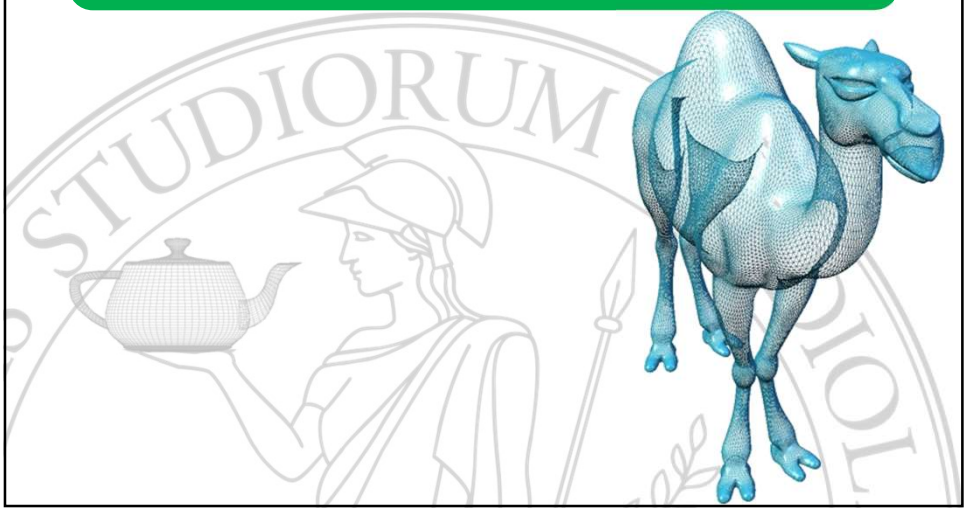
Marco Tarini - Computer Graphics 2021/2022
Università degli Studi di Milano

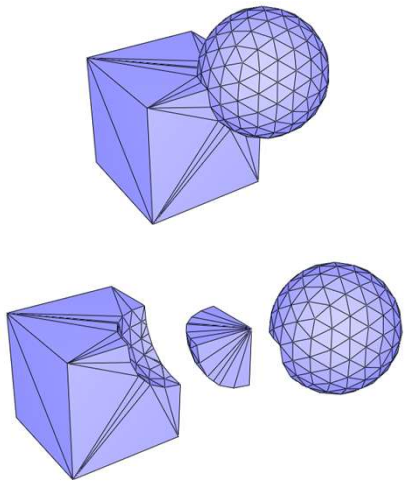
Mesh Processing - cenni



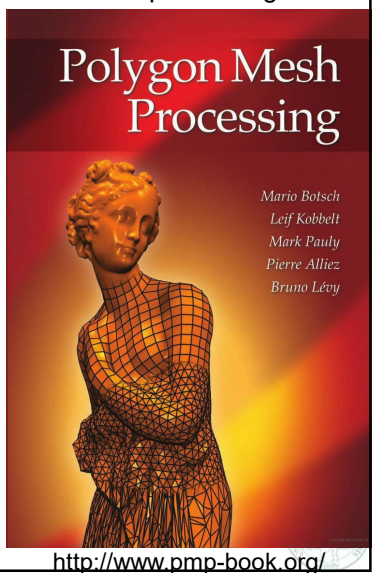
103

Mesh processing

Geometry processing eseguito su mesh poligonali



Un buon manuale per le basi al mesh processing:



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

104

Alcune lib di mesh processing (C++, opensource)

VCG-Lib

vision and
computer graphic library

CNR ( )

CGAL

computational geometry
algorithms library

INRIA ( )

OpenMesh
+ OpenFlipper

RWTH ( )

libigl

simple geometry
processing library

NYU ( )

105

Mesh Processing

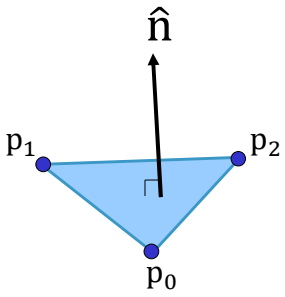
✓ Un buon applicativo
di mesh processing

MeshLab

106

Computo della normale di un triangolo

(Cioè del suo orientamento nello spazio)



due "edge vectors"

$$\vec{n} = (p_1 - p_0) \times (p_2 - p_0)$$

"area vector"
un vettore ortogonale al triangolo,
lungo quanto la doppia area del triangolo

normalizzazione

$$\hat{n} = \frac{\vec{n}}{\|\vec{n}\|}$$

"normale" del triangolo
(vett unitario)

E' la
doppia area
del triangolo

107

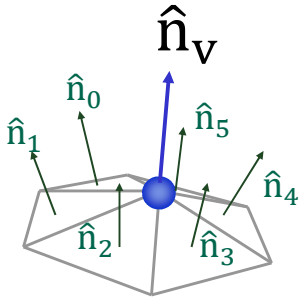
Calcolo delle normali per faccia di una mesh (note)

