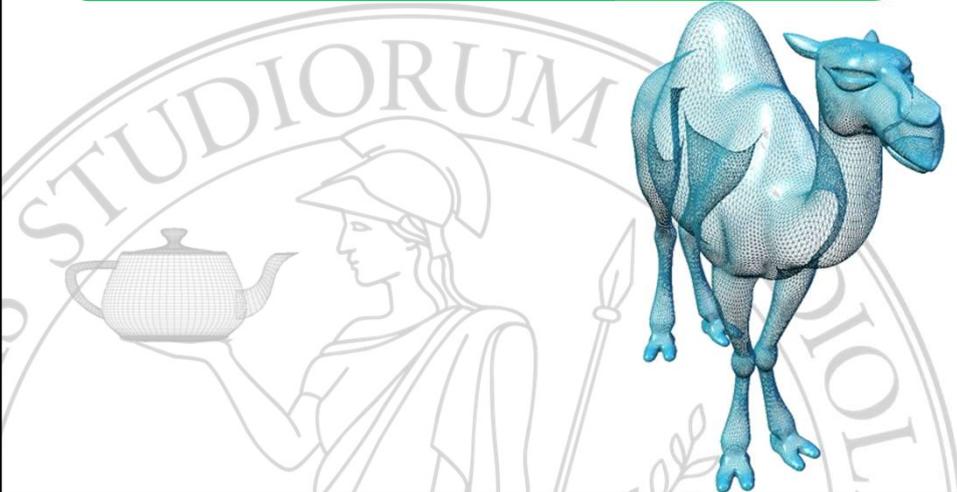


Marco Tarini - Computer Graphics 2022/2023
Università degli Studi di Milano

**Polygonal Representations:
cenni di Mesh Processing 2/2**



102

Alcune librerie di mesh processing (C++, OpenSource)

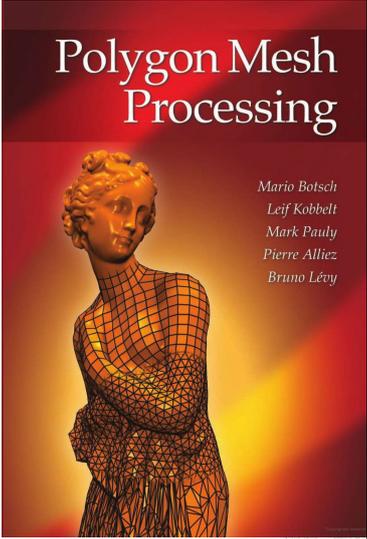
 <p>VCG-Lib vision and computer graphic library CNR ()</p>	 <p>computational geometry algorithms library INRIA ()</p>
 <p>OpenMesh +  OpenFlipper RWTH ()</p>	 <p>libigl simple geometry processing library NYU ()</p> 

103

Mesh processing

Il **Geometry Processing** eseguito su **mesh poligonali**

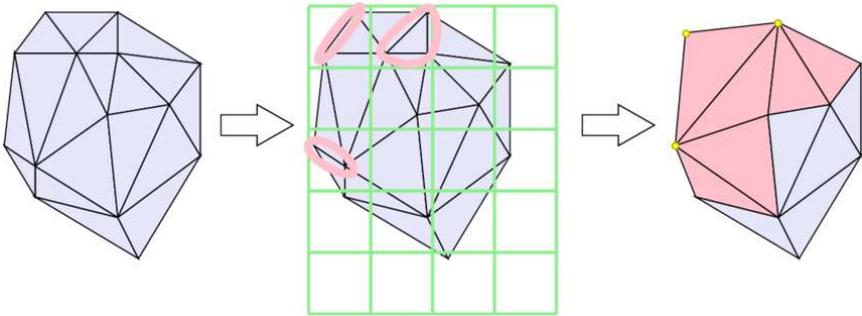
Un buon manuale per le basi al mesh processing:



<http://www.pmp-book.org/>

104

Mesh simplification: Strategia "Vertex clustering":



105

Mesh simplification: Strategia "Vertex clustering":

1. Suddivido virtualmente lo spazio 3D che contiene la mesh in una griglia regolare di "cubetti", tutti di dimensione k
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, che rappresenta tutti i vertici dentro quel cubetto,
 - ⇒ assegno al nuovo vertice assegno la posizione media di tutti i vertici originali nel cubetto
 - ⇒ ogni vertice originale è rappresentato da uno dei nuovi vertici
4. Processo ogni triangolo della mesh originale: se connette tre vertici originali rappresentati da tre vertici diversi, allora genero un triangolo che connette i tre vertici.
 - ⇒ Altrimenti, scarto il triangolo



106

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
 - ⇒ Bastano due cicli, uno sui vertici e uno sulle facce



107

Mesh simplification: Strategia a operazioni locali

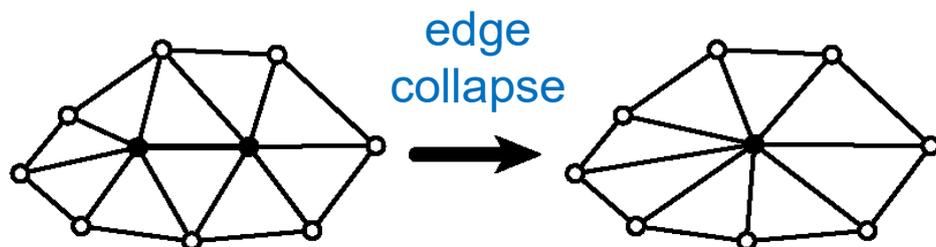
- ✓ Operazione locale: una operazione che coinvolge (e modifica) solo una piccola parte della mesh
 - ⇒ Modifica la geometria e la connettività della mesh solo in un intorno
- ✓ Iterazione di *operazioni locali*
 - ⇒ Repeat
 - scegli un'operazione locale (secondo un criterio)
 - esegui operazione locale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



