strategia "Vertex clustering":

1. Suddivido virtualmente lo spazio 3D che contiene la mesh in una griglia regolare di "cubetti", tutti di lato k (un parametro)
2. Per ogni vertice della mesh originale, trovo il cubetto a cui appartiene
 - ⇒ ogni vertice della mesh in input si trova all'interno di un cubetto.
 - ⇒ come trovo questo cubetto, data la posizione del vertice?
3. Per ogni cubetto popolato da almeno un vertice, creo UN SOLO vertice in output, il rappresentante di tutti i vertici dentro quel cubetto
 - ⇒ assegno al nuovo vertice la posizione media di tutti i vertici finiti nel cubetto
 - ⇒ nota: ciascun vertice in input è rappresentato un vertice in output
4. Processo ogni triangolo della mesh originale:
 - se connette tre vertici originali rappresentati ora da tre vertici diversi, allora genero un triangolo che connette quei tre vertici.
 - ⇒ Altrimenti, cioè quando il triangolo connette tre vertici che stanno tutti o in uno solo o in due soli cubetti distinti, scarto il triangolo

147

Mesh simplification: Strategia "Vertex clustering": note

- ✓ Il parametro k controlla la risoluzione della mesh risultante: tanto maggiore è k , tanto minore la risoluzione
 - ⇒ Se k è molto piccolo, la mesh prodotta in output è la stessa della mesh in input
- ✓ L'algoritmo è molto semplice da implementare ed è LINEARE con la risoluzione della mesh
 - ⇒ Infatti bastano due cicli: uno sui vertici e uno sulle facce



148

Mesh simplification: Strategia a operazioni locali

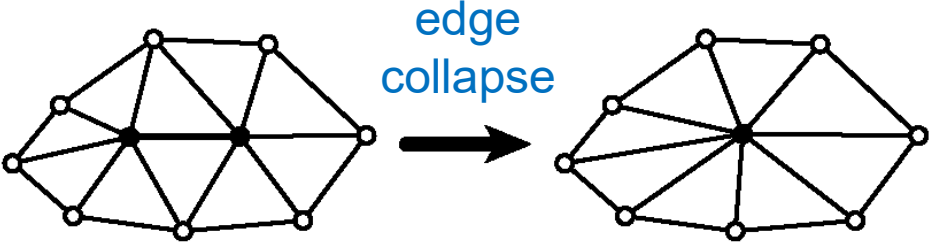
- ✓ Operazione locale: una operazione che coinvolge (e modifica) solo una piccola parte della mesh
 - ⇒ Modifica la geometria e la connettività della mesh solo in piccolo insieme di facce
- ✓ Iterazione di *operazioni locali*
 - ⇒ Repeat
 - scegli un'operazione locale (secondo un qualche criterio)
 - esegui operazione locale (modifica locale della mesh)
 - ⇒ until obiettivo raggiunto
 - es: risoluzione target raggiunta
 - es: errore massimo superato




149

Mesh simplification: Strategia a operazioni locali

✓ Ecco un'operazione locale che reduce la complessità (cioè di "coarsening")



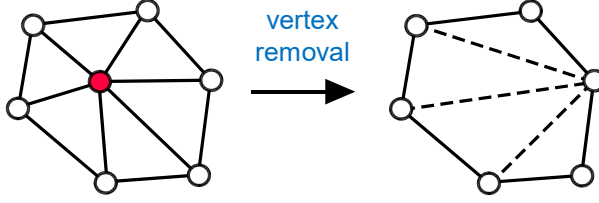
edge collapse



150

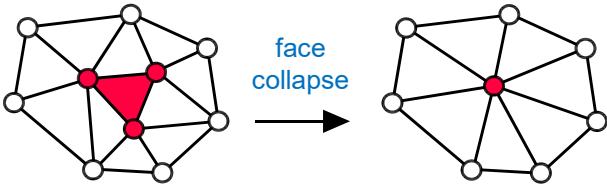
Semplificazione automatica di una mesh

✓ Altri esempi di operazioni locali di coarsening:




vertex removal

nuovi triangoli: che ri-poligonalizzano il buco lasciato dalla rimozione del vertice