- ✓ La normale delle facce triangolari è data dal semplice computo visto sopra
  - ⇒ per ciascuna faccia
- ✓ E' detta normale «geometrica» del faccia
  - ⇒ corrisponde effettivamente alla normale del piano su cui giace il triangolo (che è costante)
  - ⇒ esiste sempre (nota: questo non vale per facce quads – a meno che non siano planari)
  - ⇒ eccezione: triangoli degeneri (con 3 vertici allineati o, 2 vertici coincidenti, etc) (altrimenti, cosa succede?)
- ✓ Questa normale è orientata consistentemente (nello stesso verso) solo se la mesh è ben orientata
  - ⇒ (Altrimenti, cosa succede?)
  - ⇒ ad es, sempre verso l'esterno in una mesh chiusa

109

Normali come attributo per vertice

Normale di un vertice
condiviso da *k* triangoli:
la loro media (interpolazione)



$$\hat{n}_v = \frac{\hat{n}_0 + \cdots + \hat{n}_k}{\|\hat{n}_0 + \cdots + \hat{n}_k\|}$$

Nota: dato che si normalizza il risultato,
non c'è bisogno di divider per *k* la somma delle normali per faccia per calcolare la loro media

110

Calcolo delle normali per vertice di una mesh (note)

- ✓ La normale per vertice di una mesh non è definita in modo univoco per via geometrica
  - ⇒ quindi dobbiamo «inventarcene» una.
- ✓ La domanda che ci dobbiamo porre è:
  - ⇒ «se questa mesh (che è composta di facce *piatte*) modella una superficie invece *curva*, quale vale la normali di questa superficie curva, sul vertice?»
  - ⇒ Non esiste una risposta univoca!
  - ⇒ Dipende da quale superficie si intende rappresentare.
- ✓ Strategia spesso utilizzata in pratica:
  - ⇒ usare una interpolazione delle facce adiacenti (per es, la media)
- ✓ *le normali (per vertice) fanno parte del modello* e spesso vengono costruite insieme al modello.
  - ⇒ Solo nei casi di una mesh sprovvista di normali, sarà necessario computarle dalla geometria (e dalla connettività)

111

Algoritmo per computo di normale per vertice

Passo 1: computo delle normali per faccia

- ✓ Per facce non-triangolari: si possono mediare le normali costruite su ogni wedge della faccia
  - ⇒ prodotto cross dei due edge vectors adiacienti al wedge
  - ⇒ Per trovarli, navigo gli half edge attorno alla faccia

Passo 2: computo delle normali per vertice

- ✓ ∀ vertice **v**:
  ciclo su tutte le facce adiacenti a **v**,
  sommando la loro normale in un vettore, e normalizzo questo vettore

Problema: come trovo tutte le facce adiacenti ad un vertice dato?

- ✓ Se scandisco l'intero vettore della «lista-facce»
  il mio algoritmo diviene *quadratico*
  - ⇒ se ho k vertici e 2k facce, richiede ~ 2k^2 operazioni!
  - ⇒ nel mesh processing, gli algortmi quadratici non sono feasible (k = molto grande)

Esiste un algoritmo efficiente (lineare)
che usa solo la struttura «lista-facce» per la.
Sapresti identificare questo algoritmo?

112

Algoritmo per computo di normale per vertice

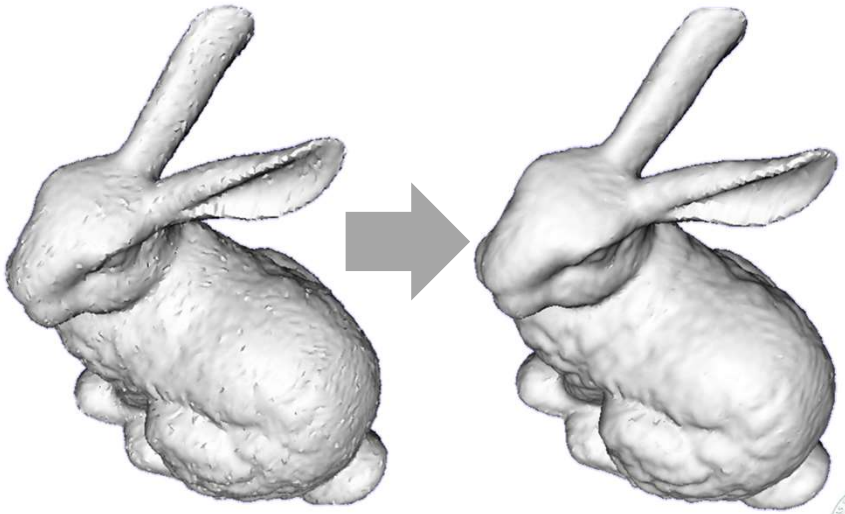
- ✓ Passo 1 ∀ vertice: azzero il suo vettore normale
  - ⇒ Questo vettore servirà per mantenere le somme parziali
- ✓ Passo 2 ∀ faccia: calcolo la sua normale,
  e la sommo alla normale di ciascun vertice ai suoi corners
  - ⇒ Alla fine di questo ciclo, su ogni vertice avrò accumulato un vettori da ciascuna faccia adiacente a quel vertice
- ✓ Passo 3 ∀ vertice: normalizzo il risultato