108

Mesh simplification: Strategia a operazioni locali

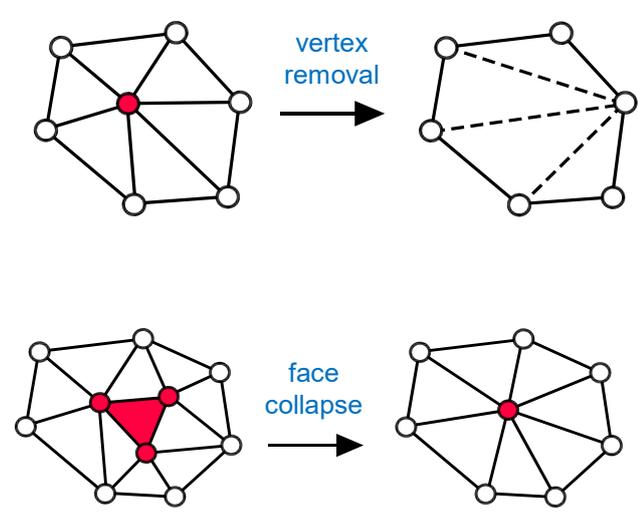
- ✓ Esempio di operazione locale di riduzione della complessità (cioè "coarsening")



109

Semplificazione automatica di una mesh

✓ Altri esempi di operazioni locali di coarsening:



The diagram shows two examples of local coarsening operations on a mesh. The first example, labeled 'vertex removal', shows a mesh with a central red vertex being removed. Dashed lines indicate the new triangles formed by re-polygonizing the hole. The second example, labeled 'face collapse', shows a mesh with a red triangle being collapsed into a single red vertex. The text explains that this can be considered a succession of two edge collapses.

vertex removal

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

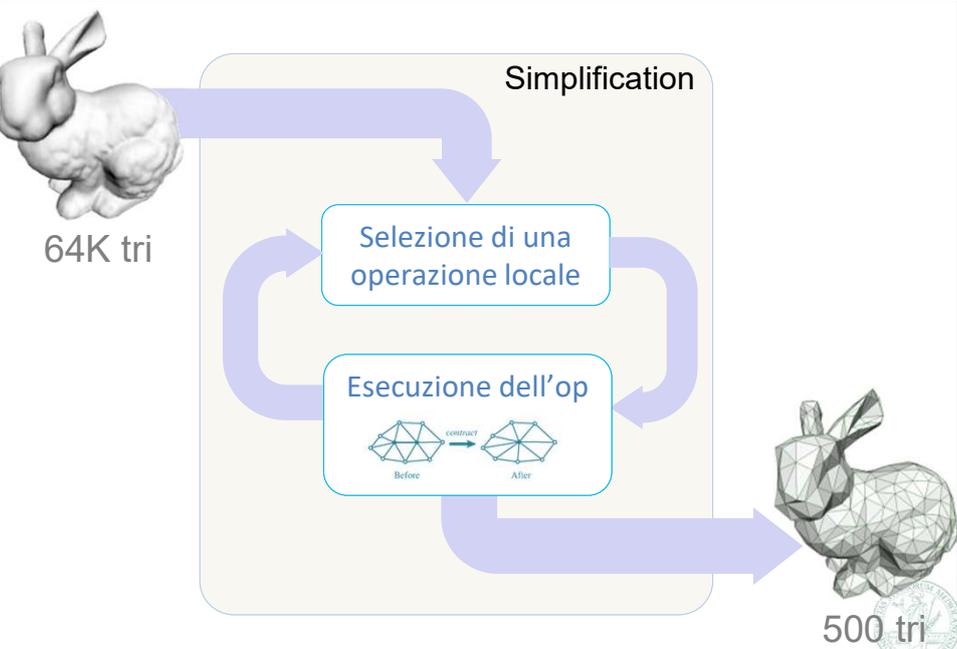
face collapse

nota: si può considerare una successione di due edge collapse



110

Semplificazione automatica di una mesh



The flowchart illustrates the automatic mesh simplification process. It starts with a high-resolution mesh of a rabbit labeled '64K tri'. This mesh enters a 'Simplification' box. Inside this box, there is a loop of two steps: 'Selezione di una operazione locale' (Selection of a local operation) and 'Esecuzione dell'op' (Execution of the op). The 'Esecuzione dell'op' step includes a small diagram showing a 'Before' state with a red triangle and an 'After' state with a red vertex. The process continues until it reaches a low-resolution mesh of the rabbit labeled '500 tri'.

64K tri

Simplification

Selezione di una operazione locale

Esecuzione dell'op

Before After

500 tri



111

Mesh simplification: Strategia a operazioni locali

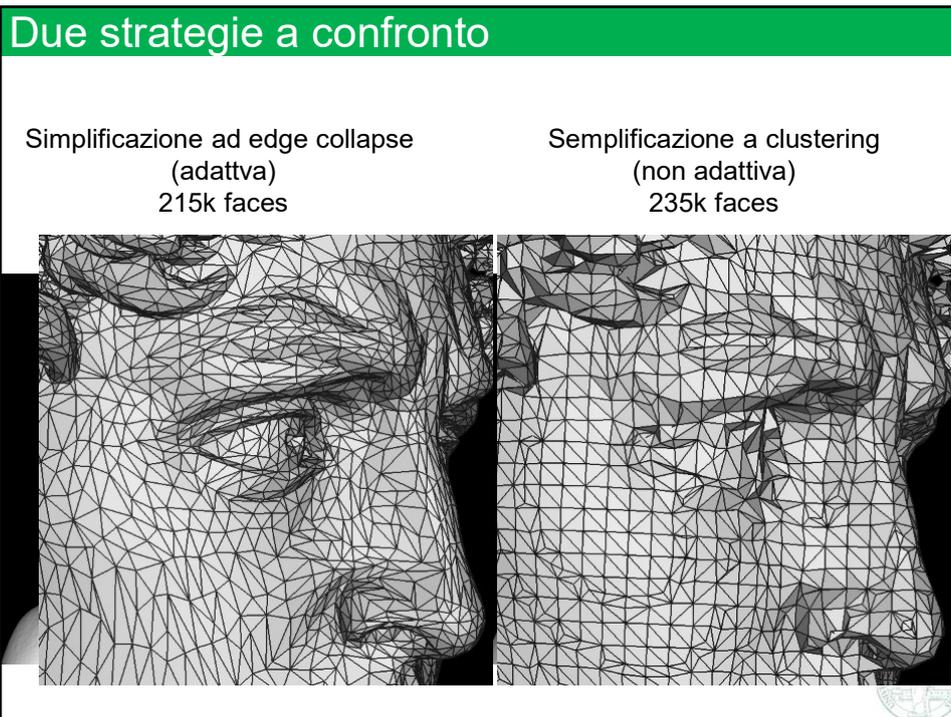
- ✓ Quali operazioni locali sono possibili?
 - ⇒ (ad un dato step)
 - ⇒ Per l'edge collapse, ogni edge della mesh rappresenta una possibile scelta
- ✓ Quale operazione locale scegliere?
 - ⇒ Possibili molti criteri: ad esempio...
 - ⇒ Quella che introduce un errore di approssimazione più contenuto
 - ⇒ 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