face collapse

nota: si può considerare una successione di due edge collapse



151

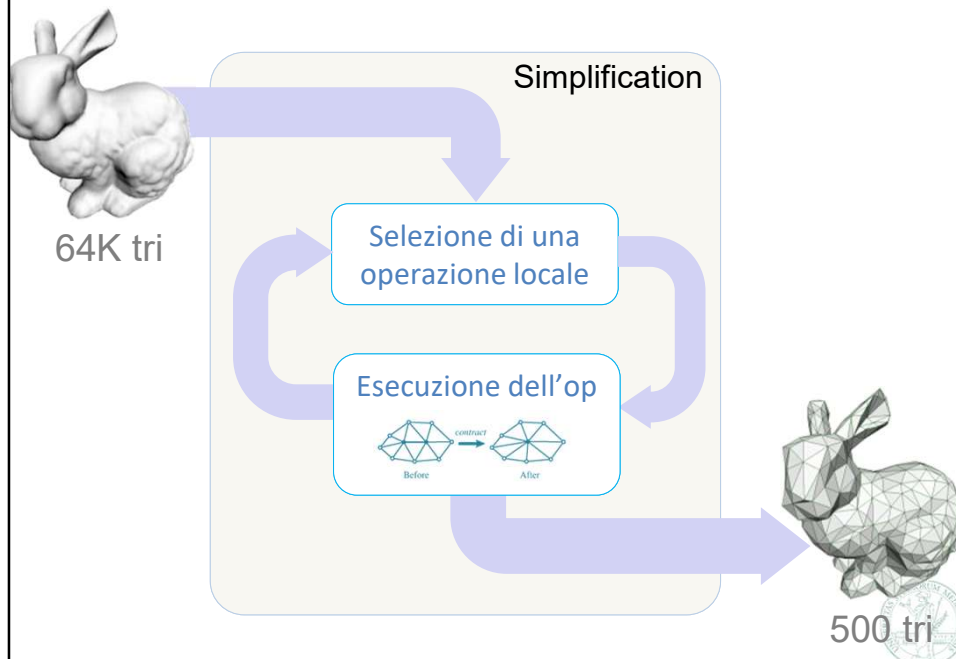
Esercizio

- ✓ Valuta le tre operazioni locali esposte sopra.
- ✓ Per ciascuna di esse, valuta di quanto decrementa la risoluzione della mesh in termini di
 - ⇒ numero di facce
 - ⇒ numero di vertici
 - ⇒ numero di edge
 - ⇒ (nota: non dipende dalla mesh, se l'operazione è fatta su un edge, un vertice, o una faccia INTERNA)
- ✓ Osserva che la proporzione fra facce e vertici e edge di una triangle mesh è mantenuta
 - ⇒ Ricorda che, come regola generale, una tri-mesh di n vertici ha circa $2n$ facce e $3n$ edges