- ✓ La strategia generale viene a volte detta scattering:
  - ⇒ Al passo 2, invece che raccogliere (*gather*) su ogni vertice le normali dalle facce adiacenti,
    distribuisco (*scatter*) da ogni faccia le normali ai vertici adiacenti
  - ⇒ La strategia dell'algoritmo precedente viene detta *gathering*
- ✓ Anche se il risultato è lo stesso, questo algoritmo è linare:
  - ⇒ Se ho k vertici e 2k facce, mi ci vogliono solo ~ k + 2k + k operazioni
  - ⇒ Nel geometry processing, gli algoritmi lineari sono feasible: sono veloci anche per mesh a risoluzione molto grande (per es, K = 10^9)

113

Geometry Processing tasks: examples

✓Mesh denoising

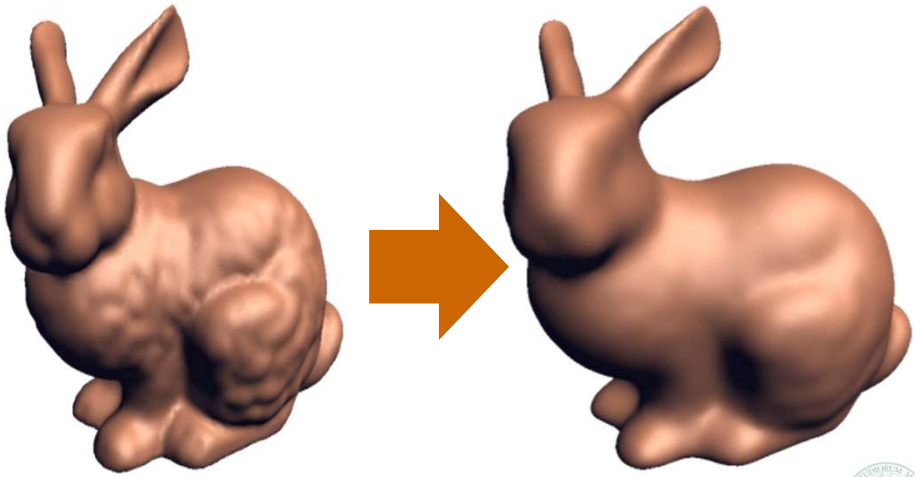


(img by Iman Sadeghi)

114

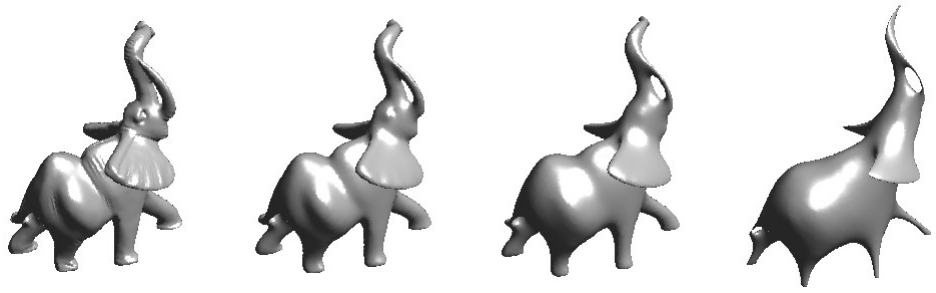
Geometry Processing tasks: examples

✓Mesh smoothing



115

Mesh smoothing



116

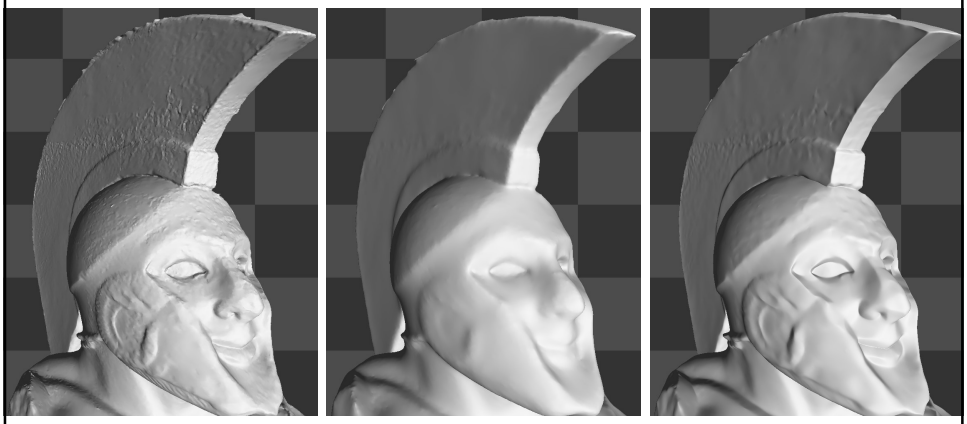
Mesh smoothing

- ✓ 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 blur di un'immagine,
  - ⇒ in cui si sostituisce ad ogni pixel il valore medio dei pixel vicini, sfocando l'immagine
- ✓ 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)
  - ⇒ la mesh tende a ridursi complessivamente in estensione

117

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)

118