112

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 | |

113



114

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 di Hausdorff" definita fra due superfici
 - ⇒ È zero se e solo se le due superfici coincidono (anche se le mesh che le rappresentano sono diverse)
 - ⇒ Vedi il prossimo lucido per la sua definizione esatta, su una mesh
- ✓ La misurazione di questa distanza richiede apposite algoritmi di geometry processing
 - ⇒ 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



115

Definizione della distanza di Hausdorff fra 2 mesh

- ✓ Distanza punto → mesh:

*dato un punto \mathbf{p} , quanto dista
il punto \mathbf{q} su M_1 più vicino possibile a \mathbf{p} ?*

$$\text{dist}(\mathbf{p}, M_0) = \min_{\mathbf{q} \in M_1} (\|\mathbf{p} - \mathbf{q}\|)$$

⇒ nota: \mathbf{q} non è necessariamente un vertice di M_1 ;
può anche essere un punto interno ad un triangolo.

- ✓ Distanza mesh → mesh:

*data una mesh M_0 , quanto dista al massimo
un punto \mathbf{p} scelto su M_0 da una seconda mesh M_1 ?*

$$\text{dist}(M_0, M_1) = \max_{\mathbf{p} \in M_0} (\text{dist}(\mathbf{p}, M_1)) = \max_{\mathbf{p} \in M_0} \left(\min_{\mathbf{q} \in M_1} (\|\mathbf{p} - \mathbf{q}\|) \right)$$

⇒ nota: \mathbf{p} non è necessariamente un vertice di M_0 ;
può anche essere un punto interno ad un triangolo.

⇒ Nota: $\text{dist}(M_0, M_1)$ non è necessariamente uguale a $\text{dist}(M_1, M_0)$

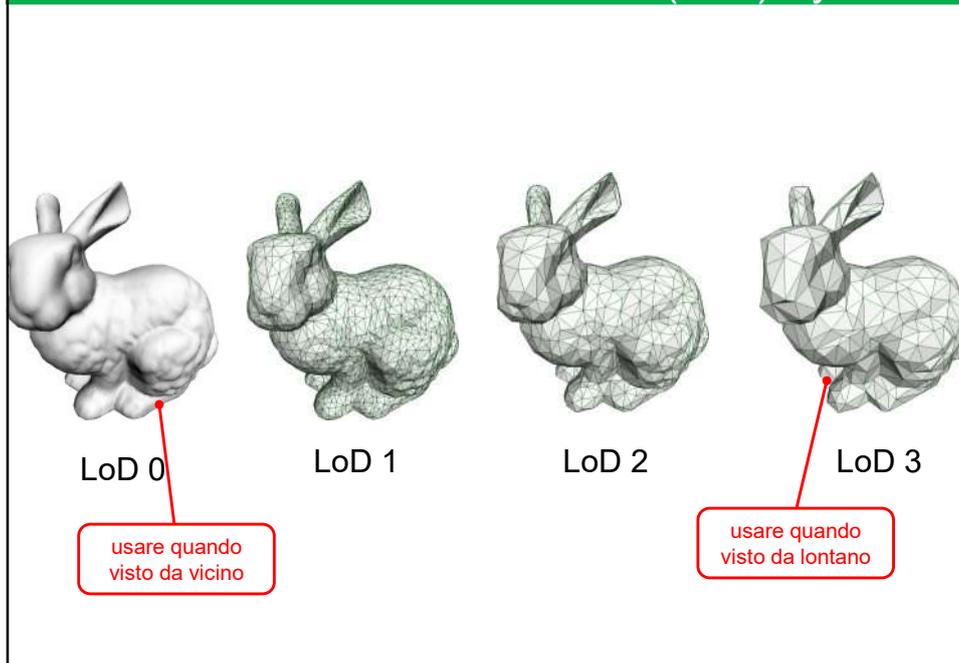
- ✓ Distanza reciproca mesh ↔ mesh

$$\text{Hausdorff}(M_0, M_1) = \max(\text{dist}(M_0, M_1), \text{dist}(M_1, M_0))$$



116

Struttura multires: "Level of Detail (LoD) Pyramid"



117

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), 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, risorse di computazione a disposizione, 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