152

Semplificazione automatica di una mesh



153

Mesh simplification: Strategia a operazioni locali

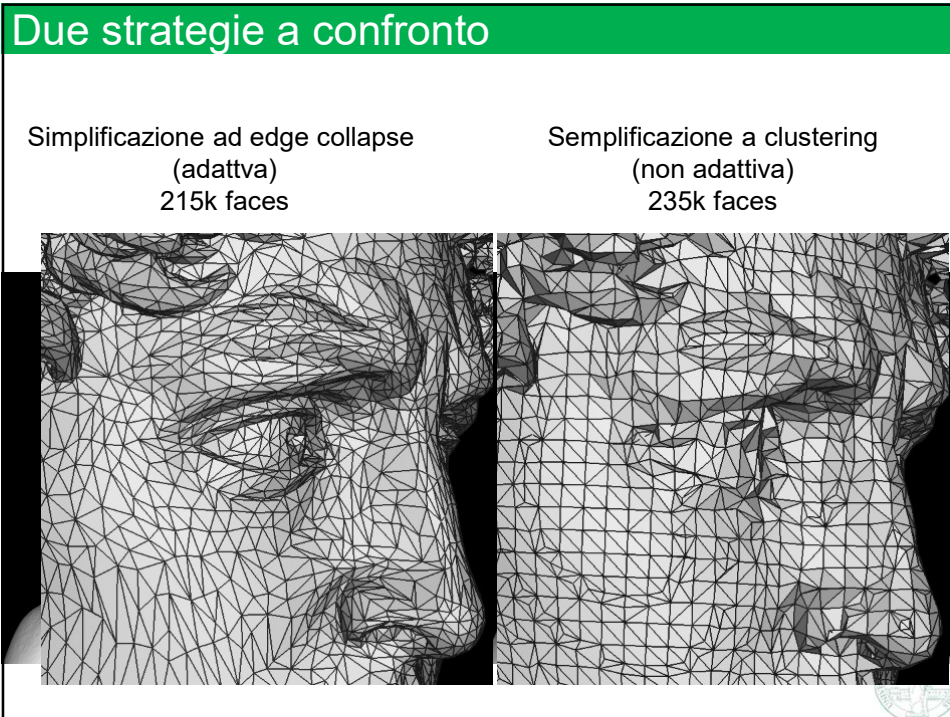
- ✓ Quali operazioni locali sono possibili?
 - ⇒ (ad un dato step)
 - ⇒ Per l'edge collapse, ogni edge della mesh rappresenta una possibile scelta
- ✓ Quale operazione locale scegliere?
 - ⇒ Possibili molti criteri: ad esempio...
 - ⇒ Quella che introduce un errore di approssimazione più contenuto
 - ⇒ Escludo dalla scelta le operazioni che inficiano le caratteristiche della mesh che mi interessa mantenere (come two-manifoldness)
- ✓ Quanto terminare il processo? (criteri possibili)
 - ⇒ Quando la risoluzione scende sotto un livello prefissato,
 - ⇒ Oppure: quando tutte le operazioni residue introducono un errore troppo elevato

154

Due strategie a confronto

- | Semplificazione ad operaz locali | Semplificazione a clustering |
|---|---|
| ✓ adattiva | ✓ non adattiva |
| ✓ continua | ✓ discreta |
| ✓ possibile mirare a triangoli di forma buona | ✓ mira a facce di dimensione simile |
| ✓ possibile specificare errore massimo | ✓ possibile solo specificare dimensione griglia (solo indirettamente, errore o numero di facce) |
| ✓ possibile specificare numero di facce target | ✓ non mantiene caratteristiche mesh |
| ✓ possibile mantenere caratteristiche mesh <ul style="list-style-type: none">⇒ two-manifoldness, chiusura, classe topologia | ✓ veloce, semplice da implementare, robusta (funziona su qualsiasi mesh) |
| ✓ spesso <i>richiede</i> mesh two-manifold in partenza | |