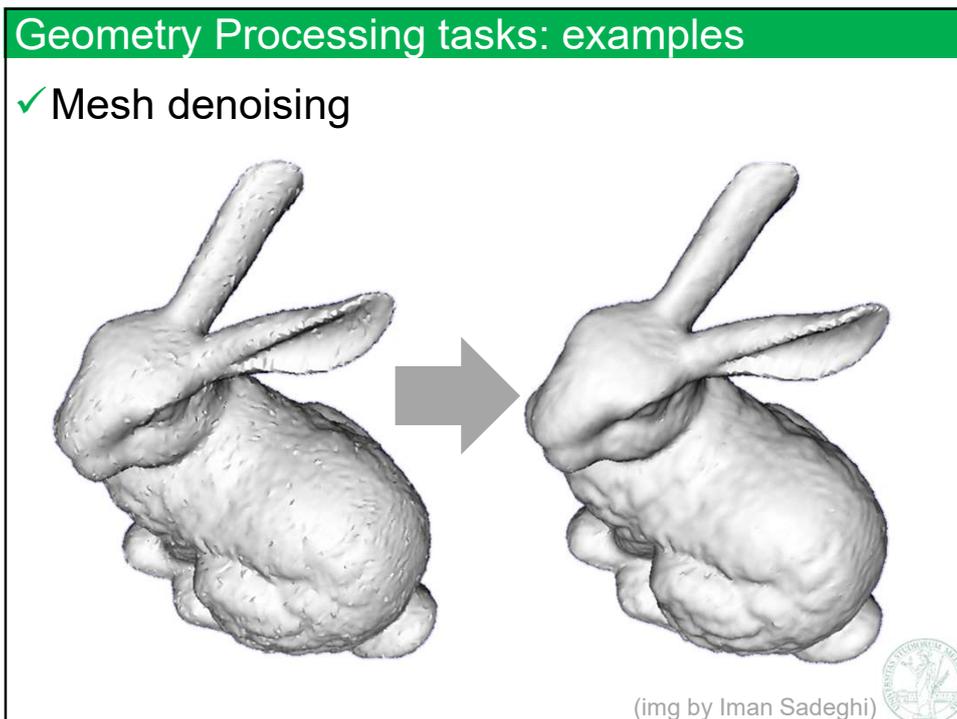
118

Piramide di livelli di dettaglio: stima del costo

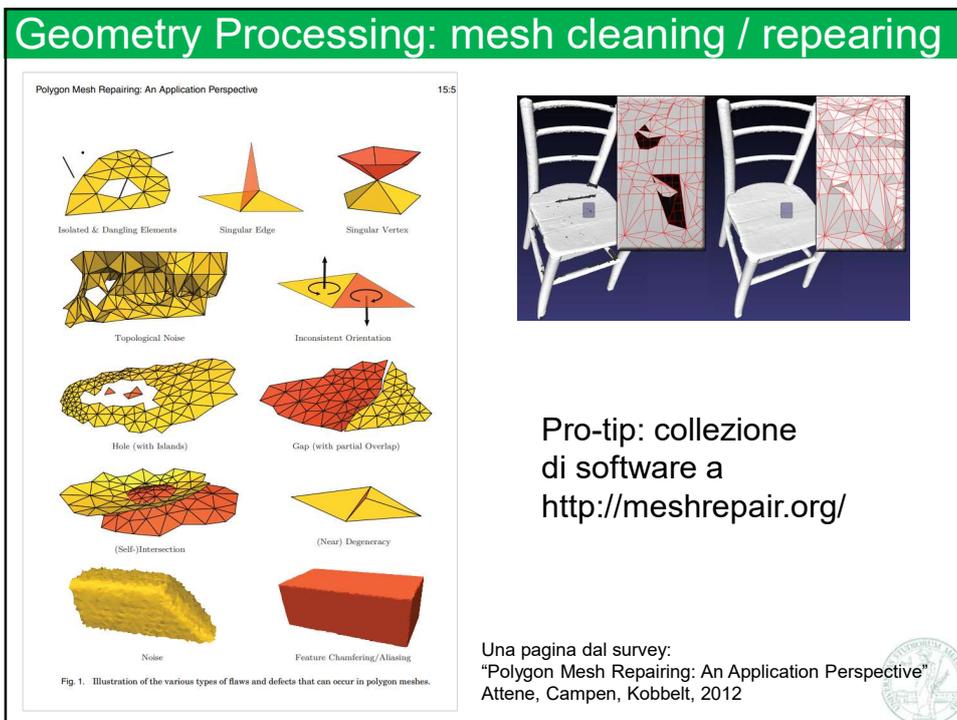
- ✓ Mettiamo che ciascun LOD abbia $\frac{1}{4}$ della risoluzione del LOD precedente (scelta ragionevole e spesso adottata)
- ✓ Domanda: quanto costa in memoria l'intera piramide, rispetto al costo (minimo) di memorizzare solo LOD-0?
- ✓ Ricordare che:
 - ⇒ Il costo in RAM di una mesh è LINEARE con la sua risoluzione,
- ⇒
$$\sum_{i=0}^{\infty} k^i = \frac{1}{1-k} \quad (\text{con } k < 1)$$
- ⇒ che, con $k = 1/4$, fa $4/3$ (cioè $1 + 1/3$)
- ✓ Risposta: solo $1/3$ in più!
- ✓ Conclusione: anche se la piramide LOD rappresenta una struttura multi-risoluzione triviale (semplice replicazione di dati a risoluzioni diverse), il suo costo in RAM non è eccessivo



119



121



122

Geometry Processing: mesh cleaning / reparing

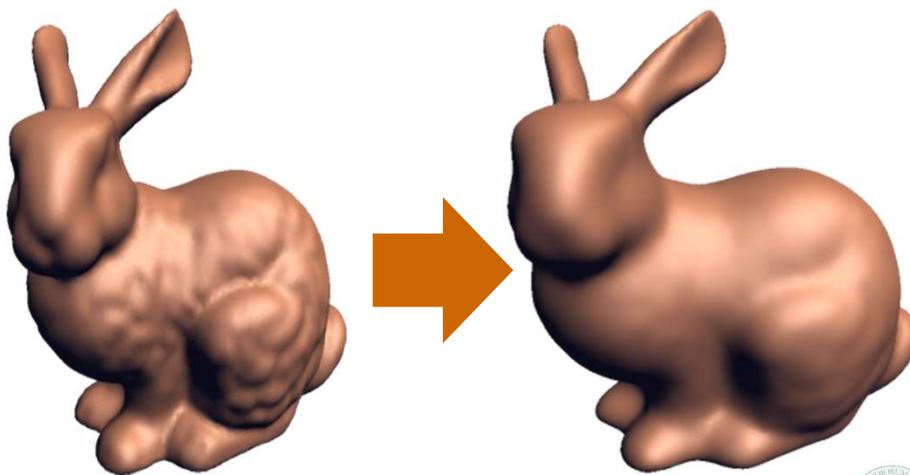
- ✓ Mesh cleaning (o reparing):
l'insieme di task per la rimozione dei difetti della di una mesh, cioè
 - ⇒ le situazioni non two-manifold,
 - ⇒ buchi (per es un poligono mancante)
 - ⇒ Orientamento non consistente delle facce
(se non è possibile farlo, ma mesh non è «ben orientabile»)
 - ⇒ Auto-intersezioni
 - ⇒ 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



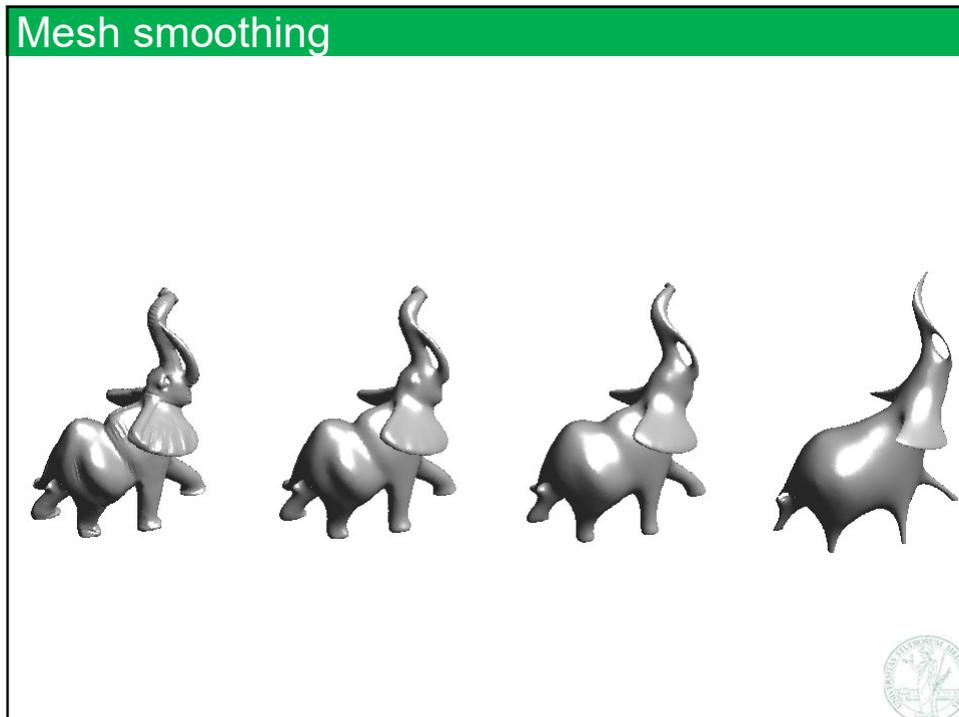
123

Geometry Processing tasks: examples

- ✓ Mesh smoothing



124



125

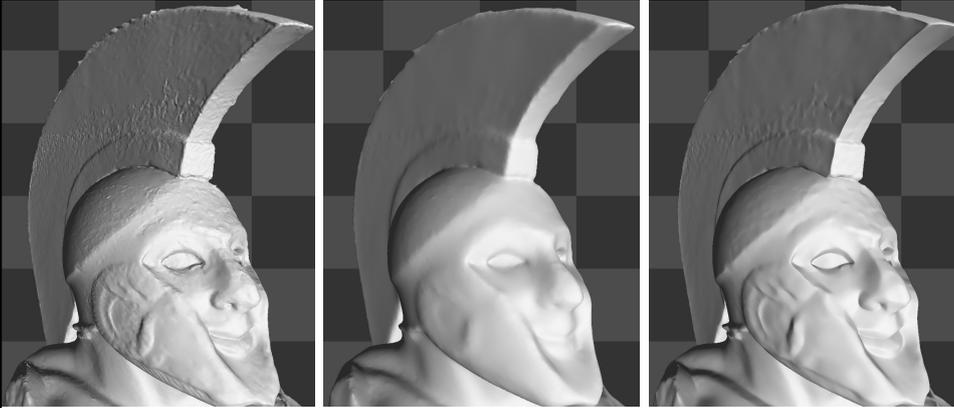
Mesh smoothing

- ✓ 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
 - ⇒ si riducono le asperità della superficie
- ✓ Effetti collaterali (spesso indesiderati):
 - ⇒ si abbattono anche i dettagli di forma utili
 - ⇒ si perdono ad esempio gli spigoli vivi (crease angles)
 - ⇒ il volume della mesh tende a ridursi

126

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)