155



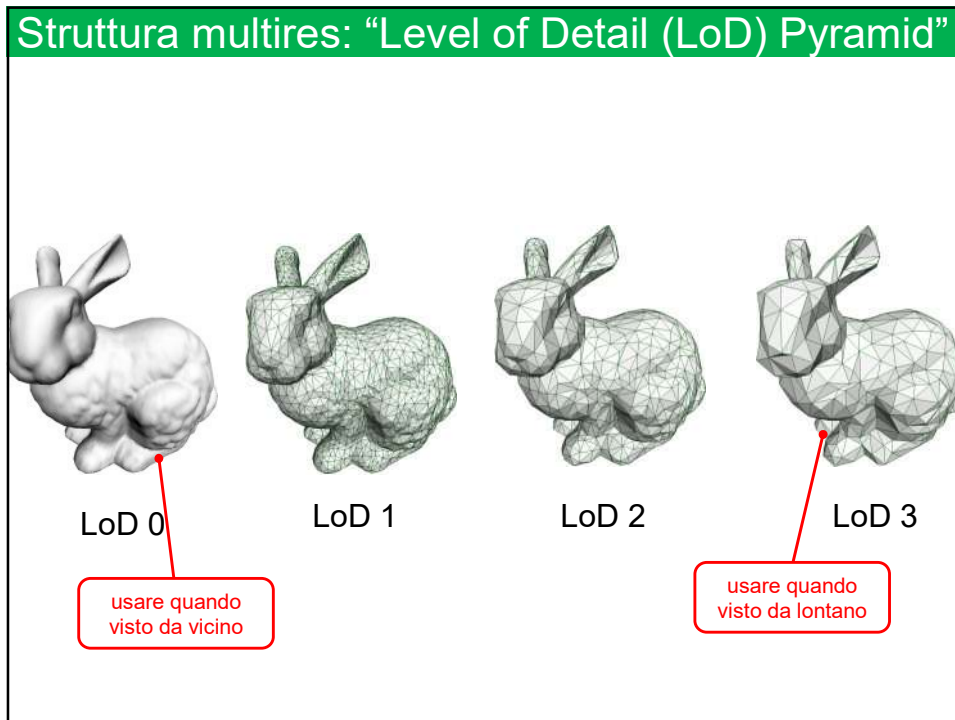
156

Semplificazione automatica di una mesh

- ✓ Mini-esercizio pratico:
testa gli algoritmo di mesh decimation
 - ⇒ Ottieni e apri MeshLab:
 - ⇒ Scarica una mesh ad alta risoluzione di esempio (ce ne sono anche sul sito)
 - ⇒ Applica uno di questi due filtri:
 - filtro «quadric edge collapse decimation»
 - filtro «clustering decimation»
 - ⇒ Paragona i risultati



157



158

Piramide di livelli di dettaglio

- ✓ Idea: tenere in memoria tante versioni diverse di una mesh, a risoluzione decrescente
 - ⇒ LoD (Level of Detail) = Livello di dettaglio = una di queste versioni
 - ⇒ LoD-0 = la versione a piena risoluzione
 - ⇒ LoD-1 = versione a risoluzione ridotta
- ✓ L'insieme dei LoD è detto «piramide» di livello di dettaglio perché, al pari di una piramide, la base (LoD-0) è assai più grande (in RAM o sul disco) della punta (LoD- n).
- ✓ In preprocessing, i livelli di dettaglio possono essere ottenuti uno dall'altro attraverso mesh-simplification
- ✓ In fase di rendering, scelgo il LoD in funzione di fattori come
 - ⇒ distanza dell'oggetto dalla telecamera, le risorse di computazione a disposizione, l'importanza dell'oggetto in quel momento, etc.
 - ⇒ le stesse considerazioni valgono per una simulazione fisica o altro
- ✓ Costo: duplicazione di dati in memoria ☹️
- ✓ Beneficio: minor tempo di rendering o processing in generale

159

Calcolo di distanza fra due mesh

- ✓ Task: date due mesh M_0 e M_1 , quantificare quanto simili (o dissimili) siano le due superfici rappresentate
 - ⇒ Nota: mesh anche molto diverse internamente (diversi poligoni e vertici) possono rappresentare superfici molto simili, o anche *identiche*
 - ⇒ (sapresti dare un esempio?)
- ✓ E' utile a questo scopo l'uso della distanza detta "**distanza di Hausdorff**" definita fra due superfici
 - ⇒ Evitiamo qui la definizione rigorosa, ma se vale (mettiamo) 3mm significa che le due superfici sono a 3mm una dall'altra, "al massimo"
 - ⇒ È zero se e solo se le due superfici coincidono (anche se le mesh che le rappresentano sono diverse)
- ✓ Appositi algoritmi di geom. Proc. misurano questa distanza
 - ⇒ Meshlab include una implementazione di questi algoritmi
- ✓ La distanza fra mesh è spesso usata in altri contesti di geom processing per valutare l'efficacia di un algoritmo
 - ⇒ Ad es: valutare l'errore introdotto da una semplificazione, misurando la distanza fra mesh originale e mesh semplificata



161

Calcolo della distanza di Hausdorff fra 2 mesh

- ✓ Altri usi di questa misura:
 - ⇒ Misura della deformazione di un manufatto nel corso del tempo («quanto si sta deformando questo mobile antico sotto l'effetto dell'umidità?»)
 - ⇒ Passo 1: scansionare il manufatto ad intervalli di tempo regolari
 - ⇒ Passo 2: allineare i due oggetti il meglio possibile uno sull'altro (ad esempio, allineando reciprocamente le nuvole di punti definite dai loro vertici, attraverso ICP)
 - ⇒ Passo 3: calcolo della distanza di Hausdorff per avere la risposta



164

Geometry Processing: remeshing

✓ **Remeshing:**

data una superficie già rappresentata come una mesh, costruire una rappresentazione mesh... diversa

✓ La mesh di partenza differisce dalla mesh di arrivo in termini di, per es...