127

Stima di proprietà geometriche

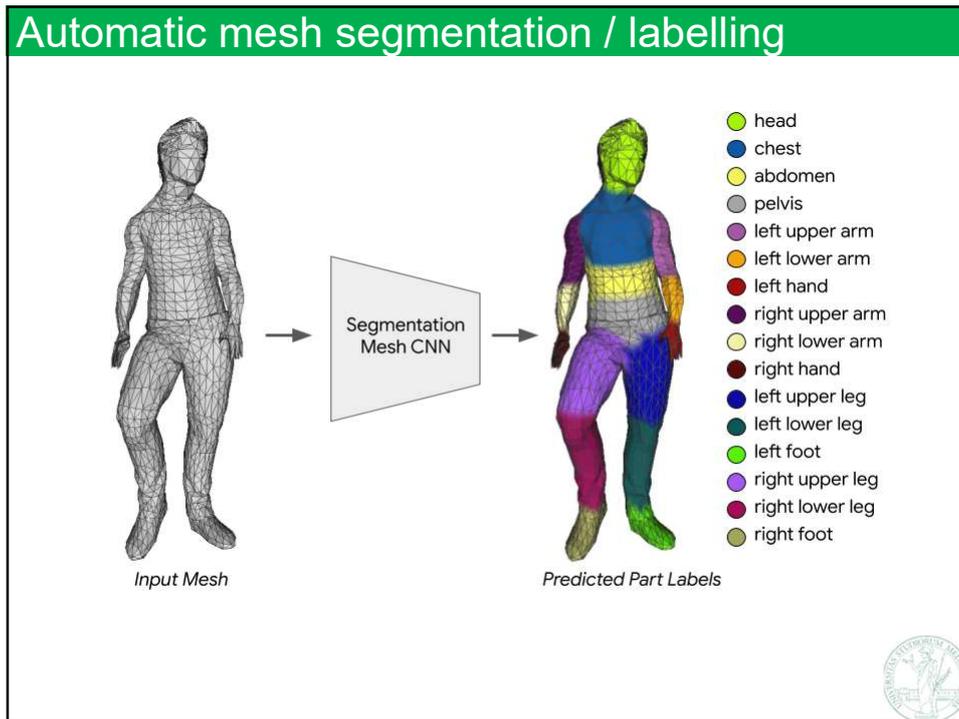


Img by Karan Singh

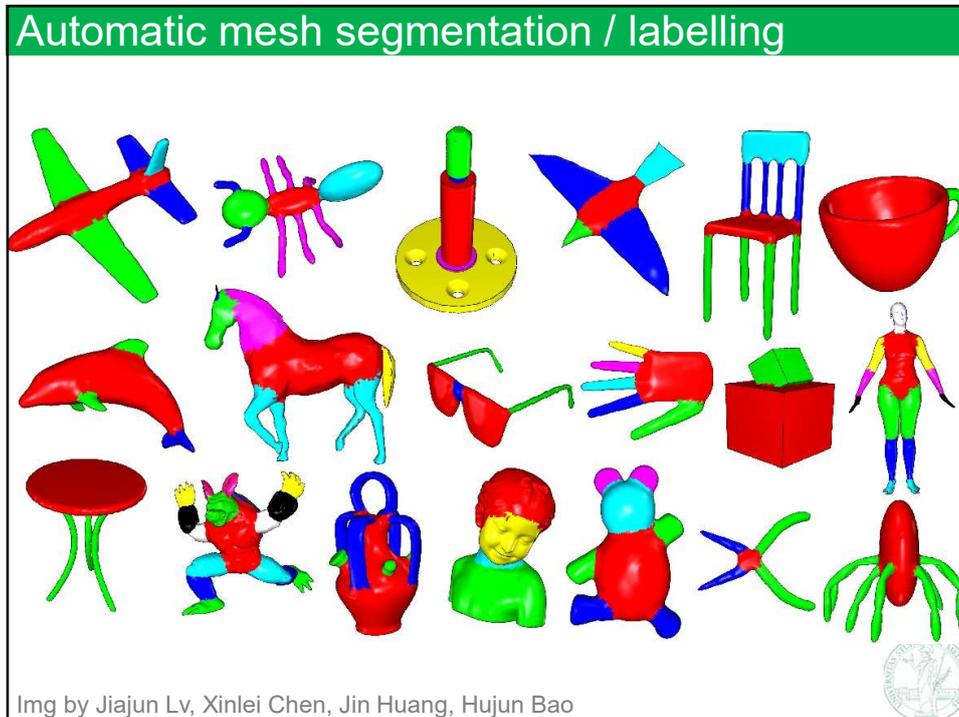
- ✓ Qui: curvatura (intrinseca)
- ✓ Abbiamo visto il caso della stima delle normali
 - ⇒ Sono solo un caso di proprietà di “forma” definite sulla superficie
 - ⇒ Proprietà di cui si occupa la geometria differenziale



128



129



130

Automatic mesh segmentation / labelling

- ✓ Mesh segmentation (in generale):
 - ⇒ Data una mesh in input, identificare le zone semanticamente o strutturalmente distinte
 - ⇒ (tipicamente, come partizione delle facce o dei vertici in zone contigue)
 - ⇒ Guidati da un'analisi della geometria, oppure data driven
- ✓ Mesh labelling
 - ⇒ Assegnare un'etichetta semantica ad ogni partizione
- ✓ Utilizzato come punto di partenza di molti altri task



131

Shape Retrieval



Img by Tamy Boubekeur, and Marc Alexa, SIGGR 2010



132

Shape Retrieval

- ✓ Il task: data una grossa collezione di modelli 3D, individuare quelli simili ad una forma target richiesta
 - ⇒ Spesso, la forma target è specificata attraverso un disegno semplificato 2D
 - ⇒ Oppure: “trovare le mesh che rappresentano oggetti simili a quello rappresentato da una mesh target data”.



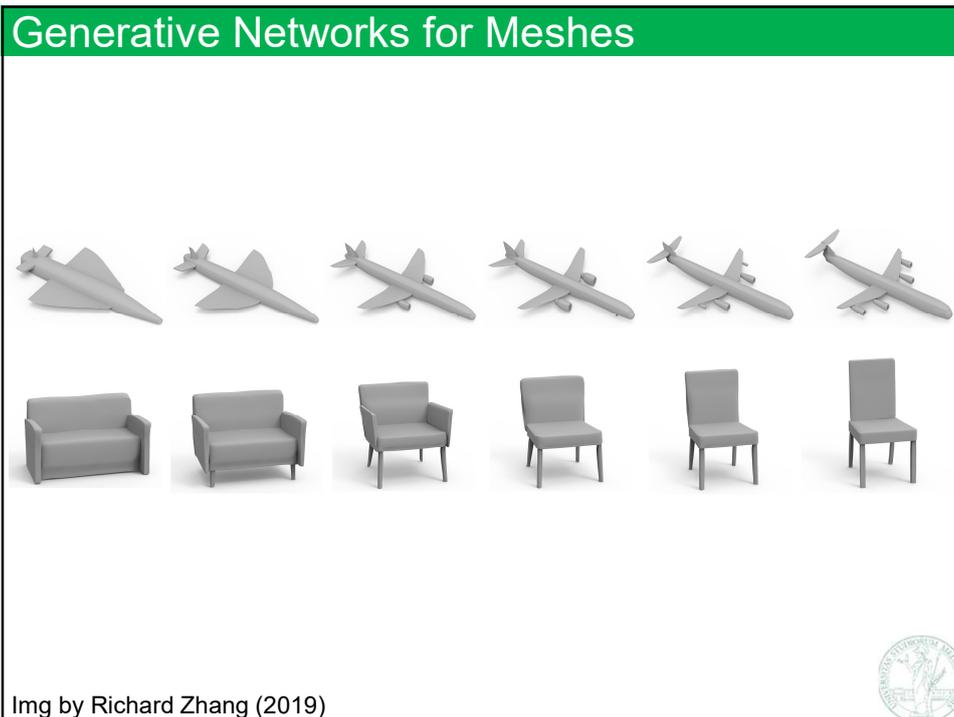
133

Procedural mesh generation

- ✓ Procedural mesh generation
 - ⇒ Nome generico per tutti quei procedimenti automatici che generano una mesh (geometria e connettività incluse)
 - ⇒ Utilizzato in videogames, VR, industria cinematografica, simulazioni...
 - ⇒ Molto studiati i casi di: piante, città, terreni, palazzi, manufatti di vario tipo, avatar umani, ...
 - ⇒ Tipicamente, guidato da parametri controllati dall'utente che determinano le caratteristiche «ad alto livello» delle forme generate (ad esempio: nella generazione automatica di mesh che rappresentano ingranaggi, un parametro può identificare «il numero di denti») e / o da scelte pseudocasuali
- ✓ Generative Networks:
L'approccio Machine learning approach alla mesh generation
 - ⇒ Data driven
 - ⇒ Per es: data una collezione di mesh che rappresentano forme di una certa classe (sedie, aerei...), produrre un insieme di varianti