- ⇒ da tri a quad-dominant o pure quad («quad-remeshing»)
- ⇒ da irregolare a (semi)regolare («semiregular-remeshing»)
- ⇒ da risoluzione non adattiva a risoluzione adattiva
- ⇒ da forma sfavorevole* delle facce a buona forma (* per esempio, triangoli lunghi e stretti o degeneri)

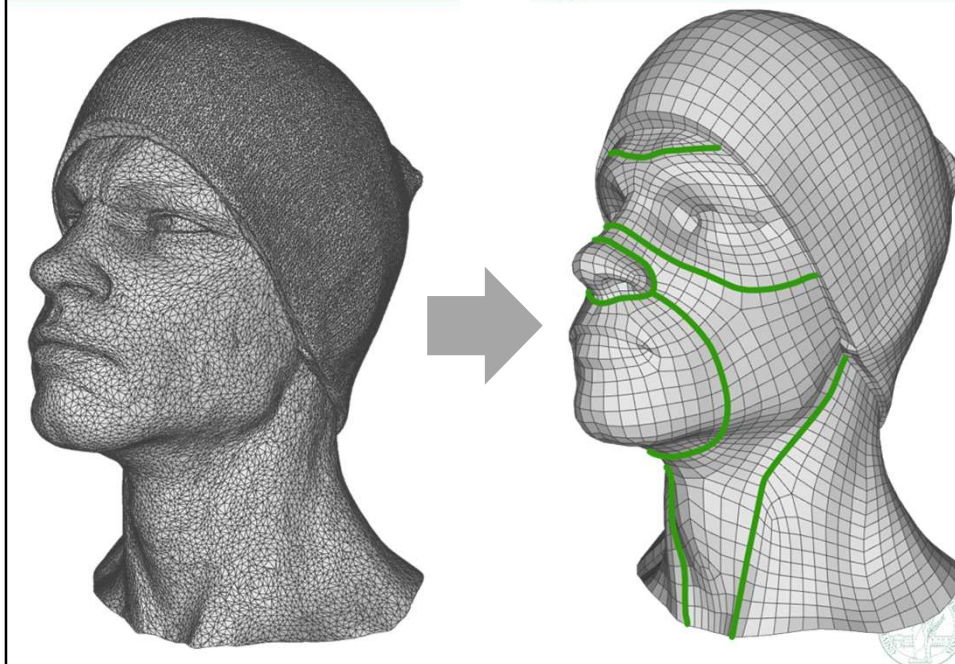
✓ In ambiente industriale, è detto anche “**retopology**”

- ⇒ A volte viene eseguito manualmente da un'artista digitale (un modellatore 3D)
- ⇒ Esistono anche algoritmi automatici



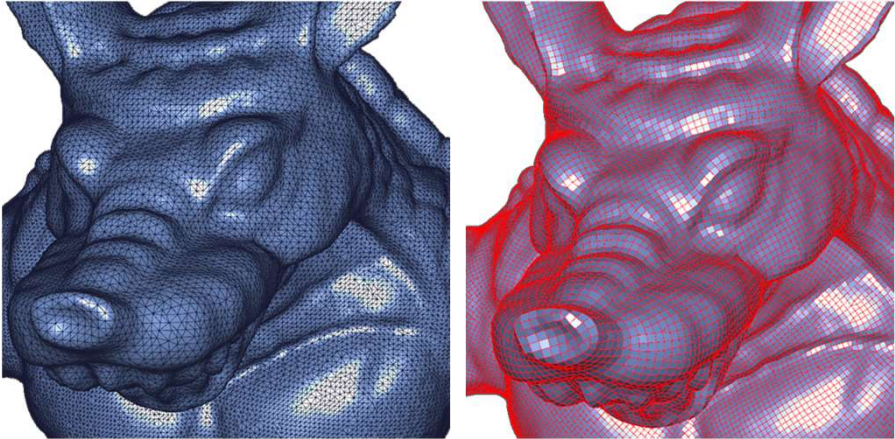
166

Semiregular Quad Remeshing: esempio



168

Semiregular Quad Remeshing: esempio



Input:
irregular tri mesh

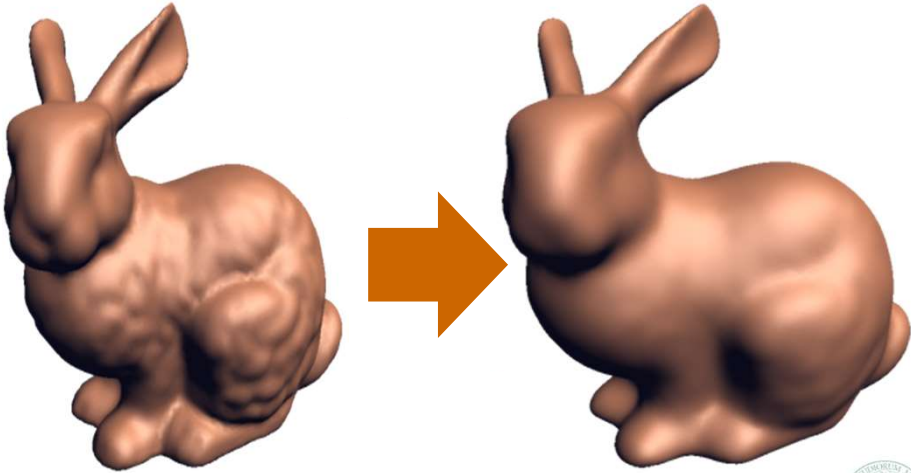
Output:
semiregular quad mesh

Esempio ottenuto con : «*instant field-aligned meshes*»