134



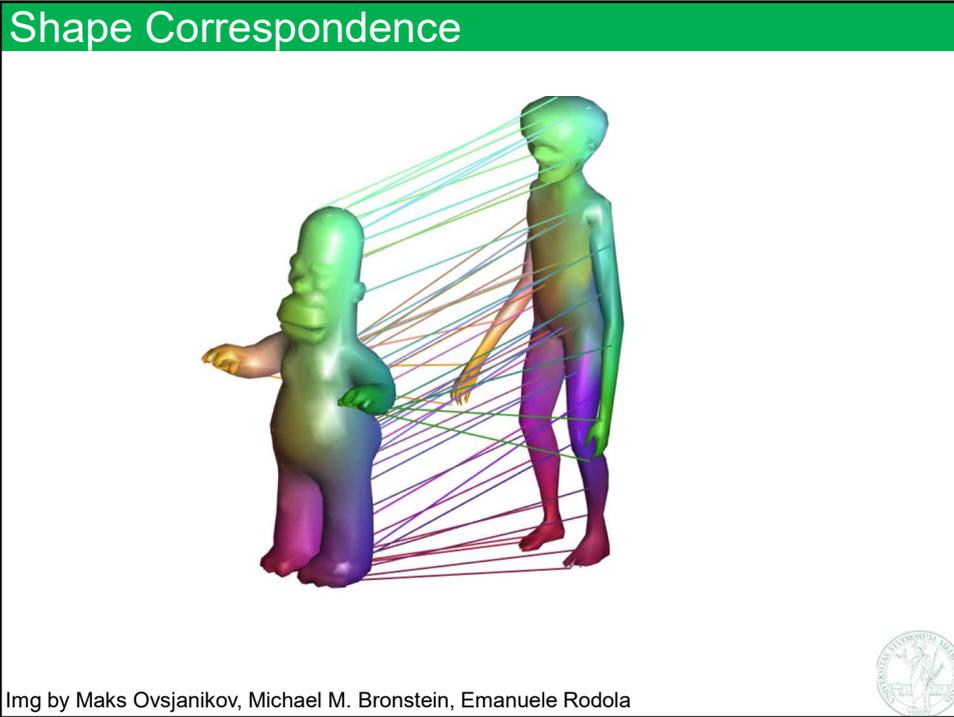
135

Shape correspondence

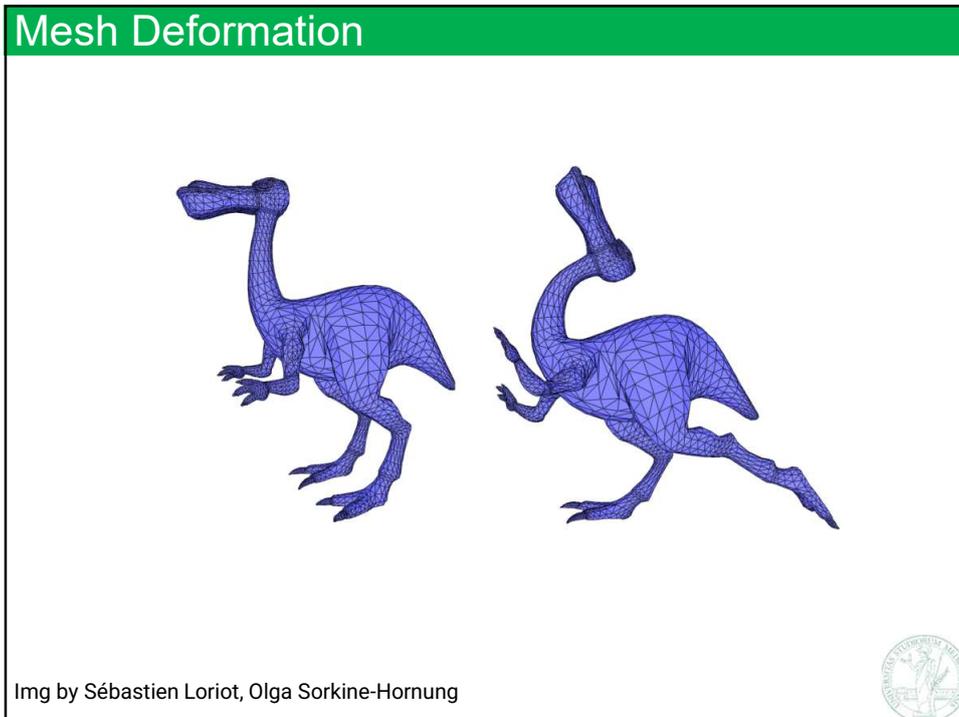
- ✓ Date due mesh che rappresentano oggetti di una stessa classe
 - ⇒ Per esempio, due forme umanoidi, o due quadrupedi
 - ⇒ Oppure, la scansione di uno stessa persona in due posizioni differenti...
- ✓ Identificare su ciascuna di essa il punto corrispondente sull'altra
 - ⇒ Ad esempio, sotto forma di un mapping (non biunivoco) fra i vertici
- ✓ Basandosi su similitudine geometrica, topologica, e caratteristiche intrinseche della forma
 - ⇒ (ad esempio: la similarità di normale è poco significativa ma quella di curvatura intrinseca può esserlo di più)



136



137



138

Mesh Deformation

- ✓ Data una mesh iniziale soggetta a degli stimoli o vincoli esterni, computare una sua deformazione spaziale
 - ⇒ cioè una nuova geometria, (la connettività della mesh è inalterata)
 - ⇒ La forma finale aderisce a principi fisici, geometrici (per es, conservazione dell'area, del volume, della forma dei triangoli, o semplicemente minimizzazione della distorsione subita dai singoli triangoli...)
 - ⇒ Esempio di vincolo esterno: una nuova la posizione xyz assegnata a solo alcuni vertici («questo vertice si sposta qui»)
- ✓ Utilizzata in animazione, design
 - ⇒ E' possibile effettuare questi computi in tempo reale