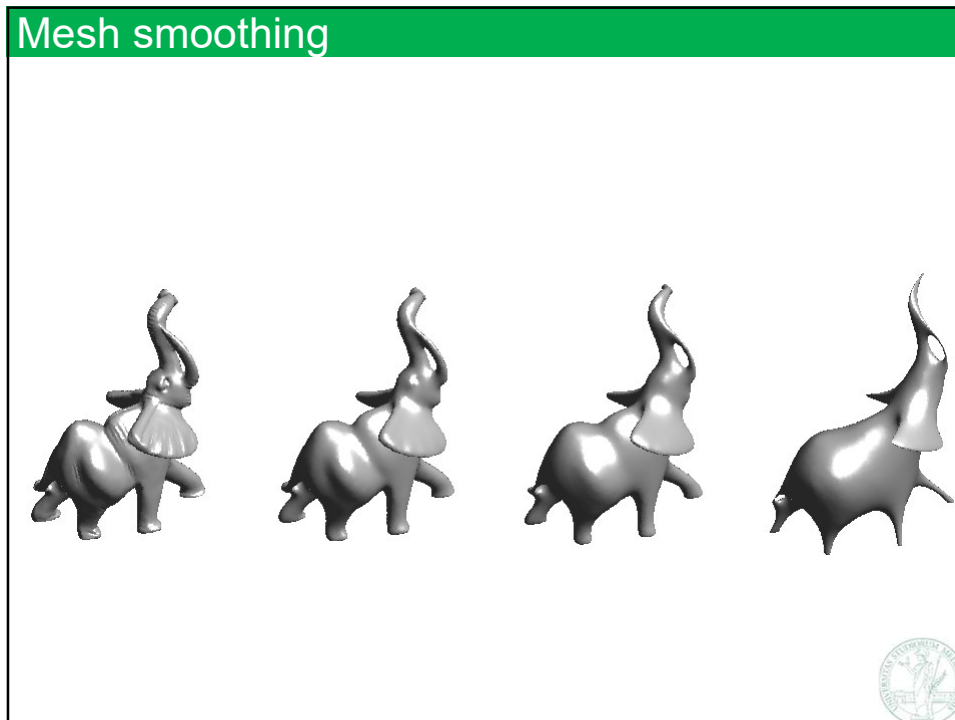
171

Geometry Processing tasks: examples

✓ Mesh smoothing




172



173

Mesh smoothing

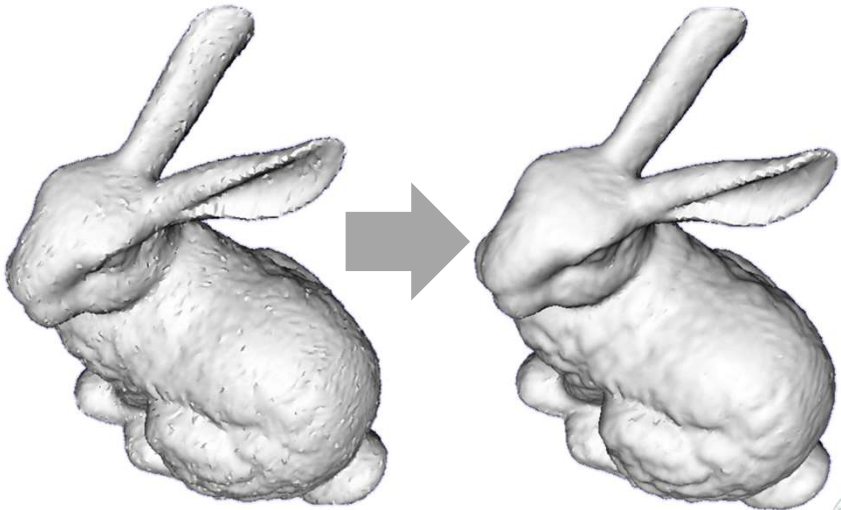
- ✓ Obiettivo: rendere la mesh più tondeggiante e liscia
- ✓ How to (concettualmente, base):
basta spostare ogni vertice nella posizione media dei vertici a lui adiacenti
 - ⇒ vertici adiacenti = vertici connessi da un edge
 - ⇒ Nota: la **connettività** della mesh rimane inalterata: si modifica solo la sua **geometria**
- ✓ Il processo si può ripetere molte volte, cumulandone gli effetti
- ✓ Analogo concettuale del processo di sfocatura (*blur*) di un'immagine
 - ⇒ se si sostituisce ad ogni pixel il valore medio dei pixel vicini, l'immagine diviene sfocata
- ✓ Concetto matematico connesso: filtro «Laplaciano»
 - ⇒ Minimizzazione del Laplaciano, cioè della distanza di ogni vertice dalla media dei suoi vicini
- ✓ L'operazione può essere fatta anche sugli attributi (colore, etc)
 - ⇒ oppure, *solo* sugli attributi
- ✓ Effetti desiderati:
 - ⇒ si abbatte il rumore: e quindi una forma di denoising
 - ⇒ si riducono le asperità della superficie
- ✓ Effetti collaterali (spesso indesiderati):
 - ⇒ si abbattano anche i dettagli di forma utili (ad alta frequenza)
 - ⇒ si perdono ad esempio gli spigoli vivi («crease angle» o «feature lines»)
 - ⇒ il volume della mesh tende a ridursi



174

Geometry Processing tasks: examples

✓ Mesh denoising



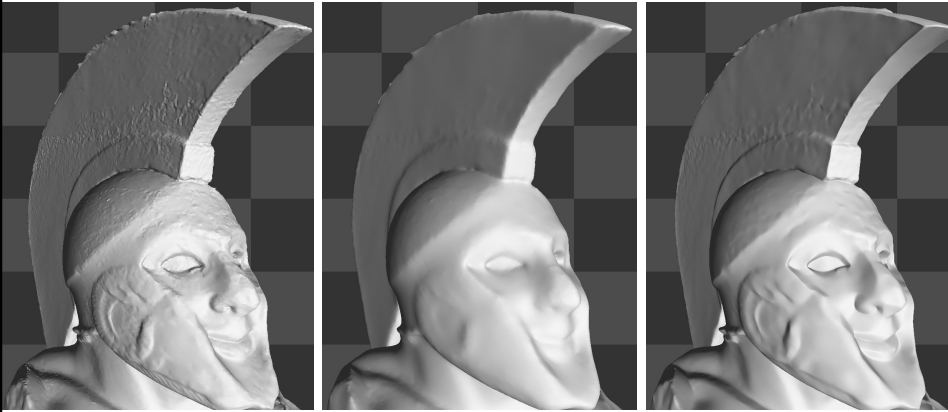
(img by Iman Sadeghi)