139

Task di Geometry Processing: meshing

- ✓ “Meshing”:
dato un modello 3D,
inizialmente non rappresentato come una mesh,
costruire una sua rappresentazione mesh
 - ⇒ è un esempio di Surface Reconstruction:
ricostruzione di una superficie
(da dati che non sono una superficie)
- ✓ Detta anche poligonizzazione,
o anche “segmentazione” in analogia con il 2D
 - ⇒ Per es,
«meshing di una nuvola di punti», come abbiamo visto
 - ⇒ Vedremo altri esempi di meshing a partire
da altre strutture dati più avanti nel corso



141

Geometry Processing: remeshing

- ✓ Remeshing:
data una superficie rappresentata come una mesh,
costruire una rappresentazione mesh diversa,
- ✓ In ambiente industriale, è detto “retopology”
⇒ specie quando fatto manualmente da un’artista
- ✓ La mesh di partenza differisce dalla mesh di arrivo
in termini di, per es...
 - ⇒ da tri a VS quad dominant VS quad («quad-remeshing»)
 - ⇒ (semi) regular VS irregular («semiregular-remeshing»)
 - ⇒ adaptive VS non adaptive resolution
 - ⇒ bad element shapes VS good element shapes



142

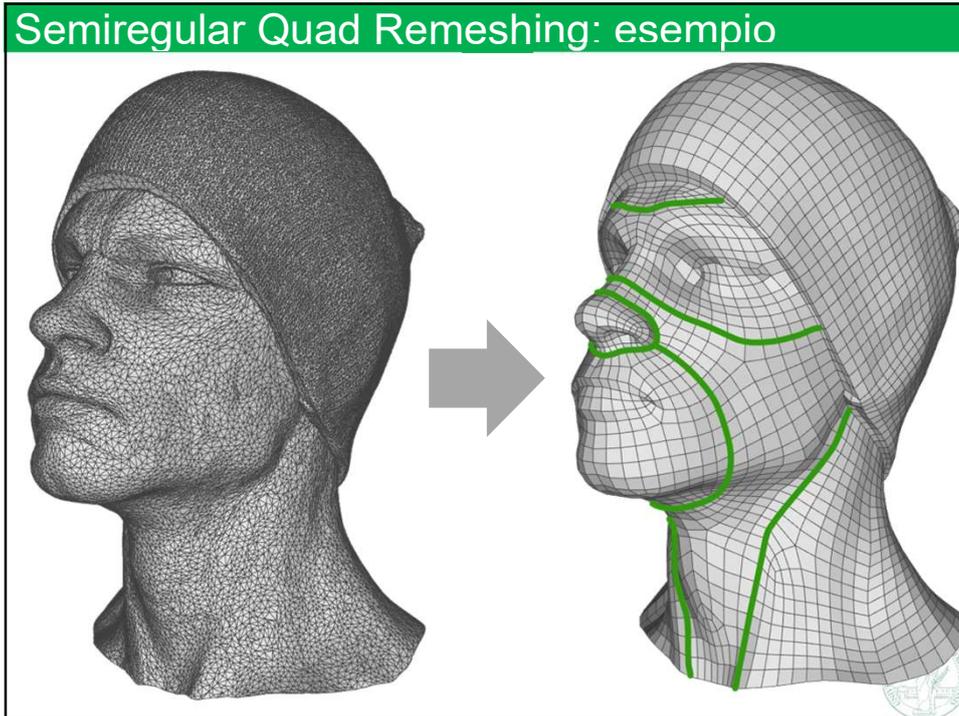
Geometry Processing: remeshing

Spesso fatto per migliorare la qualità della mesh:

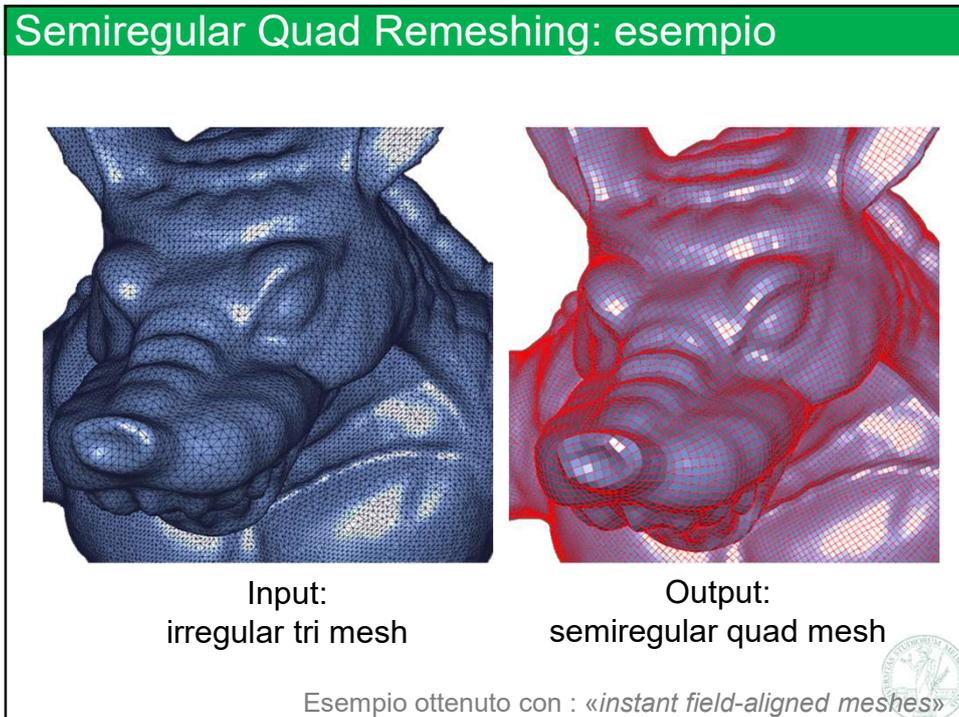
- ✓ rendere la risoluzione adattiva
- ✓ o, viceversa: rendere la risoluzione costante
 - ⇒ dimensione facce costante
(utile per es in una simulazione fisica)
- ✓ o, passare da triangoli a quads
- ✓ o, rendere la mesh regolare
 - ⇒ «semiregular remeshing»
 - ⇒ caso molto diffuso perchè:
le mesh catturate dal vero sono spesso irregolari
ma le mesh semi-regolari sono più utili:
più facili da editare, animare, etc



143



144



146