175

Mesh feature preserving smoothing

✓ Feature preserving smoothing

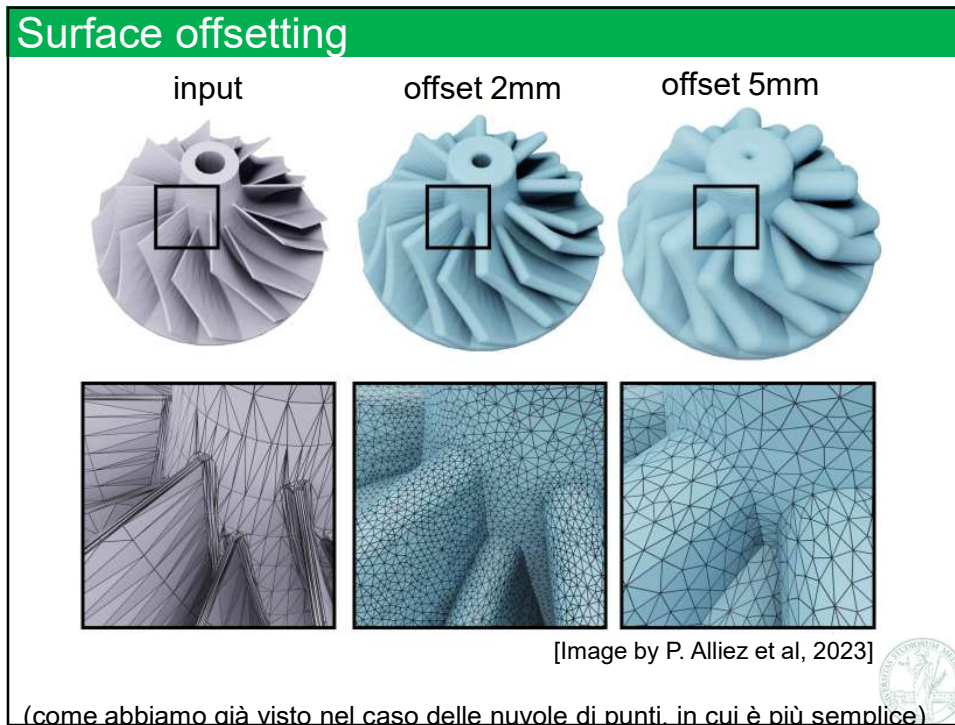
- ⇒ Ottenere uno smoothing che rimuove il rumore e le asperità a piccola scala ma evita gli «effetti collaterali» sopra descritti.
- ⇒ Task studiato e più difficile



Originale (mesh scansionata) Smoothing Feature preserving smoothing

(images by Andrea Maggiordomo)

176



177

Stima di proprietà geometriche locali


- ✓ La mesh può essere processata per ottenere, per ogni elemento (faccia, vertice, o anche edge) la stima di proprietà geometriche (cioè relative alla forma, della superficie attorno a quell'elemento)
 - ⇒ Proprietà di cui si occupa la geometria differenziale
- ✓ Il risultato può essere immagazzinato in appositi **attributi**
- ✓ Abbiamo visto il caso della stima delle normali (orientamenti locali della superficie) per faccia e per vertice
- ✓ Altri casi:
 - ⇒ Curvatura (variazione di normale, definibile in vario modo)

Img by Karan Singh

178

Stima di proprietà geometriche globali

- ✓ Con appositi algoritmi, è possibile (e semplice) stimare proprietà geometriche globali della superficie rappresentata o del solido racchiuso (se esiste) come
 - ⇒ Area complessiva
[questo sappiamo farlo! Sapresti scrivere uno pseudocodice per ottenere questo risultato?]
 - ⇒ Volume complessivo (richiede che la mesh sia two-manifold e chiusa, ma esistono algoritmi «on the wild» che lo stimano senza questa assunzione)
 - ⇒ Posizione del baricentro
 - ⇒ Momento di inerzia
 - ⇒ ...




179

Mesh processing: per saperne di più...

Una conferenza internazionale su Geometry Processing che si è svolta nel nostro ateneo alcuni anni fa:

Graduate school: lezioni introduttive ad argomenti avanzati di geometry processing (i video sono ancora disponibili!)

<https://sgp2019.di.unimi.it/>



180